Please type a plus sign (+) inside this box → [+]

MODIFIED PTO/SB/05 (03 0

Approved for use through 10/31/2002 OMB 0651-0

U S Patent and Trademark Office U.S DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 C F R 1 53(b))

Attorney Docket No	000479 00082
First Inventor	Edmond Colby Munger
Title	IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY
Express Mail Label No	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents

ADDRESS TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231

1. ☒ Fee Transmittal Form (e g , PTO/SB/17)
   (Submit an original and a duplicate for fee processing)
2. ☐ Applicant claims small entity status
   See 37 CFR 1.27
3. ☒ Specification [Total Pages 76]
   (preferred arrangement set forth below)
   - Descriptive title of the Invention
   - Cross Reference to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to sequence listing, a table, or a computer program listing appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings (if filed)
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
4. ☒ Drawing(s) (35 U S C 113) [Total Sheets 35]
   a ☒ Formal, or
   b ☐ Informal
5. Oath or Declaration [Total Pages 12]
   a ☐ Newly executed (original or copy), or
   b ☒ Copy from a prior application (37 CFR 1 63 (d))
   (for a continuation/divisional with Box 18 completed)
   i ☐ DELETION OF INVENTOR(S)
   Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1 63(d)(2) and 1 33(b)
6. ☒ Application Data Sheet. See 37 CFR 1.76
7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)
8. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
   a ☐ Computer Readable Form (CRF)
   b Specification Sequence Listing on
   i. ☐ CD-ROM or CD-R (2 copies), or
   ii. ☐ paper
   c ☐ Statements verifying identity of above copies

ACCOMPANYING APPLICATIONS PARTS

9. ☐ Assignment Papers (cover sheet & document(s))
10. ☐ 37 C F R §3.73(b) Statement (when there is an assignee) ☐ Power of Attorney
11. ☐ English Translation Document (if applicable)
12. ☒ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
13. ☐ Preliminary Amendment
14. ☒ Return Receipt Postcard (MPEP 503) (Should be specifically itemized)
15. ☐ Certified Copy of Priority Document(s) (if foreign priority is claimed)
16. ☐ Nonpublication Request under 35 U S C 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent
17. ☐ Other _____

18. If a CONTINUING APPLICATION, *check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1 76*

☐ Continuation ☒ Divisional ☐ Continuation-in-part (CIP) of prior application No. 09 / 504,783

Prior application information Examiner Krisna Lim Group / Art Unit. 2153

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. CORRESPONDENCE ADDRESS

☒ Customer Number or Bar Code Label 22907 (Insert Customer No or Attach bar code label here) or ☐ Correspondence address below

Name					
Address					
City		State		Zip Code	
Country		Telephone		Fax	

Name (Print/Type)	Ross A Dannenberg	Registration No (Attorney/Agent)	49,024
Signature	[signature]	Date	September 30, 2002

FEE TRANSMITTAL for FY 2002

Patent fees are subject to annual revision

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 1076

Complete if Known	
Application Number	TBA
Filing Date	September 30, 2002
First Named Inventor	Edmond Colby Munger
Examiner Name	TBA
Group / Art Unit	2153
Attorney Docket No	000479 00082

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account

Deposit Account Number: 19-0733

Deposit Account Name: Banner & Witcoff, Ltd

The Commissioner is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☐ Charge any additional fee(s) during the pendency of this application

☐ Charge fee(s) indicated below, **except for the filing fee**

to the above-identified deposit account

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	740	201	370	Utility filing fee	740
106	330	206	165	Design filing fee	
107	510	207	255	Plant filing fee	
108	740	208	370	Reissue filing fee	
114	160	214	80	Provisional filling fee	

SUBTOTAL (1) ($) 740

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims	20	-20 ** =	0	X		=	0
Independent Claims	7	-3 ** =	4	X	84	=	336
Multiple Dependent				X		=	0

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	84	202	42	Independent claims in excess of 3
104	280	204	140	Multiple dependent claim, if not paid
109	84	209	42	** Reissue independent claims over original patent
110	18	210	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 336

**or number previously paid, if greater, For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	400	216	200	Extension for reply within second month	
117	920	217	460	Extension for reply within third month	
118	1,440	218	720	Extension for reply within fourth month	
128	1,960	228	980	Extension for reply within fifth month	
119	320	219	160	Notice of Appeal	
120	320	220	160	Filing a brief in support of an appeal	
121	280	221	140	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive – unavoidable	
141	1,280	241	640	Petition to revive – unintentional	
142	1,280	242	640	Utility issue fee (or reissue)	
143	460	243	230	Design issue fee	
144	620	244	310	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Processing fee under 37 CFR 1 17 (q)	
126	180	126	180	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	740	246	370	Filing a submission after final rejection (37 CFR § 1 129(a))	
149	740	249	370	For each additional invention to be examined (37 CFR § 1 129(b))	
179	740	279	370	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 0

SUBMITTED BY

Complete *(if applicable)*

Name (Print/Type)	Ross A Dannenberg	*Registration No Attorney/Agent)*	49,024	*Telephone*	(202) 508-9153
Signature	Ross D			*Date*	September 30, 2002

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Application Data Sheet

Application Information

Application number::	
Filing Date::	
Application Type::	Regular
Subject Matter::	Utility
Suggested classification::	
Suggested Group Art Unit::	
CD-ROM or CD-R?::	None
Number of CD disks::	
Number of copies of CDs::	
Sequence submission?::	
Computer Readable Form (CRF)?::	
Number of copies of CRF::	
Title::	IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY
Attorney Docket Number::	000479.00082
Request for Early Publication?::	NO
Request for Non-Publication?::	NO
Suggested Drawing Figure::	
Total Drawing Sheets::	35
Small Entity?::	NO
Latin name::	
Variety denomination name::	
Petition included?::	NO
Petition Type::	
Licensed US Govt. Agency::	
Contract or Grant Numbers::	

Secrecy Order in Parent Appl.?:: NO

Applicant Information

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Edward
Middle Name::	Colby
Family Name::	Munger
Name Suffix::	
City of Residence::	Crownsville
State or Province of Residence::	MD
Country of Residence::	USA
Street of mailing address::	1101 Opaca Court
City of mailing address::	Crownsville
State or Province of mailing address::	MD
Country of mailing address::	USA
Postal or Zip Code of mailing address::	21032

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Douglas
Middle Name::	Charles
Family Name::	Schmidt
Name Suffix::	
City of Residence::	Severna Park
State or Province of Residence::	MD
Country of Residence::	USA
Street of mailing address::	230 Oak Court

City of mailing address::	Serverna Park
State or Province of mailing address::	MD
Country of mailing address::	USA
Postal or Zip Code of mailing address::	21146

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Robert
Middle Name::	Dunham
Family Name::	Short
Name Suffix::	III
City of Residence::	Leesburg
State or Province of Residence::	VA
Country of Residence::	USA
Street of mailing address::	38710 Goose Creek Lane
City of mailing address::	Leesburg
State or Province of mailing address::	VA
Country of mailing address::	USA
Postal or Zip Code of mailing address::	20175

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Victor
Middle Name::	
Family Name::	Larson
Name Suffix::	
City of Residence::	Fairfax
State or Province of Residence::	VA

Country of Residence::	USA
Street of mailing address::	12026 Lisa Marie Court
City of mailing address::	Fairfax
State or Province of mailing address::	VA
Country of mailing address::	USA
Postal or Zip Code of mailing address::	22033

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Michael
Middle Name::	
Family Name::	Williamson
Name Suffix::	
City of Residence::	South Riding
State or Province of Residence::	VA
Country of Residence::	USA
Street of mailing address::	26203 Ocala Circle
City of mailing address::	South Riding
State or Province of mailing address::	VA
Country of mailing address::	USA
Postal or Zip Code of mailing address::	20152

Correspondence Information

Correspondence Customer Number:: 22907

Representative Information

Representative Customer Number:: 22907

Domestic Priority Information

Application::	Continuity Type::	Parent Application::	Parent Filing Date::
This Application	Division of	09/504,783	02/15/00

Foreign Priority Information

Country::	Application number::	Filing Date::	Priority Claimed::

Assignee Information

Assignee name::	Science Applications International Corporation
Street of mailing address::	10260 Campus Point Drive
City of mailing address::	San Diego
State or Province of mailing address::	CA
Country of mailing address::	USA
Postal or Zip Code of mailing address::	92121

000479.00082

IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

CROSS-REFERENCE TO RELATED APPLICATIONS

[01] This application is a divisional application of 09/504,783 (filed February 15, 2000), which claims priority from and is a continuation-in-part of previously filed U.S. application serial number 09/429,643 (filed October 29, 1999). The subject matter of that application, which is bodily incorporated herein, derives from provisional U.S. application numbers 60/106,261 (filed October 30, 1998) and 60/137,704 (filed June 7, 1999).

BACKGROUND OF THE INVENTION

[02] A tremendous variety of methods have been proposed and implemented to provide security and anonymity for communications over the Internet. The variety stems, in part, from the different needs of different Internet users. A basic heuristic framework to aid in discussing these different security techniques is illustrated in FIG. 1. Two terminals, an originating terminal 100 and a destination terminal 110 are in communication over the Internet. It is desired for the communications to be secure, that is, immune to eavesdropping. For example, terminal 100 may transmit secret information to terminal 110 over the Internet 107. Also, it may be desired to prevent an eavesdropper from discovering that terminal 100 is in communication with terminal 110. For example, if terminal 100 is a user and terminal 110 hosts a web site, terminal 100's user may not want anyone in the intervening networks to know what web sites he is "visiting." Anonymity would thus be an issue, for example, for companies that want to keep their market research interests private and thus would prefer to prevent outsiders from knowing which web-sites or other Internet resources they are "visiting." These two security issues may be called data security and anonymity, respectively.

[03] Data security is usually tackled using some form of data encryption. An encryption key 48 is known at both the originating and terminating terminals 100 and 110. The keys may be private and public at the originating and destination terminals 100 and 110, respectively or they may be symmetrical keys (the same key is used by both parties to encrypt and decrypt). Many encryption methods are known and usable in this context.

[04] To hide traffic from a local administrator or ISP, a user can employ a local proxy server in communicating over an encrypted channel with an outside proxy such that the local administrator or ISP only sees the encrypted traffic. Proxy servers prevent destination servers from determining the identities of the originating clients. This system employs an intermediate server interposed between client and destination server. The destination server sees only the Internet Protocol (IP) address of the proxy server and not the originating client. The target server only sees the address of the outside proxy. This scheme relies on a trusted outside proxy server. Also, proxy schemes are vulnerable to traffic analysis methods of determining identities of transmitters and receivers. Another important limitation of proxy servers is that the server knows the identities of both calling and called parties. In many instances, an originating terminal, such as terminal A, would prefer to keep its identity concealed from the proxy, for example, if the proxy server is provided by an Internet service provider (ISP).

[05] To defeat traffic analysis, a scheme called Chaum's mixes employs a proxy server that transmits and receives fixed length messages, including dummy messages. Multiple originating terminals are connected through a mix (a server) to multiple target servers. It is difficult to tell which of the originating terminals are communicating to which of the connected target servers, and the dummy messages confuse eavesdroppers' efforts to detect communicating pairs by analyzing traffic. A drawback is that there is a risk that the mix server could be compromised. One way to deal with this risk is to spread the trust among multiple mixes. If one mix is compromised, the identities of the originating and target terminals may remain concealed. This strategy requires a number of alternative mixes so that the intermediate servers interposed between the originating and target terminals are not determinable except by compromising more than one mix. The strategy wraps the message with multiple layers of encrypted addresses. The first mix in a sequence can decrypt only the outer layer of the message to reveal the next destination mix in sequence. The second mix can decrypt the message to reveal the next mix and so on. The target server receives the message and, optionally, a multi-layer encrypted payload containing return information to send data back in the same fashion. The only way to defeat such a mix scheme is to collude among mixes. If the packets are all fixed-length and intermixed with dummy packets, there is no way to do any kind of traffic analysis.

[06] Still another anonymity technique, called 'crowds,' protects the identity of the originating terminal from the intermediate proxies by providing that originating terminals belong to groups of proxies called crowds. The crowd proxies are interposed between originating and target terminals. Each proxy through which the message is sent is randomly chosen by an upstream proxy. Each intermediate proxy can send the message either to another randomly chosen proxy in the "crowd" or to the destination. Thus, even crowd members cannot determine if a preceding proxy is the originator of the message or if it was simply passed from another proxy.

[07] ZKS (Zero-Knowledge Systems) Anonymous IP Protocol allows users to select up to any of five different pseudonyms, while desktop software encrypts outgoing traffic and wraps it in User Datagram Protocol (UDP) packets. The first server in a 2+-hop system gets the UDP packets, strips off one layer of encryption to add another, then sends the traffic to the next server, which strips off yet another layer of encryption and adds a new one. The user is permitted to control the number of hops. At the final server, traffic is decrypted with an untraceable IP address. The technique is called onion-routing. This method can be defeated using traffic analysis. For a simple example, bursts of packets from a user during low-duty periods can reveal the identities of sender and receiver.

[08] Firewalls attempt to protect LANs from unauthorized access and hostile exploitation or damage to computers connected to the LAN. Firewalls provide a server through which all access to the LAN must pass. Firewalls are centralized systems that require administrative overhead to maintain. They can be compromised by virtual-machine applications ("applets"). They instill a false sense of security that leads to security breaches for example by users sending sensitive information to servers outside the firewall or encouraging use of modems to sidestep the firewall security. Firewalls are not useful for distributed systems such as business travelers, extranets, small teams, etc.

SUMMARY OF THE INVENTION

[09] A secure mechanism for communicating over the internet, including a protocol referred to as the Tunneled Agile Routing Protocol (TARP), uses a unique two-layer encryption format and special TARP routers. TARP routers are similar in function to regular IP routers. Each TARP router has one or more IP addresses and uses normal IP protocol to send IP packet

messages ("packets" or "datagrams"). The IP packets exchanged between TARP terminals via TARP routers are actually encrypted packets whose true destination address is concealed except to TARP routers and servers. The normal or "clear" or "outside" IP header attached to TARP IP packets contains only the address of a next hop router or destination server. That is, instead of indicating a final destination in the destination field of the IP header, the TARP packet's IP header always points to a next-hop in a series of TARP router hops, or to the final destination. This means there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet since the destination could always be next-hop TARP router as well as the final destination.

[10] Each TARP packet's true destination is concealed behind a layer of encryption generated using a link key. The link key is the encryption key used for encrypted communication between the hops intervening between an originating TARP terminal and a destination TARP terminal. Each TARP router can remove the outer layer of encryption to reveal the destination router for each TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal by the sender/receiver IP numbers in the cleartext IP header.

[11] Once the outer layer of encryption is removed, the TARP router determines the final destination. Each TARP packet 140 undergoes a minimum number of hops to help foil traffic analysis. The hops may be chosen at random or by a fixed value. As a result, each TARP packet may make random trips among a number of geographically disparate routers before reaching its destination. Each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined. This feature is called *agile routing*. The fact that different packets take different routes provides distinct advantages by making it difficult for an interloper to obtain all the packets forming an entire multi-packet message. The associated advantages have to do with the inner layer of encryption discussed below. Agile routing is combined with another feature that furthers this purpose; a feature that ensures that any message is broken into multiple packets.

[12] The IP address of a TARP router can be changed, a feature called *IP agility*. Each TARP router, independently or under direction from another TARP terminal or router, can change its IP address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs.

[13] The message payload is hidden behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the intervening TARP routers. The session key is used to decrypt the payloads of the TARP packets permitting the data stream to be reconstructed.

[14] Communication may be made private using link and session keys, which in turn may be shared and used according to any desired method. For example, public/private keys or symmetric keys may be used.

[15] To transmit a data stream, a TARP originating terminal constructs a series of TARP packets from a series of IP packets generated by a network (IP) layer process. (Note that the terms "network layer," "data link layer," "application layer," etc. used in this specification correspond to the Open Systems Interconnection (OSI) network terminology.) The payloads of these packets are assembled into a block and chain-block encrypted using the session key. This assumes, of course, that all the IP packets are destined for the same TARP terminal. The block is then interleaved and the interleaved encrypted block is broken into a series of payloads, one for each TARP packet to be generated. Special TARP headers IP_T are then added to each payload using the IP headers from the data stream packets. The TARP headers can be identical to normal IP headers or customized in some way. They should contain a formula or data for deinterleaving the data at the destination TARP terminal, a time-to-live (TTL) parameter to indicate the number of hops still to be executed, a data type identifier which indicates whether the payload contains, for example, TCP or UDP data, the sender's TARP address, the destination TARP address, and an indicator as to whether the packet contains real or decoy data or a formula for filtering out decoy data if decoy data is spread in some way through the TARP payload data.

[16] Note that although chain-block encryption is discussed here with reference to the session key, any encryption method may be used. Preferably, as in chain block encryption, a method should be used that makes unauthorized decryption difficult without an entire result of the encryption process. Thus, by separating the encrypted block among multiple packets and making it difficult for an interloper to obtain access to all of such packets, the contents of the communications are provided an extra layer of security.

[17] Decoy or dummy data can be added to a stream to help foil traffic analysis by reducing the peak-to-average network load. It may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

[18] Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to portion, or entirety, of a message, and that portion or entirety then interleaved into a number of separate packets. Considering the agile IP routing of the packets, and the attendant difficulty of reconstructing an entire sequence of packets to form a single block-encrypted message element, decoy packets can significantly increase the difficulty of reconstructing an entire data stream.

[19] The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system. Because the encryption system described above is insertable between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the Network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the

difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

[20] IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

[21] As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicating that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. In addition, it may create a subprocess that maintains the original IP address and continues interacting with the attacker in some manner.

[22] Decoy packets may be generated by each TARP terminal on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more likely to insert decoy packets when a message stream is being received. Alternatively, if a large number of decoy packets is received along with regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis.

[23] In various other embodiments of the invention, a scalable version of the system may be constructed in which a plurality of IP addresses are preassigned to each pair of communicating nodes in the network. Each pair of nodes agrees upon an algorithm for "hopping" between IP addresses (both sending and receiving), such that an eavesdropper sees apparently continuously random IP address pairs (source and destination) for packets transmitted between the pair. Overlapping or "reusable" IP addresses may be allocated to different users on the same subnet, since each node merely verifies that a particular packet includes a valid source/destination pair from the agreed-upon algorithm. Source/destination pairs are preferably not reused between any two nodes during any given end-to-end session, though limited IP block sizes or lengthy sessions might require it.

[24] Further improvements described in this continuation-in-part application include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities

BRIEF DESCRIPTION OF THE DRAWINGS

[25] FIG. 1 is an illustration of secure communications over the Internet according to a prior art embodiment.

[26] FIG. 2 is an illustration of secure communications over the Internet according to a an embodiment of the invention.

[27] FIG. 3a is an illustration of a process of forming a tunneled IP packet according to an embodiment of the invention.

[28] FIG. 3b is an illustration of a process of forming a tunneled IP packet according to another embodiment of the invention.

[29] FIG. 4 is an illustration of an OSI layer location of processes that may be used to implement the invention.

[30] FIG. 5 is a flow chart illustrating a process for routing a tunneled packet according to an embodiment of the invention.

[31] FIG. 6 is a flow chart illustrating a process for forming a tunneled packet according to an embodiment of the invention.

[32] FIG. 7 is a flow chart illustrating a process for receiving a tunneled packet according to an embodiment of the invention.

[33] FIG. 8 shows how a secure session is established and synchronized between a client and a TARP router.

[34] FIG. 9 shows an IP address hopping scheme between a client computer and TARP router using transmit and receive tables in each computer.

[35] FIG. 10 shows physical link redundancy among three Internet Service Providers (ISPs) and a client computer.

[36] FIG. 11 shows how multiple IP packets can be embedded into a single "frame" such as an Ethernet frame, and further shows the use of a discriminator field to camouflage true packet recipients.

[37] FIG. 12A shows a system that employs hopped hardware addresses, hopped IP addresses, and hopped discriminator fields.

[38] FIG. 12B shows several different approaches for hopping hardware addresses, IP addresses, and discriminator fields in combination.

[39] FIG. 13 shows a technique for automatically re-establishing synchronization between sender and receiver through the use of a partially public sync value.

[40] FIG. 14 shows a "checkpoint" scheme for regaining synchronization between a sender and recipient.

[41] FIG. 15 shows further details of the checkpoint scheme of FIG. 14.

[42] FIG. 16 shows how two addresses can be decomposed into a plurality of segments for comparison with presence vectors.

[43] FIG. 17 shows a storage array for a receiver's active addresses.

[44] FIG. 18 shows the receiver's storage array after receiving a sync request.

[45] FIG. 19 shows the receiver's storage array after new addresses have been generated.

[46] FIG. 20 shows a system employing distributed transmission paths.

[47] FIG. 21 shows a plurality of link transmission tables that can be used to route packets in the system of FIG. 20.

[48] FIG. 22A shows a flowchart for adjusting weight value distributions associated with a plurality of transmission links.

[49] FIG. 22B shows a flowchart for setting a weight value to zero if a transmitter turns off.

[50] FIG. 23 shows a system employing distributed transmission paths with adjusted weight value distributions for each path.

[51] FIG. 24 shows an example using the system of FIG. 23.

[52] FIG. 25 shows a conventional domain-name look-up service.

[53] FIG. 26 shows a system employing a DNS proxy server with transparent VPN creation.

[54] FIG. 27 shows steps that can be carried out to implement transparent VPN creation based on a DNS look-up function.

[55] FIG. 28 shows a system including a link guard function that prevents packet overloading on a low-bandwidth link LOW BW.

[56] FIG. 29 shows one embodiment of a system employing the principles of FIG. 28.

[57] FIG. 30 shows a system that regulates packet transmission rates by throttling the rate at which synchronizations are performed.

[58] FIG. 31 shows a signaling server 3101 and a transport server 3102 used to establish a VPN with a client computer.

[59] FIG. 32 shows message flows relating to synchronization protocols of FIG. 31.

DETAILED DESCRIPTION OF THE INVENTION

[60] Referring to FIG. 2, a secure mechanism for communicating over the internet employs a number of special routers or servers, called TARP routers 122-127 that are similar to regular IP routers 128-132 in that each has one or more IP addresses and uses normal IP protocol to send normal-looking IP packet messages, called TARP packets 140. TARP packets 140 are identical to normal IP packet messages that are routed by regular IP routers 128-132 because each TARP packet 140 contains a destination address as in a normal IP packet. However, instead of indicating a final destination in the destination field of the IP header, the TARP packet's 140 IP header always points to a next-hop in a series of TARP router hops, or the final destination, TARP terminal 110. Because the header of the TARP packet contains only the next-hop destination, there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet 140 since the destination could always be the next-hop TARP router as well as the final destination, TARP terminal 110.

[61] Each TARP packet's true destination is concealed behind an outer layer of encryption generated using a link key 146. The link key 146 is the encryption key used for encrypted communication between the end points (TARP terminals or TARP routers) of a single link in the chain of hops connecting the originating TARP terminal 100 and the destination TARP terminal 110. Each TARP router 122-127, using the link key 146 it uses to communicate with the previous hop in a chain, can use the link key to reveal the true destination of a TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal (which may indicate the link key used) by the sender field of the clear IP header. Alternatively, this identity may be hidden behind another layer of encryption in available bits in the clear IP header. Each TARP router, upon receiving a TARP message, determines if the message is a TARP message by using authentication data in the TARP packet. This could be recorded in available bytes in the TARP packet's IP header. Alternatively, TARP packets could be authenticated by attempting to decrypt using the link key 146 and determining if the results are as expected. The former may have computational advantages because it does not involve a decryption process.

[62] Once the outer layer of decryption is completed by a TARP router 122-127, the TARP router determines the final destination. The system is preferably designed to cause each TARP packet 140 to undergo a minimum number of hops to help foil traffic analysis. The time to live counter in the IP header of the TARP message may be used to indicate a number of TARP router hops yet to be completed. Each TARP router then would decrement the counter and determine from that whether it should forward the TARP packet 140 to another TARP router 122-127 or to the destination TARP terminal 110. If the time to live counter is zero or below zero after decrementing, for an example of usage, the TARP router receiving the TARP packet 140 may forward the TARP packet 140 to the destination TARP terminal 110. If the time to live counter is above zero after decrementing, for an example of usage, the TARP router receiving the TARP packet 140 may forward the TARP packet 140 to a TARP router 122-127 that the current TARP terminal chooses at random. As a result, each TARP packet 140 is routed through some minimum number of hops of TARP routers 122-127 which are chosen at random.

[63] Thus, each TARP packet, irrespective of the traditional factors determining traffic in the Internet, makes random trips among a number of geographically disparate routers before reaching its destination and each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined as described above. This feature is called *agile routing*. For reasons that will become clear shortly, the fact that different packets take different routes provides distinct advantages by making it difficult for an interloper to obtain all the packets forming an entire multi-packet message. Agile routing is combined with another feature that furthers this purpose, a feature that ensures that any message is broken into multiple packets.

[64] A TARP router receives a TARP packet when an IP address used by the TARP router coincides with the IP address in the TARP packet's IP header IP_C. The IP address of a TARP router, however, may not remain constant. To avoid and manage attacks, each TARP router, independently or under direction from another TARP terminal or router, may change its IP address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs. In reality, whenever a TARP router looks up the address of a destination in the encrypted header, it must convert a TARP address to a real IP address using its LUT.

[65] While every TARP router receiving a TARP packet has the ability to determine the packet's final destination, the message payload is embedded behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the TARP routers 122-127 intervening between the originating 100 and destination 110 TARP terminals. The session key is used to decrypt the payloads of the TARP packets 140 permitting an entire message to be reconstructed.

[66] In one embodiment, communication may be made private using link and session keys, which in turn may be shared and used according any desired method. For example, a public key or symmetric keys may be communicated between link or session endpoints using a public key

method. Any of a variety of other mechanisms for securing data to ensure that only authorized computers can have access to the private information in the TARP packets 140 may be used as desired.

[67] Referring to FIG. 3a, to construct a series of TARP packets, a data stream 300 of IP packets 207a, 207b, 207c, etc., such series of packets being formed by a network (IP) layer process, is broken into a series of small sized segments. In the present example, equal-sized segments 1-9 are defined and used to construct a set of interleaved data packets A, B, and C. Here it is assumed that the number of interleaved packets A, B, and C formed is three and that the number of IP packets 207a-207c used to form the three interleaved packets A, B, and C is exactly three. Of course, the number of IP packets spread over a group of interleaved packets may be any convenient number as may be the number of interleaved packets over which the incoming data stream is spread. The latter, the number of interleaved packets over which the data stream is spread, is called the *interleave window*.

[68] To create a packet, the transmitting software interleaves the normal IP packets 207a *et. seq.* to form a new set of interleaved payload data 320. This payload data 320 is then encrypted using a session key to form a set of session-key-encrypted payload data 330, each of which, A, B, and C, will form the payload of a TARP packet. Using the IP header data, from the original packets 207a-207c, new TARP headers IP_T are formed. The TARP headers IP_T can be identical to normal IP headers or customized in some way. In a preferred embodiment, the TARP headers IP_T are IP headers with added data providing the following information required for routing and reconstruction of messages, some of which data is ordinarily, or capable of being, contained in normal IP headers:

1. A window sequence number – an identifier that indicates where the packet belongs in the original message sequence.
2. An interleave sequence number – an identifier that indicates the interleaving sequence used to form the packet so that the packet can be deinterleaved along with other packets in the interleave window.
3. A time-to-live (TTL) datum – indicates the number of TARP-router-hops to be executed before the packet reaches its destination. Note that the TTL parameter

may provide a datum to be used in a probabilistic formula for determining whether to route the packet to the destination or to another hop.

4. Data type identifier – indicates whether the payload contains, for example, TCP or UDP data.

5. Sender's address – indicates the sender's address in the TARP network.

6. Destination address – indicates the destination terminal's address in the TARP network.

7. Decoy/Real – an indicator of whether the packet contains real message data or dummy decoy data or a combination.

[69] Obviously, the packets going into a single interleave window must include only packets with a common destination. Thus, it is assumed in the depicted example that the IP headers of IP packets 207a-207c all contain the same destination address or at least will be received by the same terminal so that they can be deinterleaved. Note that dummy or decoy data or packets can be added to form a larger interleave window than would otherwise be required by the size of a given message. Decoy or dummy data can be added to a stream to help foil traffic analysis by leveling the load on the network. Thus, it may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

[70] Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to a portion, or the entirety, of a message, and that portion or entirety then interleaved into a number of separate packets.

[71] Referring to FIG. 3b, in an alternative mode of TARP packet construction, a series of IP packets are accumulated to make up a predefined interleave window. The payloads of the packets are used to construct a single block 520 for chain block encryption using the session key.

The payloads used to form the block are presumed to be destined for the same terminal. The block size may coincide with the interleave window as depicted in the example embodiment of FIG. 3b. After encryption, the encrypted block is broken into separate payloads and segments which are interleaved as in the embodiment of Fig 3a. The resulting interleaved packets A, B, and C, are then packaged as TARP packets with TARP headers as in the Example of FIG. 3a. The remaining process is as shown in, and discussed with reference to, FIG. 3a.

[72] Once the TARP packets 340 are formed, each entire TARP packet 340, including the TARP header IP_T, is encrypted using the link key for communication with the first-hop-TARP router. The first hop TARP router is randomly chosen. A final unencrypted IP header IP_C is added to each encrypted TARP packet 340 to form a normal IP packet 360 that can be transmitted to a TARP router. Note that the process of constructing the TARP packet 360 does not have to be done in stages as described. The above description is just a useful heuristic for describing the final product, namely, the TARP packet.

[73] Note that, TARP header IP_T could be a completely custom header configuration with no similarity to a normal IP header except that it contain the information identified above. This is so since this header is interpreted by only TARP routers.

[74] The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system. Referring to FIG. 4, a TARP transceiver 405 can be an originating terminal 100, a destination terminal 110, or a TARP router 122-127. In each TARP Transceiver 405, a transmitting process is generated to receive normal packets from the Network (IP) layer and generate TARP packets for communication over the network. A receiving process is generated to receive normal IP packets containing TARP packets and generate from these normal IP packets which are "passed up" to the Network (IP) layer. Note that where the TARP Transceiver 405 is a router, the received TARP packets 140 are not processed into a stream of IP packets 415 because they need only be authenticated as proper TARP packets and then passed to another TARP router or a TARP destination terminal 110. The intervening process, a "TARP Layer" 420, could be combined with either the data link layer 430 or the Network layer 410. In either case, it would intervene between the data link layer 430 so that the process would receive regular IP packets containing embedded TARP packets and "hand up" a series of reassembled IP packets to the Network layer 410. As an example of combining the TARP layer 420 with the data link layer 430, a program may augment the normal processes running a communications card, for example, an Ethernet card. Alternatively, the TARP layer processes may form part of a dynamically loadable module that is loaded and executed to support communications between the network and data link layers.

[75] Because the encryption system described above can be inserted between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

[76] Note that IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

[77] As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicates that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. To accomplish this, the TARP process will construct a TARP-formatted message, in the style of Internet Control Message Protocol (ICMP) datagrams as an example; this message will contain the machine's TARP address, its previous IP address, and its new IP address. The TARP layer will transmit this packet to at least one known TARP router; then upon

receipt and validation of the message, the TARP router will update its LUT with the new IP address for the stated TARP address. The TARP router will then format a similar message, and broadcast it to the other TARP routers so that they may update their LUTs. Since the total number of TARP routers on any given subnet is expected to be relatively small, this process of updating the LUTs should be relatively fast. It may not, however, work as well when there is a relatively large number of TARP routers and/or a relatively large number of clients; this has motivated a refinement of this architecture to provide scalability; this refinement has led to a second embodiment, which is discussed below.

[78] Upon detection of an attack, the TARP process may also create a subprocess that maintains the original IP address and continues interacting with the attacker. The latter may provide an opportunity to trace the attacker or study the attacker's methods (called "fishbowling" drawing upon the analogy of a small fish in a fish bowl that "thinks" it is in the ocean but is actually under captive observation). A history of the communication between the attacker and the abandoned (fishbowled) IP address can be recorded or transmitted for human analysis or further synthesized for purposes of responding in some way.

[79] As mentioned above, decoy or dummy data or packets can be added to outgoing data streams by TARP terminals or routers. In addition to making it convenient to spread data over a larger number of separate packets, such decoy packets can also help to level the load on inactive portions of the Internet to help foil traffic analysis efforts.

[80] Decoy packets may be generated by each TARP terminal 100, 110 or each router 122-127 on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more likely to insert decoy packets when a message stream is being received. That is, when a series of messages are received, the decoy packet generation rate may be increased. Alternatively, if a large number of decoy packets is received along with regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis. The rate of reception of packets, decoy or otherwise, may be indicated to the decoy packet dropping and generating processes through perishable decoy and regular packet counters. (A perishable counter is one that resets or decrements its value in response to time so that it contains a high value when it is incremented in rapid succession and a small value when incremented either slowly or a small number of times in rapid succession.) Note that destination TARP terminal 110 may generate decoy packets equal in number and size to those TARP packets received to make it appear it is merely routing packets and is therefore not the destination terminal.

[81] Referring to FIG. 5, the following particular steps may be employed in the above-described method for routing TARP packets.

- S0. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.
- S2. The TARP packet may be probed in some way to authenticate the packet before attempting to decrypt it using the link key. That is, the router may determine that the packet is an authentic TARP packet by performing a selected operation on some data included with the clear IP header attached to the encrypted TARP packet contained in the payload. This makes it possible to avoid performing decryption on packets that are not authentic TARP packets.
- S3. The TARP packet is decrypted to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.
- S4. If the packet is a decoy packet, the perishable decoy counter is incremented.
- S5. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the router may choose to throw it away. If the received packet is a decoy packet and it is determined that it should be thrown away (S6), control returns to step S0.

- S7. The TTL parameter of the TARP header is decremented and it is determined if the TTL parameter is greater than zero.
- S8. If the TTL parameter is greater than zero, a TARP address is randomly chosen from a list of TARP addresses maintained by the router and the link key and IP address corresponding to that TARP address memorized for use in creating a new IP packet containing the TARP packet.
- S9. If the TTL parameter is zero or less, the link key and IP address corresponding to the TARP address of the destination are memorized for use in creating the new IP packet containing the TARP packet.
- S10. The TARP packet is encrypted using the memorized link key.
- S11. An IP header is added to the packet that contains the stored IP address, the encrypted TARP packet wrapped with an IP header, and the completed packet transmitted to the next hop or destination.

[82] Referring to FIG. 6, the following particular steps may be employed in the above-described method for generating TARP packets.

- S20. A background loop operation applies an algorithm that determines the generation of decoy IP packets. The loop is interrupted when a data stream containing IP packets is received for transmission.
- S21. The received IP packets are grouped into a set consisting of messages with a constant IP destination address. The set is further broken down to coincide with a maximum size of an interleave window The set is encrypted, and interleaved into a set of payloads destined to become TARP packets.
- S22. The TARP address corresponding to the IP address is determined from a lookup table and stored to generate the TARP header. An initial TTL count is generated and stored in the header. The TTL count may be random with minimum and maximum values or it may be fixed or determined by some other parameter.
- S23. The window sequence numbers and interleave sequence numbers are recorded in the TARP headers of each packet.
- S24. One TARP router address is randomly chosen for each TARP packet and the IP address corresponding to it stored for use in the clear IP header. The link key corresponding to this

router is identified and used to encrypt TARP packets containing interleaved and encrypted data and TARP headers.

- S25. A clear IP header with the first hop router's real IP address is generated and added to each of the encrypted TARP packets and the resulting packets.

[83] Referring to FIG. 7, the following particular steps may be employed in the above-described method for receiving TARP packets.

- S40. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.
- S42. The TARP packet may be probed to authenticate the packet before attempting to decrypt it using the link key.
- S43. The TARP packet is decrypted with the appropriate link key to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.
- S44. If the packet is a decoy packet, the perishable decoy counter is incremented.
- S45. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the receiver may choose to throw it away.
- S46. The TARP packets are cached until all packets forming an interleave window are received.
- S47. Once all packets of an interleave window are received, the packets are deinterleaved.
- S48. The packets block of combined packets defining the interleave window is then decrypted using the session key.
- S49. The decrypted block is then divided using the window sequence data and the IP_T headers are converted into normal IP_C headers. The window sequence numbers are integrated in the IP_C headers.
- S50. The packets are then handed up to the IP layer processes.

1. SCALABILITY ENHANCEMENTS

[84] The IP agility feature described above relies on the ability to transmit IP address changes to all TARP routers. The embodiments including this feature will be referred to as "boutique" embodiments due to potential limitations in scaling these features up for a large network, such as

the Internet. (The "boutique" embodiments would, however, be robust for use in smaller networks, such as small virtual private networks, for example). One problem with the boutique embodiments is that if IP address changes are to occur frequently, the message traffic required to update all routers sufficiently quickly creates a serious burden on the Internet when the TARP router and/or client population gets large. The bandwidth burden added to the networks, for example in ICMP packets, that would be used to update all the TARP routers could overwhelm the Internet for a large scale implementation that approached the scale of the Internet. In other words, the boutique system's scalability is limited.

[85] A system can be constructed which trades some of the features of the above embodiments to provide the benefits of IP agility without the additional messaging burden. This is accomplished by IP address-hopping according to shared algorithms that govern IP addresses used between links participating in communications sessions between nodes such as TARP nodes. (Note that the IP hopping technique is also applicable to the boutique embodiment.) The IP agility feature discussed with respect to the boutique system can be modified so that it becomes decentralized under this scalable regime and governed by the above-described shared algorithm. Other features of the boutique system may be combined with this new type of IP-agility.

[86] The new embodiment has the advantage of providing IP agility governed by a local algorithm and set of IP addresses exchanged by each communicating pair of nodes. This local governance is session-independent in that it may govern communications between a pair of nodes, irrespective of the session or end points being transferred between the directly communicating pair of nodes.

[87] In the scalable embodiments, blocks of IP addresses are allocated to each node in the network. (This scalability will increase in the future, when Internet Protocol addresses are increased to 128-bit fields, vastly increasing the number of distinctly addressable nodes). Each node can thus use any of the IP addresses assigned to that node to communicate with other nodes in the network. Indeed, each pair of communicating nodes can use a plurality of source IP addresses and destination IP addresses for communicating with each other.

[88] Each communicating pair of nodes in a chain participating in any session stores two blocks of IP addresses, called netblocks, and an algorithm and randomization seed for selecting, from each netblock, the next pair of source/destination IP addresses that will be used to transmit the next message. In other words, the algorithm governs the sequential selection of IP-address pairs, one sender and one receiver IP address, from each netblock. The combination of algorithm, seed, and netblock (IP address block) will be called a "hopblock." A router issues separate transmit and receive hopblocks to its clients. The send address and the receive address of the IP header of each outgoing packet sent by the client are filled with the send and receive IP addresses generated by the algorithm. The algorithm is "clocked" (indexed) by a counter so that each time a pair is used, the algorithm turns out a new transmit pair for the next packet to be sent.

[89] The router's receive hopblock is identical to the client's transmit hopblock. The router uses the receive hopblock to predict what the send and receive IP address pair for the next expected packet from that client will be. Since packets can be received out of order, it is not possible for the router to predict with certainty what IP address pair will be on the next sequential packet. To account for this problem, the router generates a range of predictions encompassing the number of possible transmitted packet send/receive addresses, of which the next packet received could leap ahead. Thus, if there is a vanishingly small probability that a given packet will arrive at the router ahead of 5 packets transmitted by the client before the given packet, then the router can generate a series of 6 send/receive IP address pairs (or "hop window") to compare with the next received packet. When a packet is received, it is marked in the hop window as such, so that a second packet with the same IP address pair will be discarded. If an out-of-sequence packet does not arrive within a predetermined timeout period, it can be requested for retransmission or simply discarded from the receive table, depending upon the protocol in use for that communications session, or possibly by convention.

[90] When the router receives the client's packet, it compares the send and receive IP addresses of the packet with the next N predicted send and receive IP address pairs and rejects the packet if it is not a member of this set. Received packets that do not have the predicted source/destination IP addresses falling with the window are rejected, thus thwarting possible hackers. (With the number of possible combinations, even a fairly large window would be hard to fall into at random.) If it is a member of this set, the router accepts the packet and processes it further. This link-based IP-hopping strategy, referred to as "IHOP," is a network element that stands on its own and is not necessarily accompanied by elements of the boutique system described above. If the routing agility feature described in connection with the boutique embodiment is combined with this link-based IP-hopping strategy, the router's next step would be to decrypt the TARP header to determine the destination TARP router for the packet and determine what should be the next hop for the packet. The TARP router would then forward the packet to a random TARP router or the destination TARP router with which the source TARP router has a link-based IP hopping communication established.

[91] Figure 8 shows how a client computer 801 and a TARP router 811 can establish a secure session. When client 801 seeks to establish an IHOP session with TARP router 811, the client 801 sends "secure synchronization" request ("SSYN") packet 821 to the TARP router 811. This SYN packet 821 contains the client's 801 authentication token, and may be sent to the router 811 in an encrypted format. The source and destination IP numbers on the packet 821 are the client's 801 current fixed IP address, and a "known" fixed IP address for the router 811. (For security purposes, it may be desirable to reject any packets from outside of the local network that are destined for the router's known fixed IP address.) Upon receipt and validation of the client's 801 SSYN packet 821, the router 811 responds by sending an encrypted "secure synchronization acknowledgment" ("SSYN ACK") 822 to the client 801. This SSYN ACK 822 will contain the transmit and receive hopblocks that the client 801 will use when communicating with the TARP router 811. The client 801 will acknowledge the TARP router's 811 response packet 822 by generating an encrypted SSYN ACK ACK packet 823 which will be sent from the client's 801 fixed IP address and to the TARP router's 811 known fixed IP address. The client 801 will simultaneously generate a SSYN ACK ACK packet; this SSYN ACK packet, referred to as the Secure Session Initiation (SSI) packet 824, will be sent with the first {sender, receiver} IP pair in the client's transmit table 921 (FIG. 9), as specified in the transmit hopblock provided by the TARP router 811 in the SSYN ACK packet 822. The TARP router 811 will respond to the SSI packet 824 with an SSI ACK packet 825, which will be sent with the first {sender, receiver} IP pair in the TARP router's transmit table 923. Once these packets have been successfully exchanged, the secure communications session is established, and all further secure communications between the client 801 and the TARP router 811 will be conducted via this secure session, as long as synchronization is maintained. If synchronization is lost, then the client 801 and TARP router 802 may re-establish the secure session by the procedure outlined in Figure 8 and described above.

[92] While the secure session is active, both the client 901 and TARP router 911 (FIG. 9) will maintain their respective transmit tables 921, 923 and receive tables 922, 924, as provided by the TARP router during session synchronization 822. It is important that the sequence of IP pairs in the client's transmit table 921 be identical to those in the TARP router's receive table 924; similarly, the sequence of IP pairs in the client's receive table 922 must be identical to those in the router's transmit table 923. This is required for the session synchronization to be maintained. The client 901 need maintain only one transmit table 921 and one receive table 922 during the course of the secure session. Each sequential packet sent by the client 901 will employ the next {send, receive} IP address pair in the transmit table, regardless of TCP or UDP session. The TARP router 911 will expect each packet arriving from the client 901 to bear the next IP address pair shown in its receive table.

[93] Since packets can arrive out of order, however, the router 911 can maintain a "look ahead" buffer in its receive table, and will mark previously-received IP pairs as invalid for future packets; any future packet containing an IP pair that is in the look-ahead buffer but is marked as previously received will be discarded. Communications from the TARP router 911 to the client 901 are maintained in an identical manner; in particular, the router 911 will select the next IP address pair from its transmit table 923 when constructing a packet to send to the client 901, and the client 901 will maintain a look-ahead buffer of expected IP pairs on packets that it is receiving. Each TARP router will maintain separate pairs of transmit and receive tables for each client that is currently engaged in a secure session with or through that TARP router.

[94] While clients receive their hopblocks from the first server linking them to the Internet, routers exchange hopblocks. When a router establishes a link-based IP-hopping communication regime with another router, each router of the pair exchanges its transmit hopblock. The transmit hopblock of each router becomes the receive hopblock of the other router. The communication between routers is governed as described by the example of a client sending a packet to the first router.

[95] While the above strategy works fine in the IP milieu, many local networks that are connected to the Internet are Ethernet systems. In Ethernet, the IP addresses of the destination devices must be translated into hardware addresses, and vice versa, using known processes ("address resolution protocol," and "reverse address resolution protocol"). However, if the link-based IP-hopping strategy is employed, the correlation process would become explosive and burdensome. An alternative to the link-based IP hopping strategy may be employed within an Ethernet network. The solution is to provide that the node linking the Internet to the Ethernet (call it the border node) use the link-based IP-hopping communication regime to communicate with nodes outside the Ethernet LAN. Within the Ethernet LAN, each TARP node would have a single IP address which would be addressed in the conventional way. Instead of comparing the {sender, receiver} IP address pairs to authenticate a packet, the intra-LAN TARP node would use one of the IP header extension fields to do so. Thus, the border node uses an algorithm shared by the intra-LAN TARP node to generate a symbol that is stored in the free field in the IP header, and the intra-LAN TARP node generates a range of symbols based on its prediction of the next expected packet to be received from that particular source IP address. The packet is rejected if it does not fall into the set of predicted symbols (for example, numerical values) or is accepted if it does. Communications from the intra-LAN TARP node to the border node are accomplished in the same manner, though the algorithm will necessarily be different for security reasons. Thus, each of the communicating nodes will generate transmit and receive tables in a similar manner to that of Figure 9; the intra-LAN TARP nodes transmit table will be identical to the border node's receive table, and the intra-LAN TARP node's receive table will be identical to the border node's transmit table.

[96] The algorithm used for IP address-hopping can be any desired algorithm. For example, the algorithm can be a given pseudo-random number generator that generates numbers of the range covering the allowed IP addresses with a given seed. Alternatively, the session participants can assume a certain type of algorithm and specify simply a parameter for applying the

algorithm. For example the assumed algorithm could be a particular pseudo-random number generator and the session participants could simply exchange seed values.

[97] Note that there is no permanent physical distinction between the originating and destination terminal nodes. Either device at either end point can initiate a synchronization of the pair. Note also that the authentication/synchronization-request (and acknowledgment) and hopblock-exchange may all be served by a single message so that separate message exchanges may not be required.

[98] As another extension to the stated architecture, multiple physical paths can be used by a client, in order to provide link redundancy and further thwart attempts at denial of service and traffic monitoring. As shown in Figure 10, for example, client 1001 can establish three simultaneous sessions with each of three TARP routers provided by different ISPs 1011, 1012, 1013. As an example, the client 1001 can use three different telephone lines 1021, 1022, 1023 to connect to the ISPs, or two telephone lines and a cable modem, etc. In this scheme, transmitted packets will be sent in a random fashion among the different physical paths. This architecture provides a high degree of communications redundancy, with improved immunity from denial-of-service attacks and traffic monitoring.

2. FURTHER EXTENSIONS

[99] The following describes various extensions to the techniques, systems, and methods described above. As described above, the security of communications occurring between computers in a computer network (such as the Internet, an Ethernet, or others) can be enhanced by using seemingly random source and destination Internet Protocol (IP) addresses for data packets transmitted over the network. This feature prevents eavesdroppers from determining which computers in the network are communicating with each other while permitting the two communicating computers to easily recognize whether a given received data packet is legitimate or not. In one embodiment of the above-described systems, an IP header extension field is used to authenticate incoming packets on an Ethernet.

[100] Various extensions to the previously described techniques described herein include: (1) use of hopped hardware or "MAC" addresses in broadcast type network; (2) a self-synchronization technique that permits a computer to automatically regain synchronization with a sender; (3) synchronization algorithms that allow transmitting and receiving computers to quickly re-establish synchronization in the event of lost packets or other events; and (4) a fast-packet rejection mechanism for rejecting invalid packets. Any or all of these extensions can be combined with the features described above in any of various ways.

A. Hardware Address Hopping

[101] Internet protocol-based communications techniques on a LAN—or across any dedicated physical medium—typically embed the IP packets within lower-level packets, often referred to as "frames." As shown in FIG. 11, for example, a first Ethernet frame 1150 comprises a frame header 1101 and two embedded IP packets IP1 and IP2, while a second Ethernet frame 1160 comprises a different frame header 1104 and a single IP packet IP3. Each frame header generally includes a source hardware address 1101A and a destination hardware address 1101B; other well-known fields in frame headers are omitted from FIG. 11 for clarity. Two hardware nodes communicating over a physical communication channel insert appropriate source and destination hardware addresses to indicate which nodes on the channel or network should receive the frame.

[102] It may be possible for a nefarious listener to acquire information about the contents of a frame and/or its communicants by examining frames on a local network rather than (or in addition to) the IP packets themselves. This is especially true in broadcast media, such as Ethernet, where it is necessary to insert into the frame header the hardware address of the machine that generated the frame and the hardware address of the machine to which frame is being sent. All nodes on the network can potentially "see" all packets transmitted across the network. This can be a problem for secure communications, especially in cases where the communicants do not want for any third party to be able to identify who is engaging in the information exchange. One way to address this problem is to push the address-hopping scheme down to the hardware layer. In accordance with various embodiments of the invention, hardware addresses are "hopped" in a manner similar to that used to change IP addresses, such that a listener cannot determine which hardware node generated a particular message nor which node is the intended recipient.

[103] FIG. 12A shows a system in which Media Access Control ("MAC") hardware addresses are "hopped" in order to increase security over a network such as an Ethernet. While the description refers to the exemplary case of an Ethernet environment, the inventive principles are equally applicable to other types of communications media. In the Ethernet case, the MAC address of the sender and receiver are inserted into the Ethernet frame and can be observed by anyone on the LAN who is within the broadcast range for that frame. For secure communications, it becomes desirable to generate frames with MAC addresses that are not attributable to any specific sender or receiver.

[104] As shown in FIG. 12A, two computer nodes 1201 and 1202 communicate over a communication channel such as an Ethernet. Each node executes one or more application programs 1203 and 1218 that communicate by transmitting packets through communication software 1204 and 1217, respectively. Examples of application programs include video conferencing, e-mail, word processing programs, telephony, and the like. Communication software 1204 and 1217 can comprise, for example, an OSI layered architecture or "stack" that standardizes various services provided at different levels of functionality.

[105] The lowest levels of communication software 1204 and 1217 communicate with hardware components 1206 and 1214 respectively, each of which can include one or more registers 1207 and 1215 that allow the hardware to be reconfigured or controlled in accordance with various communication protocols. The hardware components (an Ethernet network interface card, for example) communicate with each other over the communication medium. Each hardware component is typically pre-assigned a fixed hardware address or MAC number that identifies the hardware component to other nodes on the network. One or more interface drivers control the operation of each card and can, for example, be configured to accept or reject packets from certain hardware addresses. As will be described in more detail below, various embodiments of the inventive principles provide for "hopping" different addresses using one or more algorithms and one or more moving windows that track a range of valid addresses to validate received packets. Packets transmitted according to one or more of the inventive principles will be generally referred to as "secure" packets or "secure communications" to

differentiate them from ordinary data packets that are transmitted in the clear using ordinary, machine-correlated addresses.

[106] One straightforward method of generating non-attributable MAC addresses is an extension of the IP hopping scheme. In this scenario, two machines on the same LAN that desire to communicate in a secure fashion exchange random-number generators and seeds, and create sequences of quasi-random MAC addresses for synchronized hopping. The implementation and synchronization issues are then similar to that of IP hopping.

[107] This approach, however, runs the risk of using MAC addresses that are currently active on the LAN—which, in turn, could interrupt communications for those machines. Since an Ethernet MAC address is at present 48 bits in length, the chance of randomly misusing an active MAC address is actually quite small. However, if that figure is multiplied by a large number of nodes (as would be found on an extensive LAN), by a large number of frames (as might be the case with packet voice or streaming video), and by a large number of concurrent Virtual Private Networks (VPNs), then the chance that a non-secure machine's MAC address could be used in an address-hopped frame can become non-trivial. In short, any scheme that runs even a small risk of interrupting communications for other machines on the LAN is bound to receive resistance from prospective system administrators. Nevertheless, it is technically feasible, and can be implemented without risk on a LAN on which there is a small number of machines, or if all of the machines on the LAN are engaging in MAC-hopped communications.

[108] Synchronized MAC address hopping may incur some overhead in the course of session establishment, especially if there are multiple sessions or multiple nodes involved in the communications. A simpler method of randomizing MAC addresses is to allow each node to receive and process *every* incident frame on the network. Typically, each network interface driver will check the destination MAC address in the header of every incident frame to see if it matches that machine's MAC address; if there is no match, then the frame is discarded. In one embodiment, however, these checks can be disabled, and every incident packet is passed to the TARP stack for processing. This will be referred to as "promiscuous" mode, since every incident frame is processed. Promiscuous mode allows the sender to use completely random, unsynchronized MAC addresses, since the destination machine is guaranteed to process the frame. The decision as to whether the packet was truly intended for that machine is handled by the TARP stack, which checks the source and destination IP addresses for a match in its IP synchronization tables. If no match is found, the packet is discarded; if there is a match, the packet is unwrapped, the inner header is evaluated, and if the inner header indicates that the packet is destined for that machine then the packet is forwarded to the IP stack—otherwise it is discarded.

[109] One disadvantage of purely-random MAC address hopping is its impact on processing overhead; that is, since every incident frame must be processed, the machine's CPU is engaged considerably more often than if the network interface driver is discriminating and rejecting packets unilaterally. A compromise approach is to select either a single fixed MAC address or a small number of MAC addresses (e.g., one for each virtual private network on an Ethernet) to use for MAC-hopped communications, regardless of the actual recipient for which the message is intended. In this mode, the network interface driver can check each incident frame against one (or a few) pre-established MAC addresses, thereby freeing the CPU from the task of physical-layer packet discrimination. This scheme does not betray any useful information to an interloper on the LAN; in particular, every secure packet can already be identified by a unique packet type in the outer header. However, since all machines engaged in secure communications would either be using the same MAC address, or be selecting from a small pool of predetermined MAC addresses, the association between a specific machine and a specific MAC address is effectively broken.

[110] In this scheme, the CPU will be engaged more often than it would be in non-secure communications (or in synchronized MAC address hopping), since the network interface driver cannot always unilaterally discriminate between secure packets that are destined for that machine, and secure packets from other VPNs. However, the non-secure traffic is easily eliminated at the network interface, thereby reducing the amount of processing required of the CPU. There are boundary conditions where these statements would not hold, of course—e.g., if *all* of the traffic on the LAN is secure traffic, then the CPU would be engaged to the same degree as it is in the purely-random address hopping case; alternatively, if each VPN on the LAN uses a different MAC address, then the network interface can perfectly discriminate secure frames destined for the local machine from those constituting other VPNs. These are engineering tradeoffs that might be best handled by providing administrative options for the users when installing the software and/or establishing VPNs.

[111] Even in this scenario, however, there still remains a slight risk of selecting MAC addresses that are being used by one or more nodes on the LAN. One solution to this problem is to formally assign one address or a range of addresses for use in MAC-hopped communications. This is typically done via an assigned numbers registration authority; e.g., in the case of Ethernet, MAC address ranges are assigned to vendors by the Institute of Electrical and Electronics Engineers (IEEE). A formally-assigned range of addresses would ensure that secure frames do not conflict with any properly-configured and properly-functioning machines on the LAN.

[112] Reference will now be made to FIGS. 12A and 12B in order to describe the many combinations and features that follow the inventive principles. As explained above, two computer nodes 1201 and 1202 are assumed to be communicating over a network or communication medium such as an Ethernet. A communication protocol in each node (1204 and 1217, respectively) contains a modified element 1205 and 1216 that performs certain functions that deviate from the standard communication protocols. In particular, computer node 1201 implements a first "hop" algorithm 1208X that selects seemingly random source and destination IP addresses (and, in one embodiment, seemingly random IP header discriminator fields) in order to transmit each packet to the other computer node. For example, node 1201 maintains a transmit table 1208 containing triplets of source (S), destination (D), and discriminator fields (DS) that are inserted into outgoing IP packet headers. The table is generated through the use of an appropriate algorithm (e.g., a random number generator that is seeded with an appropriate seed) that is known to the recipient node 1202. As each new IP packet is formed, the next sequential entry out of the sender's transmit table 1208 is used to populate the IP source, IP destination, and IP header extension field (e.g., discriminator field). It will be appreciated that the transmit table need not be created in advance but could instead be created on-the-fly by executing the algorithm when each packet is formed.

[113] At the receiving node 1202, the same IP hop algorithm 1222X is maintained and used to generate a receive table 1222 that lists valid triplets of source IP address, destination IP address, and discriminator field. This is shown by virtue of the first five entries of transmit table 1208 matching the second five entries of receive table 1222. (The tables may be slightly offset at any particular time due to lost packets, misordered packets, or transmission delays). Additionally, node 1202 maintains a receive window W3 that represents a list of valid IP source, IP destination, and discriminator fields that will be accepted when received as part of an incoming IP packet. As packets are received, window W3 slides down the list of valid entries, such that the possible valid entries change over time. Two packets that arrive out of order but are nevertheless matched to entries within window W3 will be accepted; those falling outside of window W3 will be rejected as invalid. The length of window W3 can be adjusted as necessary to reflect network delays or other factors.

[114] Node 1202 maintains a similar transmit table 1221 for creating IP packets and frames destined for node 1201 using a potentially different hopping algorithm 1221X, and node 1201 maintains a matching receive table 1209 using the same algorithm 1209X. As node 1202 transmits packets to node 1201 using seemingly random IP source, IP destination, and/or discriminator fields, node 1201 matches the incoming packet values to those falling within window W1 maintained in its receive table. In effect, transmit table 1208 of node 1201 is synchronized (i.e., entries are selected in the same order) to receive table 1222 of receiving node 1202. Similarly, transmit table 1221 of node 1202 is synchronized to receive table 1209 of node 1201. It will be appreciated that although a common algorithm is shown for the source, destination and discriminator fields in FIG. 12A (using, e.g., a different seed for each of the three fields), an entirely different algorithm could in fact be used to establish values for each of these fields. It will also be appreciated that one or two of the fields can be "hopped" rather than all three as illustrated.

[115] In accordance with another aspect of the invention, hardware or "MAC" addresses are hopped instead of or in addition to IP addresses and/or the discriminator field in order to improve security in a local area or broadcast-type network. To that end, node 1201 further maintains a transmit table 1210 using a transmit algorithm 1210X to generate source and destination

hardware addresses that are inserted into frame headers (e.g., fields 1101A and 1101B in FIG. 11) that are synchronized to a corresponding receive table 1224 at node 1202. Similarly, node 1202 maintains a different transmit table 1223 containing source and destination hardware addresses that is synchronized with a corresponding receive table 1211 at node 1201. In this manner, outgoing hardware frames appear to be originating from and going to completely random nodes on the network, even though each recipient can determine whether a given packet is intended for it or not. It will be appreciated that the hardware hopping feature can be implemented at a different level in the communications protocol than the IP hopping feature (e.g., in a card driver or in a hardware card itself to improve performance).

[116] FIG. 12B shows three different embodiments or modes that can be employed using the aforementioned principles. In a first mode referred to as "promiscuous" mode, a common hardware address (e.g., a fixed address for source and another for destination) or else a completely random hardware address is used by all nodes on the network, such that a particular packet cannot be attributed to any one node. Each node must initially accept all packets containing the common (or random) hardware address and inspect the IP addresses or discriminator field to determine whether the packet is intended for that node. In this regard, either the IP addresses or the discriminator field or both can be varied in accordance with an algorithm as described above. As explained previously, this may increase each node's overhead since additional processing is involved to determine whether a given packet has valid source and destination hardware addresses.

[117] In a second mode referred to as "promiscuous per VPN" mode, a small set of fixed hardware addresses are used, with a fixed source/destination hardware address used for all nodes communicating over a virtual private network. For example, if there are six nodes on an Ethernet, and the network is to be split up into two private virtual networks such that nodes on one VPN can communicate with only the other two nodes on its own VPN, then two sets of hardware addresses could be used: one set for the first VPN and a second set for the second VPN. This would reduce the amount of overhead involved in checking for valid frames since only packets arriving from the designated VPN would need to be checked. IP addresses and one or more discriminator fields could still be hopped as before for secure communication within the

VPN. Of course, this solution compromises the anonymity of the VPNs (i.e., an outsider can easily tell what traffic belongs in which VPN, though he cannot correlate it to a specific machine/person). It also requires the use of a discriminator field to mitigate the vulnerability to certain types of DoS attacks. (For example, without the discriminator field, an attacker on the LAN could stream frames containing the MAC addresses being used by the VPN; rejecting those frames could lead to excessive processing overhead. The discriminator field would provide a low-overhead means of rejecting the false packets.)

[118] In a third mode referred to as "hardware hopping" mode, hardware addresses are varied as illustrated in FIG. 12A, such that hardware source and destination addresses are changed constantly in order to provide non-attributable addressing. Variations on these embodiments are of course possible, and the invention is not intended to be limited in any respect by these illustrative examples.

B. Extending the Address Space

[119] Address hopping provides security and privacy. However, the level of protection is limited by the number of addresses in the blocks being hopped. A hopblock denotes a field or fields modulated on a packet-wise basis for the purpose of providing a VPN. For instance, if two nodes communicate with IP address hopping using hopblocks of 4 addresses (2 bits) each, there would be 16 possible address-pair combinations. A window of size 16 would result in most address pairs being accepted as valid most of the time. This limitation can be overcome by using a discriminator field in addition to or instead of the hopped address fields. The discriminator field would be hopped in exactly the same fashion as the address fields and it would be used to determine whether a packet should be processed by a receiver.

[120] Suppose that two clients, each using four-bit hopblocks, would like the same level of protection afforded to clients communicating via IP hopping between two A blocks (24 address bits eligible for hopping). A discriminator field of 20 bits, used in conjunction with the 4 address bits eligible for hopping in the IP address field, provides this level of protection. A 24-bit discriminator field would provide a similar level of protection if the address fields were not hopped or ignored. Using a discriminator field offers the following advantages: (1) an arbitrarily high level of protection can be provided, and (2) address hopping is unnecessary to provide protection. This may be important in environments where address hopping would cause routing problems.

C. Synchronization Techniques

[121] It is generally assumed that once a sending node and receiving node have exchanged algorithms and seeds (or similar information sufficient to generate quasi-random source and destination tables), subsequent communication between the two nodes will proceed smoothly. Realistically, however, two nodes may lose synchronization due to network delays or outages, or other problems. Consequently, it is desirable to provide means for re-establishing synchronization between nodes in a network that have lost synchronization.

[122] One possible technique is to require that each node provide an acknowledgment upon successful receipt of each packet and, if no acknowledgment is received within a certain period of time, to re-send the unacknowledged packet. This approach, however, drives up overhead costs and may be prohibitive in high-throughput environments such as streaming video or audio, for example.

[123] A different approach is to employ an automatic synchronizing technique that will be referred to herein as "self-synchronization." In this approach, synchronization information is embedded into each packet, thereby enabling the receiver to re-synchronize itself upon receipt of a single packet if it determines that is has lost synchronization with the sender. (If communications are already in progress, and the receiver determines that it is still in sync with the sender, then there is no need to re-synchronize.) A receiver could detect that it was out of synchronization by, for example, employing a "dead-man" timer that expires after a certain period of time, wherein the timer is reset with each valid packet. A time stamp could be hashed into the public sync field (see below) to preclude packet-retry attacks.

[124] In one embodiment, a "sync field" is added to the header of each packet sent out by the sender. This sync field could appear in the clear or as part of an encrypted portion of the packet. Assuming that a sender and receiver have selected a random-number generator (RNG) and seed value, this combination of RNG and seed can be used to generate a random-number sequence

(RNS). The RNS is then used to generate a sequence of source/destination IP pairs (and, if desired, discriminator fields and hardware source and destination addresses), as described above. It is not necessary, however, to generate the entire sequence (or the first N-1 values) in order to generate the Nth random number in the sequence; if the sequence index N is known, the random value corresponding to that index can be directly generated (see below). Different RNGs (and seeds) with different fundamental periods could be used to generate the source and destination IP sequences, but the basic concepts would still apply. For the sake of simplicity, the following discussion will assume that IP source and destination address pairs (only) are hopped using a single RNG sequencing mechanism.

[125] In accordance with a "self-synchronization" feature, a sync field in each packet header provides an index (i.e., a sequence number) into the RNS that is being used to generate IP pairs. Plugging this index into the RNG that is being used to generate the RNS yields a specific random number value, which in turn yields a specific IP pair. That is, an IP pair can be generated directly from knowledge of the RNG, seed, and index number; it is not necessary, in this scheme, to generate the entire sequence of random numbers that precede the sequence value associated with the index number provided.

[126] Since the communicants have presumably previously exchanged RNGs and seeds, the only new information that must be provided in order to generate an IP pair is the sequence number. If this number is provided by the sender in the packet header, then the receiver need only plug this number into the RNG in order to generate an IP pair – and thus verify that the IP pair appearing in the header of the packet is valid. In this scheme, if the sender and receiver lose synchronization, the receiver can immediately re-synchronize upon receipt of a single packet by simply comparing the IP pair in the packet header to the IP pair generated from the index number. Thus, synchronized communications can be resumed upon receipt of a single packet, making this scheme ideal for multicast communications. Taken to the extreme, it could obviate the need for synchronization tables entirely; that is, the sender and receiver could simply rely on the index number in the sync field to validate the IP pair on each packet, and thereby eliminate the tables entirely.

[127] The aforementioned scheme may have some inherent security issues associated with it — namely, the placement of the sync field. If the field is placed in the outer header, then an interloper could observe the values of the field and their relationship to the IP stream. This could potentially compromise the algorithm that is being used to generate the IP-address sequence, which would compromise the security of the communications. If, however, the value is placed in the inner header, then the sender must decrypt the inner header before it can extract the sync value and validate the IP pair; this opens up the receiver to certain types of denial-of-service (DoS) attacks, such as packet replay. That is, if the receiver must decrypt a packet before it can validate the IP pair, then it could potentially be forced to expend a significant amount of processing on decryption if an attacker simply retransmits previously valid packets. Other attack methodologies are possible in this scenario.

[128] A possible compromise between algorithm security and processing speed is to split up the sync value between an inner (encrypted) and outer (unencrypted) header. That is, if the sync value is sufficiently long, it could potentially be split into a rapidly-changing part that can be viewed in the clear, and a fixed (or very slowly changing) part that must be protected. The part that can be viewed in the clear will be called the "public sync" portion and the part that must be protected will be called the "private sync" portion.

[129] Both the public sync and private sync portions are needed to generate the complete sync value. The private portion, however, can be selected such that it is fixed or will change only occasionally. Thus, the private sync value can be stored by the recipient, thereby obviating the need to decrypt the header in order to retrieve it. If the sender and receiver have previously agreed upon the frequency with which the private part of the sync will change, then the receiver can selectively decrypt a single header in order to extract the new private sync if the communications gap that has led to lost synchronization has exceeded the lifetime of the previous private sync. This should not represent a burdensome amount of decryption, and thus should not open up the receiver to denial-of-service attack simply based on the need to occasionally decrypt a single header.

[130] One implementation of this is to use a hashing function with a one-to-one mapping to generate the private and public sync portions from the sync value. This implementation is shown

in FIG. 13, where (for example) a first ISP 1302 is the sender and a second ISP 1303 is the receiver. (Other alternatives are possible from FIG. 13.) A transmitted packet comprises a public or "outer" header 1305 that is not encrypted, and a private or "inner" header 1306 that is encrypted using for example a link key. Outer header 1305 includes a public sync portion while inner header 1306 contains the private sync portion. A receiving node decrypts the inner header using a decryption function 1307 in order to extract the private sync portion. This step is necessary only if the lifetime of the currently buffered private sync has expired. (If the currently-buffered private sync is still valid, then it is simply extracted from memory and "added" (which could be an inverse hash) to the public sync, as shown in step 1308.) The public and decrypted private sync portions are combined in function 1308 in order to generate the combined sync 1309. The combined sync (1309) is then fed into the RNG (1310) and compared to the IP address pair (1311) to validate or reject the packet.

[131] An important consideration in this architecture is the concept of "future" and "past" where the public sync values are concerned. Though the sync values, themselves, should be random to prevent spoofing attacks, it may be important that the receiver be able to quickly identify a sync value that has already been sent — even if the packet containing that sync value was never actually received by the receiver. One solution is to hash a time stamp or sequence number into the public sync portion, which could be quickly extracted, checked, and discarded, thereby validating the public sync portion itself.

[132] In one embodiment, packets can be checked by comparing the source/destination IP pair generated by the sync field with the pair appearing in the packet header. If (1) they match, (2) the time stamp is valid, and (3) the dead-man timer has expired, then re-synchronization occurs; otherwise, the packet is rejected. If enough processing power is available, the dead-man timer and synchronization tables can be avoided altogether, and the receiver would simply resynchronize (e.g., validate) on every packet.

[133] The foregoing scheme may require large-integer (e.g., 160-bit) math, which may affect its implementation. Without such large-integer registers, processing throughput would be affected, thus potentially affecting security from a denial-of-service standpoint. Nevertheless, as large-integer math processing features become more prevalent, the costs of implementing such a feature will be reduced.

D. Other Synchronization Schemes

[134] As explained above, if W or more consecutive packets are lost between a transmitter and receiver in a VPN (where W is the window size), the receiver's window will not have been updated and the transmitter will be transmitting packets not in the receiver's window. The sender and receiver will not recover synchronization until perhaps the random pairs in the window are repeated by chance. Therefore, there is a need to keep a transmitter and receiver in synchronization whenever possible and to re-establish synchronization whenever it is lost.

[135] A "checkpoint" scheme can be used to regain synchronization between a sender and a receiver that have fallen out of synchronization. In this scheme, a checkpoint message comprising a random IP address pair is used for communicating synchronization information. In one embodiment, two messages are used to communicate synchronization information between a sender and a recipient:

1. SYNC_REQ is a message used by the sender to indicate that it wants to synchronize; and
2. SYNC_ACK is a message used by the receiver to inform the transmitter that it has been synchronized.

[136] According to one variation of this approach, both the transmitter and receiver maintain three checkpoints (see FIG. 14):

1. In the transmitter, ckpt_o ("checkpoint old") is the IP pair that was used to re-send the last SYNC_REQ packet to the receiver. In the receiver, ckpt_o ("checkpoint old") is the IP pair that receives repeated SYNC_REQ packets from the transmitter.
2. In the transmitter, ckpt_n ("checkpoint new") is the IP pair that will be used to send the next SYNC_REQ packet to the receiver. In the receiver, ckpt_n ("checkpoint new") is the IP pair that receives a new SYNC_REQ packet from the transmitter and which causes the receiver's window to be re-aligned, ckpt_o set to ckpt_n, a new ckpt_n to be generated and a new ckpt_r to be generated.

3. In the transmitter, ckpt_r is the IP pair that will be used to send the next SYNC_ACK packet to the receiver. In the receiver, ckpt_r is the IP pair that receives a new SYNC_ACK packet from the transmitter and which causes a new ckpt_n to be generated. Since SYNC_ACK is transmitted from the receiver ISP to the sender ISP, the transmitter ckpt_r refers to the ckpt_r of the receiver and the receiver ckpt_r refers to the ckpt_r of the transmitter (see FIG. 14).

[137] When a transmitter initiates synchronization, the IP pair it will use to transmit the next data packet is set to a predetermined value and when a receiver first receives a SYNC_REQ, the receiver window is updated to be centered on the transmitter's next IP pair. This is the primary mechanism for checkpoint synchronization.

[138] Synchronization can be initiated by a packet counter (e.g., after every N packets transmitted, initiate a synchronization) or by a timer (every S seconds, initiate a synchronization) or a combination of both. See FIG. 15. From the transmitter's perspective, this technique operates as follows: (1) Each transmitter periodically transmits a "sync request" message to the receiver to make sure that it is in sync. (2) If the receiver is still in sync, it sends back a "sync ack" message. (If this works, no further action is necessary). (3) If no "sync ack" has been received within a period of time, the transmitter retransmits the sync request again. If the transmitter reaches the next checkpoint without receiving a "sync ack" response, then synchronization is broken, and the transmitter should stop transmitting. The transmitter will continue to send sync_reqs until it receives a sync_ack , at which point transmission is reestablished.

[139] From the receiver's perspective, the scheme operates as follows: (1) when it receives a "sync request" request from the transmitter, it advances its window to the next checkpoint position (even skipping pairs if necessary), and sends a "sync ack" message to the transmitter. If sync was never lost, then the "jump ahead" really just advances to the next available pair of addresses in the table (i.e., normal advancement).

[140] If an interloper intercepts the "sync request" messages and tries to interfere with communication by sending new ones, it will be ignored if the synchronization has been established or it it will actually help to re-establish synchronization.

[141] A window is realigned whenever a re-synchronization occurs. This realignment entails updating the receiver's window to straddle the address pairs used by the packet transmitted immediately after the transmission of the SYNC_REQ packet. Normally, the transmitter and receiver are in synchronization with one another. However, when network events occur, the receiver's window may have to be advanced by many steps during resynchronization. In this case, it is desirable to move the window ahead without having to step through the intervening random numbers sequentially. (This feature is also desirable for the auto-sync approach discussed above).

E. Random Number Generator with a Jump-Ahead capability

[142] An attractive method for generating randomly hopped addresses is to use identical random number generators in the transmitter and receiver and advance them as packets are transmitted and received. There are many random number generation algorithms that could be used. Each one has strengths and weaknesses for address hopping applications.

[143] Linear congruential random number generators (LCRs) are fast, simple and well characterized random number generators that can be made to jump ahead *n* steps efficiently. An LCR generates random numbers $X_1, X_2, X_3 \ldots X_k$ starting with seed X_0 using a recurrence

$$X_i=(a\, X_{i-1} + b) \bmod c \qquad (1)$$

where a, b and c define a particular LCR. Another expression for X_i,

$$X_i=((a^i(X_0+b)-b)/(a-1)) \bmod c \qquad (2)$$

enables the jump-ahead capability. The factor a^i can grow very large even for modest i if left unfettered. Therefore some special properties of the modulo operation can be used to control the size and processing time required to compute (2). (2) can be rewritten as:

$$X_i=(a^i\,(X_0(a-1)+b)-b)/(a-1) \bmod c \qquad (3)$$

[144] It can be shown that:

$$(a^i(X_0(a-1)+b)-b)/(a-1) \bmod c = ((a^i \bmod((a-1)c)(X_0(a-1)+b) - b)/(a-1)) \bmod c \qquad (4)$$

$(X_0(a-1)+b)$ can be stored as $(X_0(a-1)+b) \bmod c$, b as b mod c and compute $a^i \bmod((a-1)c)$ (this requires O(log(i)) steps).

[145] A practical implementation of this algorithm would jump a fixed distance, n, between synchronizations; this is tantamount to synchronizing every *n* packets. The window would commence *n* IP pairs from the start of the previous window. Using X_j^w, the random number at the j^{th} checkpoint, as X_0 and *n* as *i*, a node can store $a^n \bmod((a-1)c)$ once per LCR and set

$$X_{j+1}^w = X_{n(j+1)} = ((a^n \bmod((a-1)c)\ (X_j^w\ (a-1)+b)-b)/(a-1)) \bmod c, \qquad (5)$$

to generate the random number for the $j+1^{th}$ synchronization. Using this construction, a node could jump ahead an arbitrary (but fixed) distance between synchronizations in a constant amount of time (independent of n).

[146] Pseudo-random number generators, in general, and LCRs, in particular, will eventually repeat their cycles. This repetition may present vulnerability in the IP hopping scheme. An adversary would simply have to wait for a repeat to predict future sequences. One way of coping with this vulnerability is to create a random number generator with a known long cycle. A random sequence can be replaced by a new random number generator before it repeats. LCRs can be constructed with known long cycles. This is not currently true of many random number generators.

[147] Random number generators can be cryptographically insecure. An adversary can derive the RNG parameters by examining the output or part of the output. This is true of LCGs. This vulnerability can be mitigated by incorporating an encryptor, designed to scramble the output as part of the random number generator. The random number generator prevents an adversary from mounting an attack—e.g., a known plaintext attack—against the encryptor.

F. Random Number Generator Example

[148] Consider a RNG where a=31,b=4 and c=15. For this case equation (1) becomes:

$$X_i = (31\ X_{i-1} + 4) \bmod 15. \qquad (6)$$

[149] If one sets $X_0=1$, equation (6) will produce the sequence 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 0, 4, 8, 12. This sequence will repeat indefinitely. For a jump ahead of 3 numbers in this sequence $a^n= 31^3=29791$, $c*(a-1)=15*30=450$ and $a^n \bmod((a-1)c) = 31^3 \bmod(15*30)=29791 \bmod(450)=91$. Equation (5) becomes:

$$((91\ (X_i 30+4)-4)/30) \bmod 15 \qquad (7)$$

[150] Table 1 shows the jump ahead calculations from (7) . The calculations start at 5 and jump ahead 3.

TABLE 1

I	X_i	(X_i30+4)	$91\ (X_i30+4)-4$	$((91\ (X_i30+4)-4)/30$	X_{i+3}
1	5	154	14010	467	2
4	2	64	5820	194	14
7	14	424	38580	1286	11
10	11	334	30390	1013	8
13	8	244	22200	740	5

G. Fast Packet Filter

[151] Address hopping VPNs must rapidly determine whether a packet has a valid header and thus requires further processing, or has an invalid header (a hostile packet) and should be immediately rejected. Such rapid determinations will be referred to as "fast packet filtering." This capability protects the VPN from attacks by an adversary who streams hostile packets at the receiver at a high rate of speed in the hope of saturating the receiver's processor (a so-called "denial of service" attack). Fast packet filtering is an important feature for implementing VPNs on shared media such as Ethernet.

[152] Assuming that all participants in a VPN share an unassigned "A" block of addresses, one possibility is to use an experimental "A" block that will never be assigned to any machine that is not address hopping on the shared medium. "A" blocks have a 24 bits of address that can be hopped as opposed to the 8 bits in "C" blocks. In this case a hopblock will be the "A" block. The use of the experimental "A" block is a likely option on an Ethernet because:

1. The addresses have no validity outside of the Ethernet and will not be routed out to a valid outside destination by a gateway.
2. There are 2^{24} (~16 million) addresses that can be hopped within each "A" block. This yields >280 trillion possible address pairs making it very unlikely that an adversary would guess a valid address. It also provides acceptably low probability of collision between separate VPNs (all VPNs on a shared medium independently generate random address pairs from the same "A" block).
3. The packets will not be received by someone on the Ethernet who is not on a VPN (unless the machine is in promiscuous mode) minimizing impact on non-VPN computers.

[153] The Ethernet example will be used to describe one implementation of fast packet filtering. The ideal algorithm would quickly examine a packet header, determine whether the packet is hostile, and reject any hostile packets or determine which active IP pair the packet header matches. The problem is a classical associative memory problem. A variety of techniques have been developed to solve this problem (hashing, B–trees etc). Each of these approaches has its strengths and weaknesses. For instance, hash tables can be made to operate quite fast in a statistical sense, but can occasionally degenerate into a much slower algorithm. This slowness can persist for a period of time. Since there is a need to discard hostile packets quickly at all times, hashing would be unacceptable.

H. Presence Vector Algorithm

[154] A presence vector is a bit vector of length 2^n that can be indexed by n-bit numbers (each ranging from 0 to 2^n-1). One can indicate the presence of k n-bit numbers (not necessarily unique), by setting the bits in the presence vector indexed by each number to 1. Otherwise, the bits in the presence vector are 0. An n-bit number, x, is one of the k numbers if and only if the x^{th} bit of the presence vector is 1. A fast packet filter can be implemented by indexing the presence vector and looking for a 1, which will be referred to as the "test."

[155] For example, suppose one wanted to represent the number 135 using a presence vector. The 135^{th} bit of the vector would be set. Consequently, one could very quickly determine whether an address of 135 was valid by checking only one bit: the 135^{th} bit. The presence vectors could be created in advance corresponding to the table entries for the IP addresses. In effect, the incoming addresses can be used as indices into a long vector, making comparisons very fast. As each RNG generates a new address, the presence vector is updated to reflect the information. As the window moves, the presence vector is updated to zero out addresses that are no longer valid.

[156] There is a trade-off between efficiency of the test and the amount of memory required for storing the presence vector(s). For instance, if one were to use the 48 bits of hopping addresses as an index, the presence vector would have to be 35 terabytes. Clearly, this is too large for practical purposes. Instead, the 48 bits can be divided into several smaller fields. For instance, one could subdivide the 48 bits into four 12-bit fields (see FIG. 16). This reduces the storage requirement to 2048 bytes at the expense of occasionally having to process a hostile packet. In effect, instead of one long presence vector, the decomposed address portions must match all four shorter presence vectors before further processing is allowed. (If the first part of the address portion doesn't match the first presence vector, there is no need to check the remaining three presence vectors).

[157] A presence vector will have a 1 in the y^{th} bit if and only if one or more addresses with a corresponding field of y are active. An address is active only if each presence vector indexed by the appropriate sub-field of the address is 1.

[158] Consider a window of 32 active addresses and 3 checkpoints. A hostile packet will be rejected by the indexing of one presence vector more than 99% of the time. A hostile packet will be rejected by the indexing of all 4 presence vectors more than 99.9999995% of the time. On average, hostile packets will be rejected in less than 1.02 presence vector index operations.

[159] The small percentage of hostile packets that pass the fast packet filter will be rejected when matching pairs are not found in the active window or are active checkpoints. Hostile packets that serendipitously match a header will be rejected when the VPN software attempts to decrypt the header. However, these cases will be extremely rare. There are many other ways this method can be configured to arbitrate the space/speed tradeoffs.

I. <u>Further Synchronization Enhancements</u>

[160] A slightly modified form of the synchronization techniques described above can be employed. The basic principles of the previously described checkpoint synchronization scheme remain unchanged. The actions resulting from the reception of the checkpoints are, however, slightly different. In this variation, the receiver will maintain between OoO ("Out of Order") and 2×WINDOW_SIZE+OoO active addresses (1 ≤OoO ≤WINDOW_SIZE and WINDOW_SIZE ≥1). OoO and WINDOW_SIZE are engineerable parameters, where OoO is the minimum number of addresses needed to accommodate lost packets due to events in the network or out of order arrivals and WINDOW_SIZE is the number of packets transmitted before a SYNC_REQ is issued. FIG. 17 depicts a storage array for a receiver's active addresses.

[161] The receiver starts with the first 2×WINDOW_SIZE addresses loaded and active (ready to receive data). As packets are received, the corresponding entries are marked as "used" and are no longer eligible to receive packets. The transmitter maintains a packet counter, initially set to 0, containing the number of data packets transmitted since the last *initial* transmission of a SYNC_REQ for which SYNC_ACK has been received. When the transmitter packet counter equals WINDOW_SIZE, the transmitter generates a SYNC_REQ and does its initial transmission. When the receiver receives a SYNC_REQ corresponding to its current CKPT_N, it generates the next WINDOW_SIZE addresses and starts loading them in order starting at the first location after the last active address wrapping around to the beginning of the array after the end of the array has been reached. The receiver's array might look like FIG. 18 when a SYNC_REQ has been received. In this case a couple of packets have been either lost or will be received out of order when the SYNC_REQ is received.

[162] FIG. 19 shows the receiver's array after the new addresses have been generated. If the transmitter does not receive a SYNC_ACK, it will re-issue the SYNC_REQ at regular intervals. When the transmitter receives a SYNC_ACK, the packet counter is decremented by WINDOW_SIZE. If the packet counter reaches 2×WINDOW_SIZE – OoO then the transmitter ceases sending data packets until the appropriate SYNC_ACK is finally received. The transmitter then resumes sending data packets. Future behavior is essentially a repetition of this initial cycle. The advantages of this approach are:

1. There is no need for an efficient jump ahead in the random number generator,

2. No packet is ever transmitted that does not have a corresponding entry in the receiver side
3. No timer based re-synchronization is necessary. This is a consequence of 2.
4. The receiver will always have the ability to accept data messages transmitted within OoO messages of the most recently transmitted message.

J. Distributed Transmission Path Variant

[163] Another embodiment incorporating various inventive principles is shown in FIG. 20. In this embodiment, a message transmission system includes a first computer 2001 in communication with a second computer 2002 through a network 2011 of intermediary computers. In one variant of this embodiment, the network includes two edge routers 2003 and 2004 each of which is linked to a plurality of Internet Service Providers (ISPs) 2005 through 2010. Each ISP is coupled to a plurality of other ISPs in an arrangement as shown in FIG. 20, which is a representative configuration only and is not intended to be limiting. Each connection between ISPs is labeled in FIG. 20 to indicate a specific physical transmission path (e.g., AD is a physical path that links ISP A (element 2005) to ISP D (element 2008)). Packets arriving at each edge router are selectively transmitted to one of the ISPs to which the router is attached on the basis of a randomly or quasi-randomly selected basis.

[164] As shown in FIG. 21, computer 2001 or edge router 2003 incorporates a plurality of link transmission tables 2100 that identify, for each potential transmission path through the network, valid sets of IP addresses that can be used to transmit the packet. For example, AD table 2101 contains a plurality of IP source/destination pairs that are randomly or quasi-randomly generated. When a packet is to be transmitted from first computer 2001 to second computer 2002, one of the link tables is randomly (or quasi-randomly) selected, and the next valid source/destination address pair from that table is used to transmit the packet through the network. If path AD is randomly selected, for example, the next source/destination IP address pair (which is pre-determined to transmit between ISP A (element 2005) and ISP B (element 2008)) is used to transmit the packet. If one of the transmission paths becomes degraded or inoperative, that link table can be set to a "down" condition as shown in table 2105, thus preventing addresses from being selected from that table. Other transmission paths would be unaffected by this broken link.

3. CONTINUATION-IN-PART IMPROVEMENTS

[165] The following describes various improvements and features that can be applied to the embodiments described above. The improvements include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities. Each is discussed separately below.

A. Load Balancer

[166] Various embodiments described above include a system in which a transmitting node and a receiving node are coupled through a plurality of transmission paths, and wherein successive packets are distributed quasi-randomly over the plurality of paths. See, for example, FIGS. 20 and 21 and accompanying description. The improvement extends this basic concept to encompass distributing packets across different paths in such a manner that the loads on the paths are generally balanced according to transmission link quality.

[167] In one embodiment, a system includes a transmitting node and a receiving node that are linked via a plurality of transmission paths having potentially varying transmission quality. Successive packets are transmitted over the paths based on a weight value distribution function for each path. The rate that packets will be transmitted over a given path can be different for each path. The relative "health" of each transmission path is monitored in order to identify paths that have become degraded. In one embodiment, the health of each path is monitored in the transmitter by comparing the number of packets transmitted to the number of packet acknowledgements received. Each transmission path may comprise a physically separate path (e.g., via dial-up phone line, computer network, router, bridge, or the like), or may comprise logically separate paths contained within a broadband communication medium (e.g., separate channels in an FDM, TDM, CDMA, or other type of modulated or unmodulated transmission link).

[168] When the transmission quality of a path falls below a predetermined threshold and there are other paths that can transmit packets, the transmitter changes the weight value used for that path, making it less likely that a given packet will be transmitted over that path. The weight will preferably be set no lower than a minimum value that keeps nominal traffic on the path. The weights of the other available paths are altered to compensate for the change in the affected path. When the quality of a path degrades to where the transmitter is turned off by the synchronization function (i.e., no packets are arriving at the destination), the weight is set to zero. If all transmitters are turned off, no packets are sent.

[169] Conventional TCP/IP protocols include a "throttling" feature that reduces the transmission rate of packets when it is determined that delays or errors are occurring in transmission. In this respect, timers are sometimes used to determine whether packets have been received. These conventional techniques for limiting transmission of packets, however, do not involve multiple transmission paths between two nodes wherein transmission across a particular path relative to the others is changed based on link quality.

[170] According to certain embodiments, in order to damp oscillations that might otherwise occur if weight distributions are changed drastically (e.g., according to a step function), a linear or an exponential decay formula can be applied to gradually decrease the weight value over time that a degrading path will be used. Similarly, if the health of a degraded path improves, the weight value for that path is gradually increased.

[171] Transmission link health can be evaluated by comparing the number of packets that are acknowledged within the transmission window (see embodiments discussed above) to the number of packets transmitted within that window and by the state of the transmitter (i.e., on or off). In other words, rather than accumulating general transmission statistics over time for a path, one specific implementation uses the "windowing" concepts described above to evaluate transmission path health.

[172] The same scheme can be used to shift virtual circuit paths from an "unhealthy" path to a "healthy" one, and to select a path for a new virtual circuit.

[173] FIG. 22A shows a flowchart for adjusting weight values associated with a plurality of transmission links. It is assumed that software executing in one or more computer nodes executes the steps shown in FIG. 22A. It is also assumed that the software can be stored on a computer-readable medium such as a magnetic or optical disk for execution by a computer.

[174] Beginning in step 2201, the transmission quality of a given transmission path is measured. As described above, this measurement can be based on a comparison between the number of packets transmitted over a particular link to the number of packet acknowledgements received over the link (e.g., per unit time, or in absolute terms). Alternatively, the quality can be evaluated by comparing the number of packets that are acknowledged within the transmission window to the number of packets that were transmitted within that window. In yet another variation, the number of missed synchronization messages can be used to indicate link quality. Many other variations are of course possible.

[175] In step 2202, a check is made to determine whether more than one transmitter (e.g., transmission path) is turned on. If not, the process is terminated and resumes at step 2201.

[176] In step 2203, the link quality is compared to a given threshold (e.g., 50%, or any arbitrary number). If the quality falls below the threshold, then in step 2207 a check is made to determine whether the weight is above a minimum level (e.g., 1%). If not, then in step 2209 the weight is set to the minimum level and processing resumes at step 2201. If the weight is above the minimum level, then in step 2208 the weight is gradually decreased for the path, then in step 2206 the weights for the remaining paths are adjusted accordingly to compensate (e.g., they are increased).

[177] If in step 2203 the quality of the path was greater than or equal to the threshold, then in step 2204 a check is made to determine whether the weight is less than a steady-state value for that path. If so, then in step 2205 the weight is increased toward the steady-state value, and in step 2206 the weights for the remaining paths are adjusted accordingly to compensate (e.g., they are decreased). If in step 2204 the weight is not less than the steady-state value, then processing resumes at step 2201 without adjusting the weights.

[178] The weights can be adjusted incrementally according to various functions, preferably by changing the value gradually. In one embodiment, a linearly decreasing function is used to adjust the weights; according to another embodiment, an exponential decay function is used. Gradually changing the weights helps to damp oscillators that might otherwise occur if the probabilities were abruptly.

[179] Although not explicitly shown in FIG. 22A the process can be performed only periodically (e.g., according to a time schedule), or it can be continuously run, such as in a background mode of operation. In one embodiment, the combined weights of all potential paths should add up to unity (e.g., when the weighting for one path is decreased, the corresponding weights that the other paths will be selected will increase).

[180] Adjustments to weight values for other paths can be prorated. For example, a decrease of 10% in weight value for one path could result in an evenly distributed increase in the weights for the remaining paths. Alternatively, weightings could be adjusted according to a weighted formula as desired (e.g., favoring healthy paths over less healthy paths). In yet another variation, the difference in weight value can be amortized over the remaining links in a manner that is proportional to their traffic weighting.

[181] FIG. 22B shows steps that can be executed to shut down transmission links where a transmitter turns off. In step 2210, a transmitter shut-down event occurs. In step 2211, a test is made to determine whether at least one transmitter is still turned on. If not, then in step 2215 all packets are dropped until a transmitter turns on. If in step 2211 at least one transmitter is turned on, then in step 2212 the weight for the path is set to zero, and the weights for the remaining paths are adjusted accordingly.

[182] FIG. 23 shows a computer node 2301 employing various principles of the above-described embodiments. It is assumed that two computer nodes of the type shown in FIG. 23 communicate over a plurality of separate physical transmission paths. As shown in FIG. 23, four transmission paths X1 through X4 are defined for communicating between the two nodes. Each node includes a packet transmitter 2302 that operates in accordance with a transmit table 2308 as described above. (The packet transmitter could also operate without using the IP-hopping

features described above, but the following description assumes that some form of hopping is employed in conjunction with the path selection mechanism.). The computer node also includes a packet receiver 2303 that operates in accordance with a receive table 2309, including a moving window W that moves as valid packets are received. Invalid packets having source and destination addresses that do not fall within window W are rejected.

[183] As each packet is readied for transmission, source and destination IP addresses (or other discriminator values) are selected from transmit table 2308 according to any of the various algorithms described above, and packets containing these source/destination address pairs, which correspond to the node to which the four transmission paths are linked, are generated to a transmission path switch 2307. Switch 2307, which can comprise a software function, selects from one of the available transmission paths according to a weight distribution table 2306. For example, if the weight for path X1 is 0.2, then every fifth packet will be transmitted on path X1. A similar regime holds true for the other paths as shown. Initially, each link's weight value can be set such that it is proportional to its bandwidth, which will be referred to as its "steady-state" value.

[184] Packet receiver 2303 generates an output to a link quality measurement function 2304 that operates as described above to determine the quality of each transmission path. (The input to packet receiver 2303 for receiving incoming packets is omitted for clarity). Link quality measurement function 2304 compares the link quality to a threshold for each transmission link and, if necessary, generates an output to weight adjustment function 2305. If a weight adjustment is required, then the weights in table 2306 are adjusted accordingly, preferably according to a gradual (e.g., linearly or exponentially declining) function. In one embodiment, the weight values for all available paths are initially set to the same value, and only when paths degrade in quality are the weights changed to reflect differences.

[185] Link quality measurement function 2304 can be made to operate as part of a synchronizer function as described above. That is, if resynchronization occurs and the receiver detects that synchronization has been lost (e.g., resulting in the synchronization window W being advanced out of sequence), that fact can be used to drive link quality measurement function 2304. According to one embodiment, load balancing is performed using information garnered during the normal synchronization, augmented slightly to communicate link health from the receiver to the transmitter. The receiver maintains a count, MESS_R(W), of the messages received in synchronization window W. When it receives a synchronization request (SYNC_REQ) corresponding to the end of window W, the receiver includes counter MESS_R in the resulting synchronization acknowledgement (SYNC_ACK) sent back to the transmitter. This allows the transmitter to compare messages sent to messages received in order to asses the health of the link.

[186] If synchronization is completely lost, weight adjustment function 2305 decreases the weight value on the affected path to zero. When synchronization is regained, the weight value for the affected path is gradually increased to its original value. Alternatively, link quality can be measured by evaluating the length of time required for the receiver to acknowledge a synchronization request. In one embodiment, separate transmit and receive tables are used for each transmission path.

[187] When the transmitter receives a SYNC_ACK, the MESS_R is compared with the number of messages transmitted in a window (MESS_T). When the transmitter receives a SYNC_ACK, the traffic probabilities will be examined and adjusted if necessary. MESS_R is compared with the number of messages transmitted in a window (MESS_T). There are two possibilities:

1. If MESS_R is less than a threshold value, THRESH, then the link will be deemed to be unhealthy. If the transmitter was turned off, the transmitter is turned on and the weight P for that link will be set to a minimum value MIN. This will keep a trickle of traffic on the link for monitoring purposes until it recovers. If the transmitter was turned on, the weight P for that link will be set to:

$$P' = \alpha \times MIN + (1 - \alpha) \times P \qquad (1)$$

Equation 1 will exponentially damp the traffic weight value to MIN during sustained periods of degraded service.

2. If MESS_R for a link is greater than or equal to THRESH, the link will be deemed healthy. If the weight P for that link is greater than or equal to the steady state value S for that link, then P is left unaltered. If the weight P for that link is less than THRESH then P will be set to:

$$P' = \beta \times S + (1-\beta) \times P \quad (2)$$

where β is a parameter such that $0 <= \beta <= 1$ that determines the damping rate of P.

[188] Equation 2 will increase the traffic weight to S during sustained periods of acceptable service in a damped exponential fashion.

[189] A detailed example will now be provided with reference to FIG. 24. As shown in FIG. 24, a first computer 2401 communicates with a second computer 2402 through two routers 2403 and 2404. Each router is coupled to the other router through three transmission links. As described above, these may be physically diverse links or logical links (including virtual private networks).

[190] Suppose that a first link L1 can sustain a transmission bandwidth of 100 Mb/s and has a window size of 32; link L2 can sustain 75 Mb/s and has a window size of 24; and link L3 can sustain 25 Mb/s and has a window size of 8. The combined links can thus sustain 200Mb/s. The steady state traffic weights are 0.5 for link L1; 0.375 for link L2, and 0.125 for link L3. MIN=1Mb/s, THRESH =0.8 MESS_T for each link, α=.75 and β=.5. These traffic weights will remain stable until a link stops for synchronization or reports a number of packets received less than its THRESH. Consider the following sequence of events:

1. Link L1 receives a SYNC_ACK containing a MESS_R of 24, indicating that only 75% of the MESS_T (32) messages transmitted in the last window were successfully received. Link 1 would be below THRESH (0.8). Consequently, link L1's traffic weight value would be reduced to 0.12825, while link L2's traffic weight value would be increased to 0.65812 and link L3's traffic weight value would be increased to 0.217938.
2. Link L2 and L3 remained healthy and link L1 stopped to synchronize. Then link L1's traffic weight value would be set to 0, link L2's traffic weight value would be set to 0.75, and link L33's traffic weight value would be set to 0.25.
3. Link L1 finally received a SYNC_ACK containing a MESS_R of 0 indicating that none of the MESS_T (32) messages transmitted in the last window were successfully received. Link L1 would be below THRESH. Link L1's traffic weight value would be

increased to .005, link L2's traffic weight value would be decreased to 0.74625, and link L3's traffic weight value would be decreased to 0.24875.

4. Link L1 received a SYNC_ACK containing a MESS_R of 32 indicating that 100% of the MESS_T (32) messages transmitted in the last window were successfully received. Link L1 would be above THRESH. Link L1's traffic weight value would be increased to 0.2525, while link L2's traffic weight value would be decreased to 0.560625 and link L3's traffic weight value would be decreased to .186875.
5. Link L1 received a SYNC_ACK containing a MESS_R of 32 indicating that 100% of the MESS_T (32) messages transmitted in the last window were successfully received. Link L1 would be above THRESH. Link L1's traffic weight value would be increased to 0.37625; link L2's traffic weight value would be decreased to 0.4678125, and link L3's traffic weight value would be decreased to 0.1559375.
6. Link L1 remains healthy and the traffic probabilities approach their steady state traffic probabilities.

B. Use of a DNS Proxy to Transparently Create Virtual Private Networks

[191] A second improvement concerns the automatic creation of a virtual private network (VPN) in response to a domain-name server look-up function.

[192] Conventional Domain Name Servers (DNSs) provide a look-up function that returns the IP address of a requested computer or host. For example, when a computer user types in the web name "Yahoo.com," the user's web browser transmits a request to a DNS, which converts the name into a four-part IP address that is returned to the user's browser and then used by the browser to contact the destination web site.

[193] This conventional scheme is shown in FIG. 25. A user's computer 2501 includes a client application 2504 (for example, a web browser) and an IP protocol stack 2505. When the user enters the name of a destination host, a request DNS REQ is made (through IP protocol stack 2505) to a DNS 2502 to look up the IP address associated with the name. The DNS returns the IP address DNS RESP to client application 2504, which is then able to use the IP address to communicate with the host 2503 through separate transactions such as PAGE REQ and PAGE RESP.

[194] In the conventional architecture shown in FIG. 25, nefarious listeners on the Internet could intercept the DNS REQ and DNS RESP packets and thus learn what IP addresses the user was contacting. For example, if a user wanted to set up a secure communication path with a web site having the name "Target.com," when the user's browser contacted a DNS to find the IP address for that web site, the true IP address of that web site would be revealed over the Internet as part of the DNS inquiry. This would hamper anonymous communications on the Internet.

[195] One conventional scheme that provides secure virtual private networks over the Internet provides the DNS server with the public keys of the machines that the DNS server has the addresses for. This allows hosts to retrieve automatically the public keys of a host that the host is to communicate with so that the host can set up a VPN without having the user enter the public key of the destination host. One implementation of this standard is presently being developed as part of the FreeS/WAN project(RFC 2535).

[196] The conventional scheme suffers from certain drawbacks. For example, any user can perform a DNS request. Moreover, DNS requests resolve to the same value for all users.

[197] According to certain aspects of the invention, a specialized DNS server traps DNS requests and, if the request is from a special type of user (e.g., one for which secure communication services are defined), the server does not return the true IP address of the target node, but instead automatically sets up a virtual private network between the target node and the user. The VPN is preferably implemented using the IP address "hopping" features of the basic invention described above, such that the true identity of the two nodes cannot be determined even if packets during the communication are intercepted. For DNS requests that are determined to not require secure services (e.g., an unregistered user), the DNS server transparently "passes through" the request to provide a normal look-up function and return the IP address of the target web server, provided that the requesting host has permissions to resolve unsecured sites. Different users who make an identical DNS request could be provided with different results.

[198] FIG. 26 shows a system employing various principles summarized above. A user's computer 2601 includes a conventional client (e.g., a web browser) 2605 and an IP protocol stack 2606 that preferably operates in accordance with an IP hopping function 2607 as outlined above. A modified DNS server 2602 includes a conventional DNS server function 2609 and a DNS proxy 2610. A gatekeeper server 2603 is interposed between the modified DNS server and a secure target site 2704. An "unsecure" target site 2611 is also accessible via conventional IP protocols.

[199] According to one embodiment, DNS proxy 2610 intercepts all DNS lookup functions from client 2605 and determines whether access to a secure site has been requested. If access to a secure site has been requested (as determined, for example, by a domain name extension, or by reference to an internal table of such sites), DNS proxy 2610 determines whether the user has sufficient security privileges to access the site. If so, DNS proxy 2610 transmits a message to gatekeeper 2603 requesting that a virtual private network be created between user computer 2601 and secure target site 2604. In one embodiment, gatekeeper 2603 creates "hopblocks" to be used by computer 2601 and secure target site 2604 for secure communication. Then, gatekeeper 2603 communicates these to user computer 2601. Thereafter, DNS proxy 2610 returns to user computer 2601 the resolved address passed to it by the gatekeeper (this address could be different from the actual target computer) 2604, preferably using a secure administrative VPN. The address that is returned need not be the actual address of the destination computer.

[200] Had the user requested lookup of a non-secure web site such as site 2611, DNS proxy would merely pass through to conventional DNS server 2609 the look-up request, which would be handled in a conventional manner, returning the IP address of non-secure web site 2611. If the user had requested lookup of a secure web site but lacked credentials to create such a connection, DNS proxy 2610 would return a "host unknown" error to the user. In this manner, different users requesting access to the same DNS name could be provided with different look-up results.

[201] Gatekeeper 2603 can be implemented on a separate computer (as shown in FIG. 26) or as a function within modified DNS server 2602. In general, it is anticipated that gatekeeper 2703 facilitates the allocation and exchange of information needed to communicate securely, such as using "hopped" IP addresses. Secure hosts such as site 2604 are assumed to be equipped with a secure communication function such as an IP hopping function 2608.

[202] It will be appreciated that the functions of DNS proxy 2610 and DNS server 2609 can be combined into a single server for convenience. Moreover, although element 2602 is shown as combining the functions of two servers, the two servers can be made to operate independently.

[203] FIG. 27 shows steps that can be executed by DNS proxy server 2610 to handle requests for DNS look-up for secure hosts. In step 2701, a DNS look-up request is received for a target host. In step 2702, a check is made to determine whether access to a secure host was requested. If not, then in step 2703 the DNS request is passed to conventional DNS server 2609, which looks up the IP address of the target site and returns it to the user's application for further processing.

[204] In step 2702, if access to a secure host was requested, then in step 2704 a further check is made to determine whether the user is authorized to connect to the secure host. Such a check can be made with reference to an internally stored list of authorized IP addresses, or can be made by communicating with gatekeeper 2603 (e.g., over an "administrative" VPN that is secure). It will be appreciated that different levels of security can also be provided for different categories of hosts. For example, some sites may be designated as having a certain security level, and the security level of the user requesting access must match that security level. The user's security level can also be determined by transmitting a request message back to the user's computer requiring that it prove that it has sufficient privileges.

[205] If the user is not authorized to access the secure site, then a "host unknown" message is returned (step 2705). If the user has sufficient security privileges, then in step 2706 a secure VPN is established between the user's computer and the secure target site. As described above, this is preferably done by allocating a hopping regime that will be carried out between the user's computer and the secure target site, and is preferably performed transparently to the user (i.e., the user need not be involved in creating the secure link). As described in various embodiments of this application, any of various fields can be "hopped" (e.g., IP source/destination addresses; a field in the header; etc.) in order to communicate securely.

[206] Some or all of the security functions can be embedded in gatekeeper 2603, such that it handles all requests to connect to secure sites. In this embodiment, DNS proxy 2610

communicates with gatekeeper 2603 to determine (preferably over a secure administrative VPN) whether the user has access to a particular web site. Various scenarios for implementing these features are described by way of example below:

Scenario #1: Client has permission to access target computer, and gatekeeper has a rule to make a VPN for the client. In this scenario, the client's DNS request would be received by the DNS proxy server 2610, which would forward the request to gatekeeper 2603. The gatekeeper would establish a VPN between the client and the requested target. The gatekeeper would provide the address of the destination to the DNS proxy, which would then return the resolved name as a result. The resolved address can be transmitted back to the client in a secure administrative VPN.

Scenario #2: Client does not have permission to access target computer. In this scenario, the client's DNS request would be received by the DNS proxy server 2610, which would forward the request to gatekeeper 2603. The gatekeeper would reject the request, informing DNS proxy server 2610 that it was unable to find the target computer. The DNS proxy 2610 would then return a "host unknown" error message to the client.

Scenario #3: Client has permission to connect using a normal non-VPN link, and the gatekeeper does not have a rule to set up a VPN for the client to the target site. In this scenario, the client's DNS request is received by DNS proxy server 2610, which would check its rules and determine that no VPN is needed. Gatekeeper 2603 would then inform the DNS proxy server to forward the request to conventional DNS server 2609, which would resolve the request and return the result to the DNS proxy server and then back to the client.

Scenario #4: Client does not have permission to establish a normal/non-VPN link, and the gatekeeper does not have a rule to make a VPN for the client to the target site. In this scenario, the DNS proxy server would receive the client's DNS request and forward it to gatekeeper 2603. Gatekeeper 2603 would determine that no special VPN was needed, but that the client is not authorized to communicate with non-VPN members. The gatekeeper would reject the request, causing DNS proxy server 2610 to return an error message to the client.

C. Large Link to Small Link Bandwidth Management

[207] One feature of the basic architecture is the ability to prevent so-called "denial of service" attacks that can occur if a computer hacker floods a known Internet node with packets, thus preventing the node from communicating with other nodes. Because IP addresses or other fields are "hopped" and packets arriving with invalid addresses are quickly discarded, Internet nodes are protected against flooding targeted at a single IP address.

[208] In a system in which a computer is coupled through a link having a limited bandwidth (e.g., an edge router) to a node that can support a much higher-bandwidth link (e.g., an Internet Service Provider), a potential weakness could be exploited by a determined hacker. Referring to FIG. 28, suppose that a first host computer 2801 is communicating with a second host computer 2804 using the IP address hopping principles described above. The first host computer is coupled through an edge router 2802 to an Internet Service Provider (ISP) 2803 through a low bandwidth link (LOW BW), and is in turn coupled to second host computer 2804 through parts of the Internet through a high bandwidth link (HIGH BW). In this architecture, the ISP is able to support a high bandwidth to the internet, but a much lower bandwidth to the edge router 2802.

[209] Suppose that a computer hacker is able to transmit a large quantity of dummy packets addressed to first host computer 2801 across high bandwidth link HIGH BW. Normally, host computer 2801 would be able to quickly reject the packets since they would not fall within the acceptance window permitted by the IP address hopping scheme. However, because the packets must travel across low bandwidth link LOW BW, the packets overwhelm the lower bandwidth link before they are received by host computer 2801. Consequently, the link to host computer 2801 is effectively flooded before the packets can be discarded.

[210] According to one inventive improvement, a "link guard" function 2805 is inserted into the high-bandwidth node (e.g., ISP 2803) that quickly discards packets destined for a low-bandwidth target node if they are not valid packets. Each packet destined for a low-bandwidth node is cryptographically authenticated to determine whether it belongs to a VPN. If it is not a valid VPN packet, the packet is discarded at the high-bandwidth node. If the packet is authenticated as belonging to a VPN, the packet is passed with high preference. If the packet is a valid non-VPN packet, it is passed with a lower quality of service (e.g., lower priority).

[211] In one embodiment, the ISP distinguishes between VPN and non-VPN packets using the protocol of the packet. In the case of IPSEC [rfc 2401], the packets have IP protocols 420 and

421. In the case of the TARP VPN, the packets will have an IP protocol that is not yet defined. The ISP's link guard, 2805, maintains a table of valid VPNs which it uses to validate whether VPN packets are cryptographically valid.

[212] According to one embodiment, packets that do not fall within any hop windows used by nodes on the low-bandwidth link are rejected, or are sent with a lower quality of service. One approach for doing this is to provide a copy of the IP hopping tables used by the low-bandwidth nodes to the high-bandwidth node, such that both the high-bandwidth and low-bandwidth nodes track hopped packets (e.g., the high-bandwidth node moves its hopping window as valid packets are received). In such a scenario, the high-bandwidth node discards packets that do not fall within the hopping window before they are transmitted over the low-bandwidth link. Thus, for example, ISP 2903 maintains a copy 2910 of the receive table used by host computer 2901. Incoming packets that do not fall within this receive table are discarded. According to a different embodiment, link guard 2805 validates each VPN packet using a keyed hashed message authentication code (HMAC) [rfc 2104]. According to another embodiment, separate VPNs (using, for example, hopblocks) can be established for communicating between the low-bandwidth node and the high-bandwidth node (i.e., packets arriving at the high-bandwidth node are converted into different packets before being transmitted to the low-bandwidth node).

[213] As shown in FIG. 29, for example, suppose that a first host computer 2900 is communicating with a second host computer 2902 over the Internet, and the path includes a high bandwidth link HIGH BW to an ISP 2901 and a low bandwidth link LOW BW through an edge router 2904. In accordance with the basic architecture described above, first host computer 2900 and second host computer 2902 would exchange hopblocks (or a hopblock algorithm) and would be able to create matching transmit and receive tables 2905, 2906, 2912 and 2913. Then in accordance with the basic architecture, the two computers would transmit packets having seemingly random IP source and destination addresses, and each would move a corresponding hopping window in its receive table as valid packets were received.

[214] Suppose that a nefarious computer hacker 2903 was able to deduce that packets having a certain range of IP addresses (e.g., addresses 100 to 200 for the sake of simplicity) are being transmitted to ISP 2901, and that these packets are being forwarded over a low-bandwidth link.

Hacker computer 2903 could thus "flood" packets having addresses falling into the range 100 to 200, expecting that they would be forwarded along low bandwidth link LOW BW, thus causing the low bandwidth link to become overwhelmed. The fast packet reject mechanism in first host computer 3000 would be of little use in rejecting these packets, since the low bandwidth link was effectively jammed before the packets could be rejected. In accordance with one aspect of the improvement, however, VPN link guard 2911 would prevent the attack from impacting the performance of VPN traffic because the packets would either be rejected as invalid VPN packets or given a lower quality of service than VPN traffic over the lower bandwidth link. A denial-of-service flood attack could, however, still disrupt non-VPN traffic.

[215] According to one embodiment of the improvement, ISP 2901 maintains a separate VPN with first host computer 2900, and thus translates packets arriving at the ISP into packets having a different IP header before they are transmitted to host computer 2900. The cryptographic keys used to authenticate VPN packets at the link guard 2911 and the cryptographic keys used to encrypt and decrypt the VPN packets at host 2902 and host 2901 can be different, so that link guard 2911 does not have access to the private host data; it only has the capability to authenticate those packets.

[216] According to yet a third embodiment, the low-bandwidth node can transmit a special message to the high-bandwidth node instructing it to shut down all transmissions on a particular IP address, such that only hopped packets will pass through to the low-bandwidth node. This embodiment would prevent a hacker from flooding packets using a single IP address. According to yet a fourth embodiment, the high-bandwidth node can be configured to discard packets transmitted to the low-bandwidth node if the transmission rate exceeds a certain predetermined threshold for any given IP address; this would allow hopped packets to go through. In this respect, link guard 2911 can be used to detect that the rate of packets on a given IP address are exceeding a threshold rate; further packets addressed to that same IP address would be dropped or transmitted at a lower priority (e.g., delayed).

D. <u>Traffic Limiter</u>

[217] In a system in which multiple nodes are communicating using "hopping" technology, a treasonous insider could internally flood the system with packets. In order to prevent this possibility, one inventive improvement involves setting up "contracts" between nodes in the system, such that a receiver can impose a bandwidth limitation on each packet sender. One technique for doing this is to delay acceptance of a checkpoint synchronization request from a sender until a certain time period (e.g., one minute) has elapsed. Each receiver can effectively control the rate at which its hopping window moves by delaying "SYNC ACK" responses to "SYNC_REQ" messages.

[218] A simple modification to the checkpoint synchronizer will serve to protect a receiver from accidental or deliberate overload from an internally treasonous client. This modification is based on the observation that a receiver will not update its tables until a SYNC_REQ is received on hopped address CKPT_N. It is a simple matter of deferring the generation of a new CKPT_N until an appropriate interval after previous checkpoints.

[219] Suppose a receiver wished to restrict reception from a transmitter to 100 packets a second, and that checkpoint synchronization messages were triggered every 50 packets. A compliant transmitter would not issue new SYNC_REQ messages more often than every 0.5 seconds. The receiver could delay a non-compliant transmitter from synchronizing by delaying the issuance of CKPT_N for 0.5 second after the last SYNC_REQ was accepted.

[220] In general, if M receivers need to restrict N transmitters issuing new SYNC_REQ messages after every W messages to sending R messages a second in aggregate, each receiver could defer issuing a new CKPT_N until MxNxW/R seconds have elapsed since the last SYNC_REQ has been received and accepted. If the transmitter exceeds this rate between a pair of checkpoints, it will issue the new checkpoint before the receiver is ready to receive it, and the SYNC_REQ will be discarded by the receiver. After this, the transmitter will re-issue the SYNC_REQ every T1 seconds until it receives a SYNC_ACK. The receiver will eventually update CKPT_N and the SYNC_REQ will be acknowledged. If the transmission rate greatly exceeds the allowed rate, the transmitter will stop until it is compliant. If the transmitter exceeds the allowed rate by a little, it will eventually stop after several rounds of delayed synchronization until it is in compliance. Hacking the transmitter's code to not shut off only permits the transmitter to lose the acceptance window. In this case it can recover the window and proceed only after it is compliant again.

[221] Two practical issues should be considered when implementing the above scheme:

1. The receiver rate should be slightly higher than the permitted rate in order to allow for statistical fluctuations in traffic arrival times and non-uniform load balancing.

2. Since a transmitter will rightfully continue to transmit for a period after a SYNC_REQ is transmitted, the algorithm above can artificially reduce the transmitter's bandwidth. If events prevent a compliant transmitter from synchronizing for a period (e.g. the network dropping a SYNC_REQ or a SYNC_ACK) a SYNC_REQ will be accepted later than expected. After this, the transmitter will transmit fewer than expected messages before encountering the next checkpoint. The new checkpoint will not have been activated and the transmitter will have to retransmit the SYNC_REQ. This will appear to the receiver as if the transmitter is not compliant. Therefore, the next checkpoint will be accepted late from the transmitter's perspective. This has the effect of reducing the transmitter's allowed packet rate until the transmitter transmits at a packet rate below the agreed upon rate for a period of time.

[222] To guard against this, the receiver should keep track of the times that the last C SYNC_REQs were received and accepted and use the minimum of MxNxW/R seconds after the last SYNC_REQ has been received and accepted, 2xMxNxW/R seconds after next to the last SYNC_REQ has been received and accepted, CxMxNxW/R seconds after $(C-1)^{th}$ to the last SYNC_REQ has been received, as the time to activate CKPT_N. This prevents the receiver from inappropriately limiting the transmitter's packet rate if at least one out of the last C SYNC_REQs was processed on the first attempt.

[223] FIG. 30 shows a system employing the above-described principles. In FIG. 30, two computers 3000 and 3001 are assumed to be communicating over a network N in accordance with the "hopping" principles described above (e.g., hopped IP addresses, discriminator values, etc.). For the sake of simplicity, computer 3000 will be referred to as the receiving computer and computer 3001 will be referred to as the transmitting computer, although full duplex operation is of course contemplated. Moreover, although only a single transmitter is shown, multiple transmitters can transmit to receiver 3000.

[224] As described above, receiving computer 3000 maintains a receive table 3002 including a window W that defines valid IP address pairs that will be accepted when appearing in incoming

data packets. Transmitting computer 3001 maintains a transmit table 3003 from which the next IP address pairs will be selected when transmitting a packet to receiving computer 3000. (For the sake of illustration, window W is also illustrated with reference to transmit table 3003). As transmitting computer moves through its table, it will eventually generate a SYNC_REQ message as illustrated in function 3010. This is a request to receiver 3000 to synchronize the receive table 3002, from which transmitter 3001 expects a response in the form of a CKPT_N (included as part of a SYNC_ACK message). If transmitting computer 3001 transmits more messages than its allotment, it will prematurely generate the SYNC_REQ message. (If it has been altered to remove the SYNC_REQ message generation altogether, it will fall out of synchronization since receiver 3000 will quickly reject packets that fall outside of window W, and the extra packets generated by transmitter 3001 will be discarded).

[225] In accordance with the improvements described above, receiving computer 3000 performs certain steps when a SYNC_REQ message is received, as illustrated in FIG. 30. In step 3004, receiving computer 3000 receives the SYNC_REQ message. In step 3005, a check is made to determine whether the request is a duplicate. If so, it is discarded in step 3006. In step 3007, a check is made to determine whether the SYNC_REQ received from transmitter 3001 was received at a rate that exceeds the allowable rate R (i.e., the period between the time of the last SYNC_REQ message). The value R can be a constant, or it can be made to fluctuate as desired. If the rate exceeds R, then in step 3008 the next activation of the next CKPT_N hopping table entry is delayed by W/R seconds after the last SYNC_REQ has been accepted.

[226] Otherwise, if the rate has not been exceeded, then in step 3109 the next CKPT_N value is calculated and inserted into the receiver's hopping table prior to the next SYNC_REQ from the transmitter 3101. Transmitter 3101 then processes the SYNC_REQ in the normal manner.

E. <u>Signaling Synchronizer</u>

[227] In a system in which a large number of users communicate with a central node using secure hopping technology, a large amount of memory must be set aside for hopping tables and their supporting data structures. For example, if one million subscribers to a web site occasionally communicate with the web site, the site must maintain one million hopping tables, thus using up valuable computer resources, even though only a small percentage of the users may actually be using the system at any one time. A desirable solution would be a system that permits a certain maximum number of simultaneous links to be maintained, but which would "recognize" millions of registered users at any one time. In other words, out of a population of a million registered users, a few thousand at a time could simultaneously communicate with a central server, without requiring that the server maintain one million hopping tables of appreciable size.

[228] One solution is to partition the central node into two nodes: a signaling server that performs session initiation for user log-on and log-off (and requires only minimally sized tables), and a transport server that contains larger hopping tables for the users. The signaling server listens for the millions of known users and performs a fast-packet reject of other (bogus) packets. When a packet is received from a known user, the signaling server activates a virtual private link (VPL) between the user and the transport server, where hopping tables are allocated and maintained. When the user logs onto the signaling server, the user's computer is provided with hop tables for communicating with the transport server, thus activating the VPL. The VPLs can be torn down when they become inactive for a time period, or they can be torn down upon user log-out. Communication with the signaling server to allow user log-on and log-off can be accomplished using a specialized version of the checkpoint scheme described above.

[229] FIG. 31 shows a system employing certain of the above-described principles. In FIG. 31, a signaling server 3101 and a transport server 3102 communicate over a link. Signaling server 3101 contains a large number of small tables 3106 and 3107 that contain enough information to authenticate a communication request with one or more clients 3103 and 3104. As described in more detail below, these small tables may advantageously be constructed as a special case of the synchronizing checkpoint tables described previously. Transport server 3102, which is preferably a separate computer in communication with signaling server 3101, contains a smaller number of larger hopping tables 3108, 3109, and 3110 that can be allocated to create a VPN with one of the client computers.

[230] According to one embodiment, a client that has previously registered with the system (e.g., via a system administration function, a user registration procedure, or some other method) transmits a request for information from a computer (e.g., a web site). In one variation, the

request is made using a "hopped" packet, such that signaling server 3101 will quickly reject invalid packets from unauthorized computers such as hacker computer 3105. An "administrative" VPN can be established between all of the clients and the signaling server in order to ensure that a hacker cannot flood signaling server 3101 with bogus packets. Details of this scheme are provided below.

[231] Signaling server 3101 receives the request 3111 and uses it to determine that client 3103 is a validly registered user. Next, signaling server 3101 issues a request to transport server 3102 to allocate a hopping table (or hopping algorithm or other regime) for the purpose of creating a VPN with client 3103. The allocated hopping parameters are returned to signaling server 3101 (path 3113), which then supplies the hopping parameters to client 3103 via path 3114, preferably in encrypted form.

[232] Thereafter, client 3103 communicates with transport server 3102 using the normal hopping techniques described above. It will be appreciated that although signaling server 3101 and transport server 3102 are illustrated as being two separate computers, they could of course be combined into a single computer and their functions performed on the single computer. Alternatively, it is possible to partition the functions shown in FIG. 31 differently from as shown without departing from the inventive principles.

[233] One advantage of the above-described architecture is that signaling server 3101 need only maintain a small amount of information on a large number of potential users, yet it retains the capability of quickly rejecting packets from unauthorized users such as hacker computer 3105. Larger data tables needed to perform the hopping and synchronization functions are instead maintained in a transport server 3102, and a smaller number of these tables are needed since they are only allocated for "active" links. After a VPN has become inactive for a certain time period (e.g., one hour), the VPN can be automatically torn down by transport server 3102 or signaling server 3101.

[234] A more detailed description will now be provided regarding how a special case of the checkpoint synchronization feature can be used to implement the signaling scheme described above.

[235] The signaling synchronizer may be required to support many (millions) of standing, low bandwidth connections. It therefore should minimize per-VPL memory usage while providing the security offered by hopping technology. In order to reduce memory usage in the signaling server, the data hopping tables can be completely eliminated and data can be carried as part of the SYNC_REQ message. The table used by the server side (receiver) and client side (transmitter) is shown schematically as element 3106 in FIG. 31.

[236] The meaning and behaviors of CKPT_N, CKPT_O and CKPT_R remain the same from the previous description, except that CKPT_N can receive a combined data and SYNC_REQ message or a SYNC_REQ message without the data.

[237] The protocol is a straightforward extension of the earlier synchronizer. Assume that a client transmitter is on and the tables are synchronized. The initial tables can be generated "out of band." For example, a client can log into a web server to establish an account over the Internet. The client will receive keys etc encrypted over the Internet. Meanwhile, the server will set up the signaling VPN on the signaling server.

[238] Assuming that a client application wishes to send a packet to the server on the client's standing signaling VPL:

1. The client sends the message marked as a data message on the inner header using the transmitter's CKPT_N address. It turns the transmitter off and starts a timer T1 noting CKPT_O. Messages can be one of three types: DATA, SYNC_REQ and SYNC_ACK. In the normal algorithm, some potential problems can be prevented by identifying each message type as part of the encrypted inner header field. In this algorithm, it is important to distinguish a data packet and a SYNC_REQ in the signaling synchronizer since the data and the SYNC_REQ come in on the same address.
2. When the server receives a data message on its CKPT_N, it verifies the message and passes it up the stack. The message can be verified by checking message type and and other information (i.e user credentials) contained in the inner header It replaces its CKPT_O with CKPT_N and generates the next CKPT_N. It updates its transmitter side

CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.

3. When the client side receiver receives a SYNC_ACK on its CKPT_R with a payload matching its transmitter side CKPT_O and the transmitter is off, the transmitter is turned on and the receiver side CKPT_R is updated. If the SYNC_ACK's payload does not match the transmitter side CKPT_O or the transmitter is on, the SYNC_ACK is simply discarded.
4. T1 expires: If the transmitter is off and the client's transmitter side CKPT_O matches the CKPT_O associated with the timer, it starts timer T1 noting CKPT_O again, and a SYNC_REQ is sent using the transmitter's CKPT_O address. Otherwise, no action is taken.
5. When the server receives a SYNC_REQ on its CKPT_N, it replaces its CKPT_O with CKPT_N and generates the next CKPT_N. It updates its transmitter side CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.
6. When the server receives a SYNC_REQ on its CKPT_O, it updates its transmitter side CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.

[239] FIG. 32 shows message flows to highlight the protocol. Reading from top to bottom, the client sends data to the server using its transmitter side CKPT_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT_N into CKPT_O and updates CKPT_N. This message is successfully received and a passed up the stack. It also synchronizes the receiver i.e, the server loads CKPT_N into CKPT_O and generates a new CKPT_N, it generates a new CKPT_R in the server side transmitter and transmits a SYNC_ACK containing the server side receiver's CKPT_O the server. The SYNC_ACK is successfully received at the client. The client side receiver's CKPT_R is updated, the transmitter is turned on and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

[240] Next, the client sends data to the server using its transmitter side CKPT_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT_N into CKPT_O and updates CKPT_N. This message is lost. The client side timer expires and as a result a SYNC_REQ is transmitted on the client side transmitter's CKPT_O (this will keep happening until the SYNC_ACK has been received at the client). The SYNC_REQ is successfully received at the server. It synchronizes the receiver i.e, the server loads CKPT_N into CKPT_O and generates a new CKPT_N, it generates an new CKPT_R in the server side transmitter and transmits a SYNC_ACK containing the server side receiver's CKPT_O the server. The SYNC_ACK is successfully received at the client. The client side receiver's CKPT_R is updated, the transmitter is turned off and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

[241] There are numerous other scenarios that follow this flow. For example, the SYNC_ACK could be lost. The transmitter would continue to re-send the SYNC_REQ until the receiver synchronizes and responds.

[242] The above-described procedures allow a client to be authenticated at signaling server 3201 while maintaining the ability of signaling server 3201 to quickly reject invalid packets, such as might be generated by hacker computer 3205. In various embodiments, the signaling synchronizer is really a derivative of the synchronizer. It provides the same protection as the hopping protocol, and it does so for a large number of low bandwidth connections.

CLAIMS

We Claim:

1. A method for establishing an encrypted channel between a client and a target computer, comprising the steps of:

(i) intercepting a DNS request sent by the client; and

(ii) based on the DNS request, establishing the encrypted channel between the client and the target.

2. The method of claim 1, wherein step (ii) comprises steps of:

a) determining whether the client is authorized to access the target;

b) when the client is authorized to access the target, initiating the encrypted channel; and

c) when the client is not authorized to access the target, sending an error message to the client.

3. The method of claim 2, wherein step b) comprises sending encrypted channel parameters to the client.

4. The method of claim 1, wherein step (ii) occurs in a communication protocol independently of an application program.

5. The method of claim 1, wherein step (i) comprises a DNS proxy server intercepting the DNS request sent by the client.

6. The method of claim 1, wherein step (ii) comprises establishing the encrypted channel responsive to intercepting a DNS request for a domain name comprising a predetermined domain name extension.

7. A method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name extension.

8. The method of claim 7, wherein the creating step is performed in a communication protocol independently of an application program.

9. A method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel in response to detecting a request for access to a predetermined IP address.

10. The method of claim 9, wherein the creating step is performed in a communication protocol independently of an application program.

11. A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:

(i) determining whether the intercepted DNS request corresponds to a secure server;

(ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

12. The data processing device of claim 11, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

13. The data processing device of claim 12, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

14. The data processing device of claim 13, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

15. A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

16. A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) intercepting a DNS request sent by a client;

(ii) determining whether the intercepted DNS request corresponds to a secure server;

(iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

17. The computer readable medium of claim 16, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

18. The computer readable medium of claim 17, wherein step (iii) further comprises the step of:

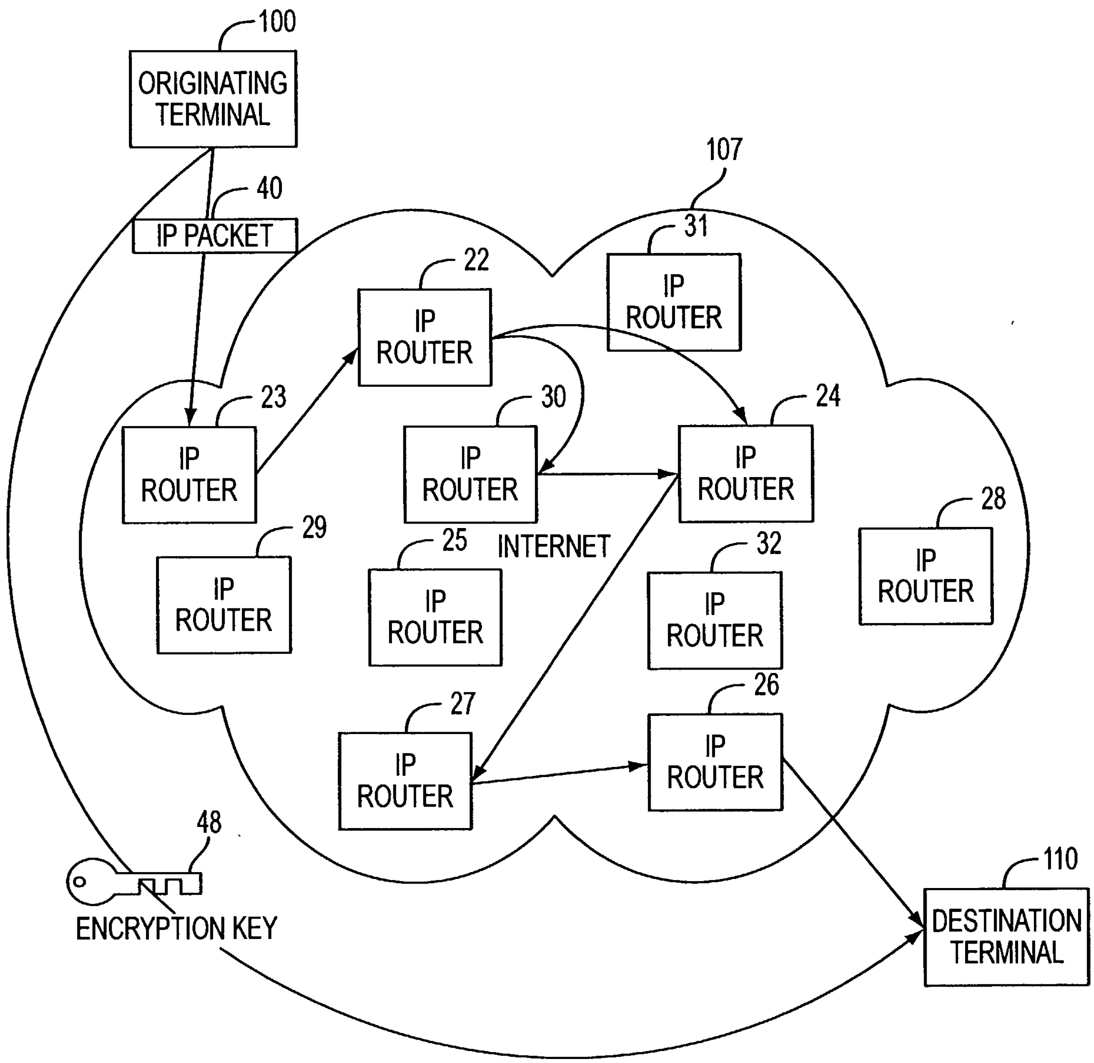
(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

19. The computer readable medium of claim 18, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

20. A computer readable medium comprising computer readable instructions that, when executed, cause a domain name server (DNS) proxy module to intercept DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

ABSTRACT

A plurality of computer nodes communicate using seemingly random Internet Protocol source and destination addresses. Data packets matching criteria defined by a moving window of valid addresses are accepted for further processing, while those that do not meet the criteria are quickly rejected. Improvements to the basic design include (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities.



100
ORIGINATING TERMINAL
40
IP PACKET
107
31
IP ROUTER
22
IP ROUTER
23
IP ROUTER
30
IP ROUTER
24
IP ROUTER
29
IP ROUTER
25
IP ROUTER
INTERNET
32
IP ROUTER
28
IP ROUTER
27
IP ROUTER
26
IP ROUTER
48
ENCRYPTION KEY
110
DESTINATION TERMINAL

FIG. 1

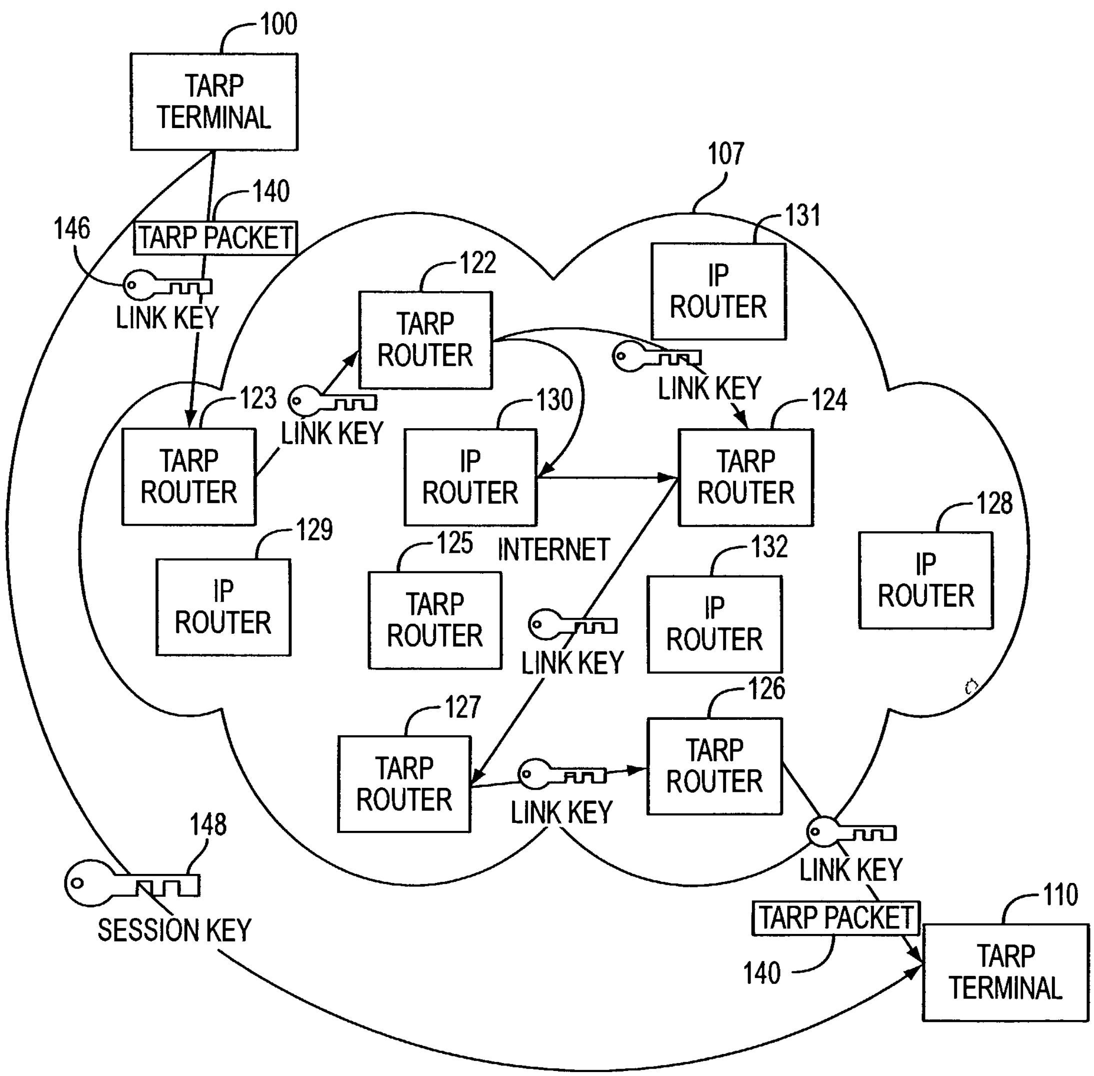
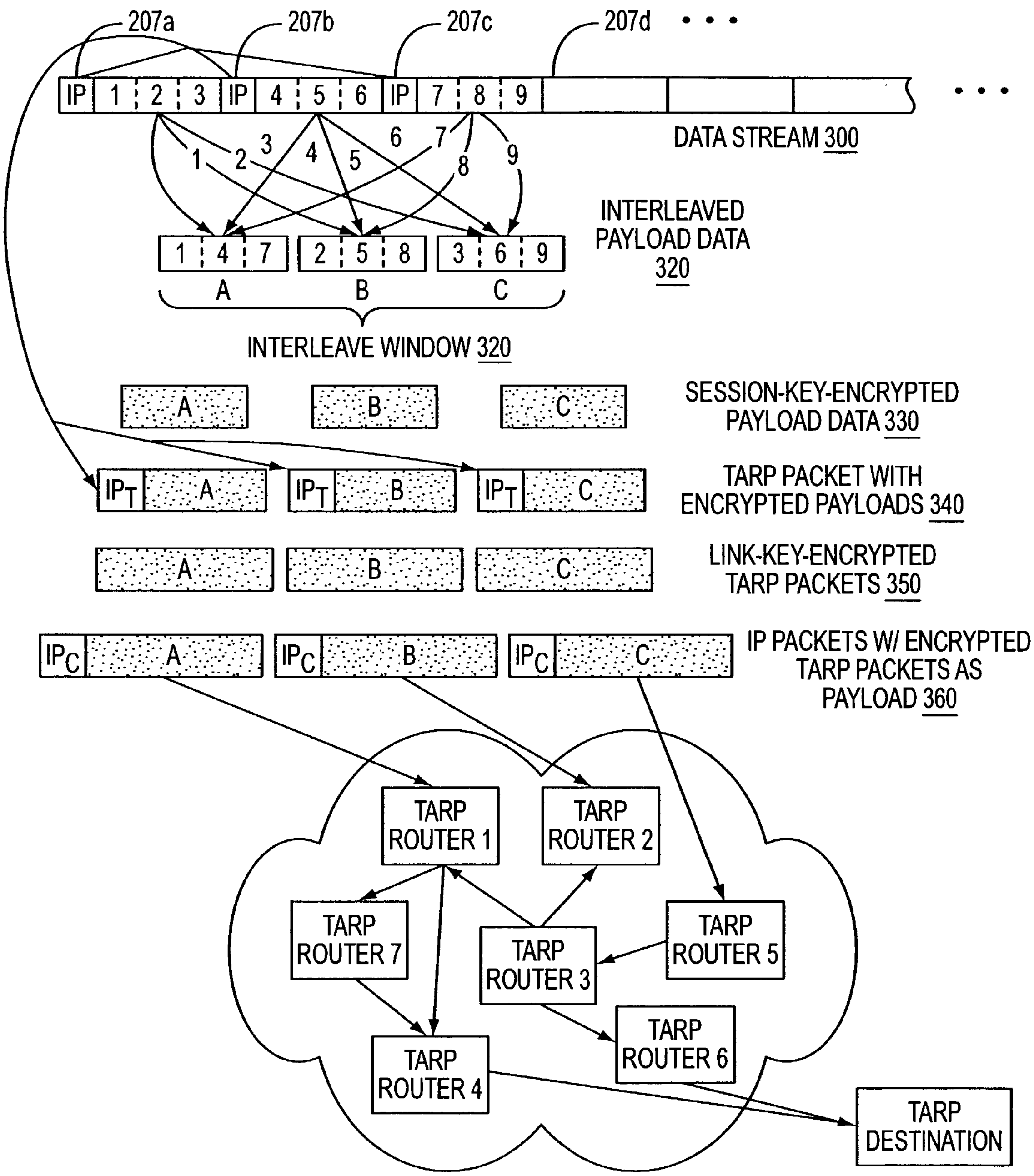

100
TARP TERMINAL
140
TARP PACKET
146
LINK KEY
107
131
IP ROUTER
122
TARP ROUTER
LINK KEY
123
LINK KEY
130
124
TARP ROUTER
IP ROUTER
TARP ROUTER
129
IP ROUTER
125
INTERNET
TARP ROUTER
132
IP ROUTER
128
IP ROUTER
LINK KEY
127
TARP ROUTER
126
TARP ROUTER
LINK KEY
LINK KEY
148
SESSION KEY
TARP PACKET
140
110
TARP TERMINAL

FIG. 2



207a
207b
207c
207d
IP
1
2
3
IP
4
5
6
IP
7
8
9
DATA STREAM 300
INTERLEAVED PAYLOAD DATA 320
1
4
7
2
5
8
3
6
9
A
B
C
INTERLEAVE WINDOW 320
SESSION-KEY-ENCRYPTED PAYLOAD DATA 330
TARP PACKET WITH ENCRYPTED PAYLOADS 340
LINK-KEY-ENCRYPTED TARP PACKETS 350
IP PACKETS W/ ENCRYPTED TARP PACKETS AS PAYLOAD 360
TARP ROUTER 1
TARP ROUTER 2
TARP ROUTER 7
TARP ROUTER 3
TARP ROUTER 5
TARP ROUTER 6
TARP ROUTER 4
TARP DESTINATION

FIG. 3A

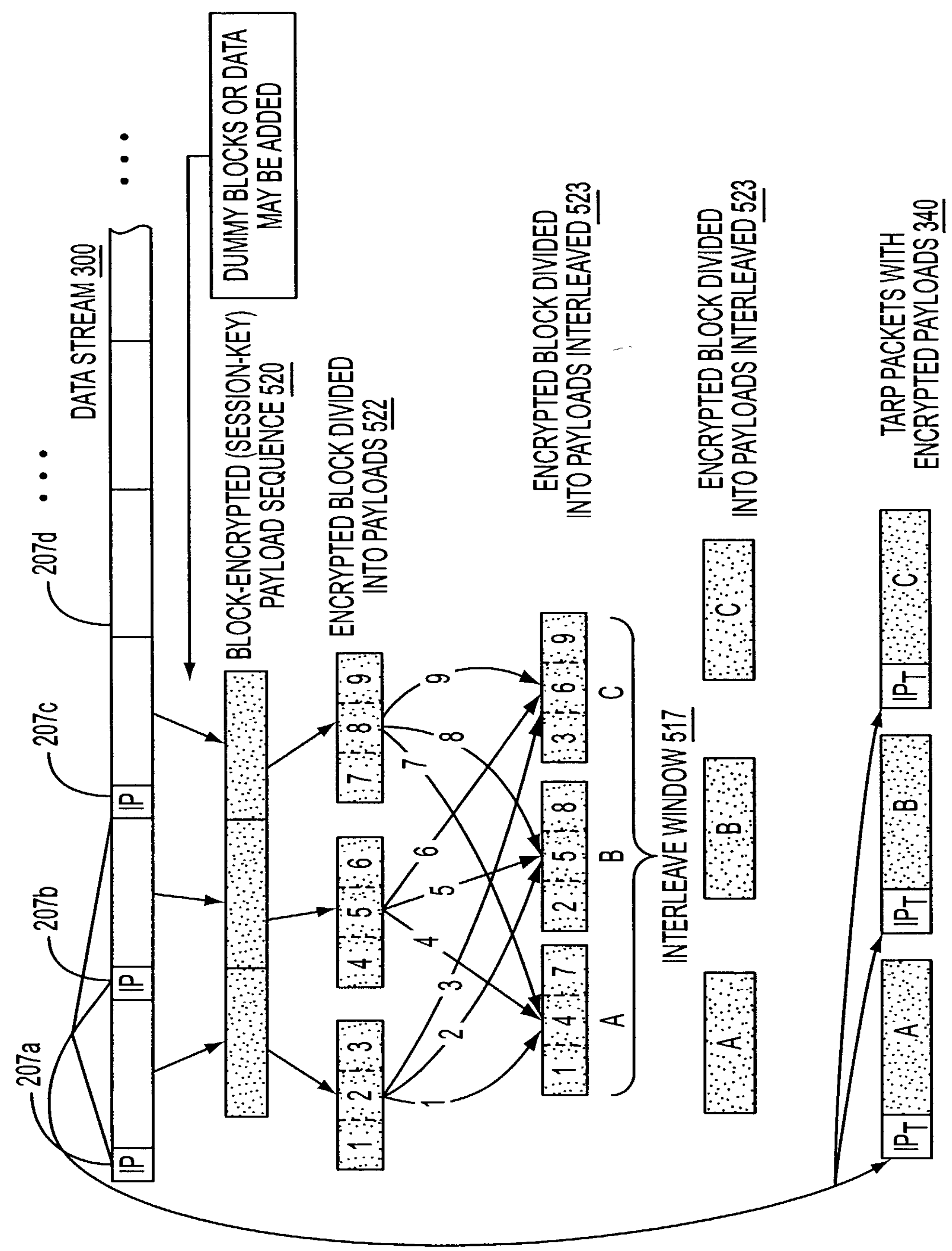

DATA STREAM 300
207a
207b
207c
207d
IP
IP
IP
DUMMY BLOCKS OR DATA MAY BE ADDED
BLOCK-ENCRYPTED (SESSION-KEY) PAYLOAD SEQUENCE 520
ENCRYPTED BLOCK DIVIDED INTO PAYLOADS 522
1 2 3
4 5 6
7 8 9
ENCRYPTED BLOCK DIVIDED INTO PAYLOADS INTERLEAVED 523
1 4 7
2 5 8
3 6 9
A
B
C
INTERLEAVE WINDOW 517
ENCRYPTED BLOCK DIVIDED INTO PAYLOADS INTERLEAVED 523
TARP PACKETS WITH ENCRYPTED PAYLOADS 340
IP$_T$

FIG. 3B

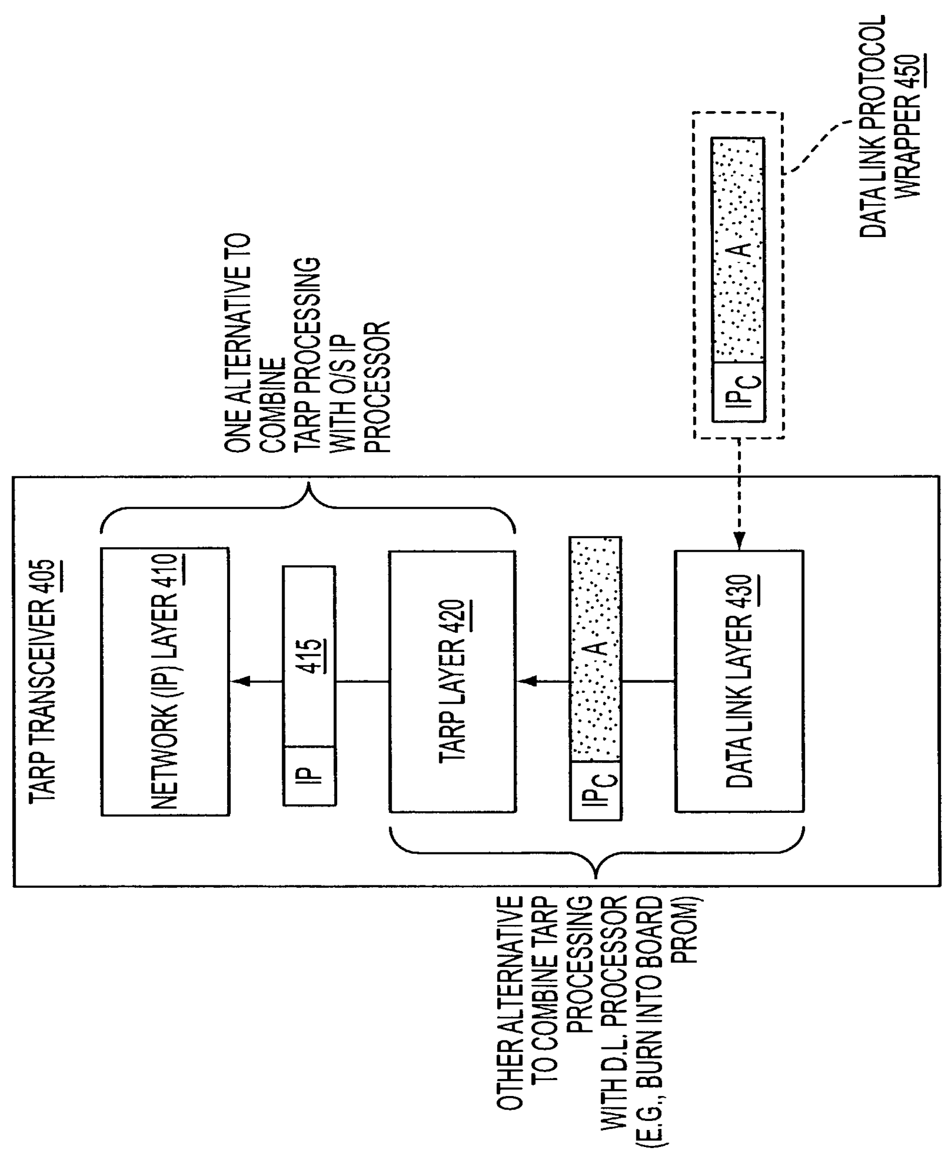

TARP TRANSCEIVER 405
NETWORK (IP) LAYER 410
IP
415
TARP LAYER 420
IPC
A
DATA LINK LAYER 430
ONE ALTERNATIVE TO COMBINE TARP PROCESSING WITH O/S IP PROCESSOR
OTHER ALTERNATIVE TO COMBINE TARP PROCESSING WITH D.L. PROCESSOR (E.G., BURN INTO BOARD PROM)
IPC
A
DATA LINK PROTOCOL WRAPPER 450

FIG. 4

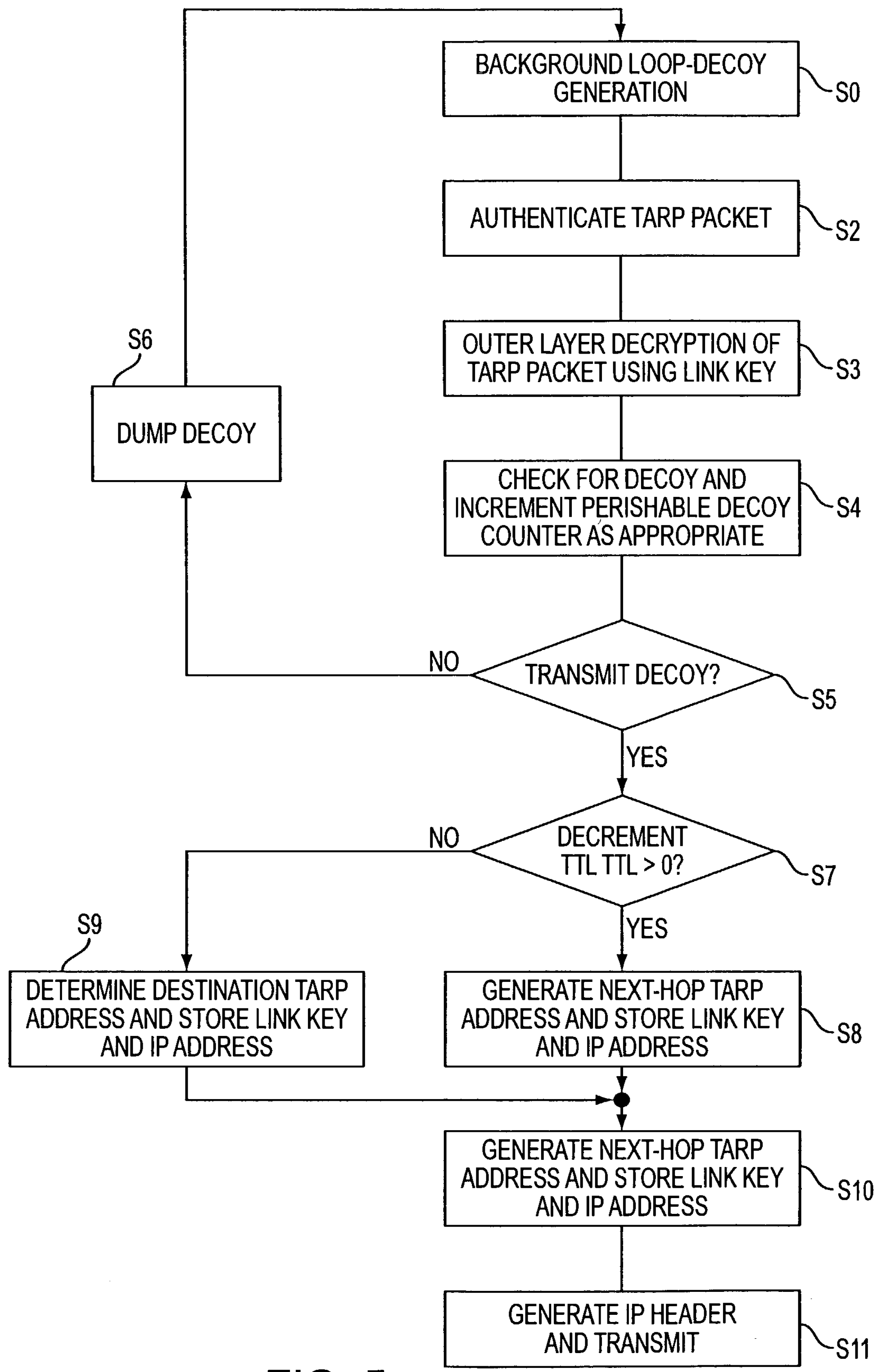

BACKGROUND LOOP-DECOY GENERATION
S0
AUTHENTICATE TARP PACKET
S2
OUTER LAYER DECRYPTION OF TARP PACKET USING LINK KEY
S3
CHECK FOR DECOY AND INCREMENT PERISHABLE DECOY COUNTER AS APPROPRIATE
S4
TRANSMIT DECOY?
S5
NO
S6
DUMP DECOY
YES
DECREMENT TTL TTL > 0?
S7
NO
YES
S9
DETERMINE DESTINATION TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
GENERATE NEXT-HOP TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
S8
GENERATE NEXT-HOP TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
S10
GENERATE IP HEADER AND TRANSMIT
S11

FIG. 5

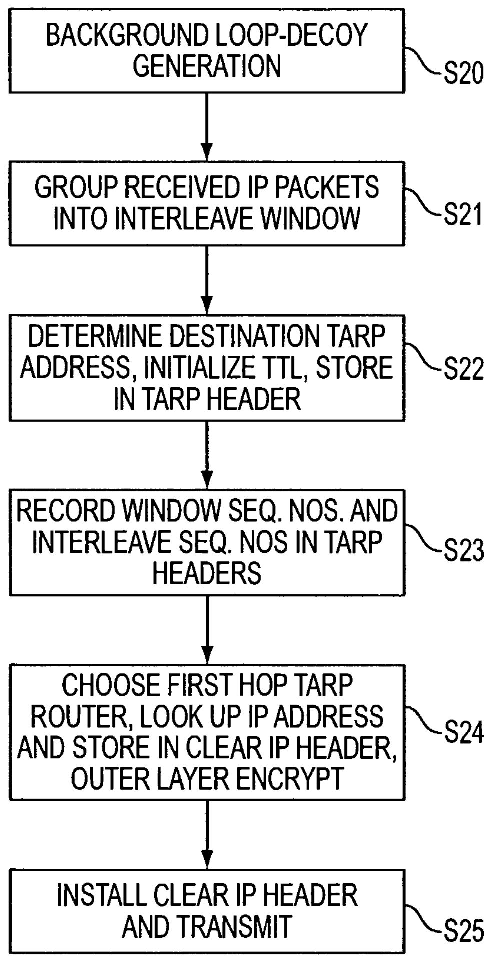

BACKGROUND LOOP-DECOY GENERATION
S20
GROUP RECEIVED IP PACKETS INTO INTERLEAVE WINDOW
S21
DETERMINE DESTINATION TARP ADDRESS, INITIALIZE TTL, STORE IN TARP HEADER
S22
RECORD WINDOW SEQ. NOS. AND INTERLEAVE SEQ. NOS IN TARP HEADERS
S23
CHOOSE FIRST HOP TARP ROUTER, LOOK UP IP ADDRESS AND STORE IN CLEAR IP HEADER, OUTER LAYER ENCRYPT
S24
INSTALL CLEAR IP HEADER AND TRANSMIT
S25

FIG. 6

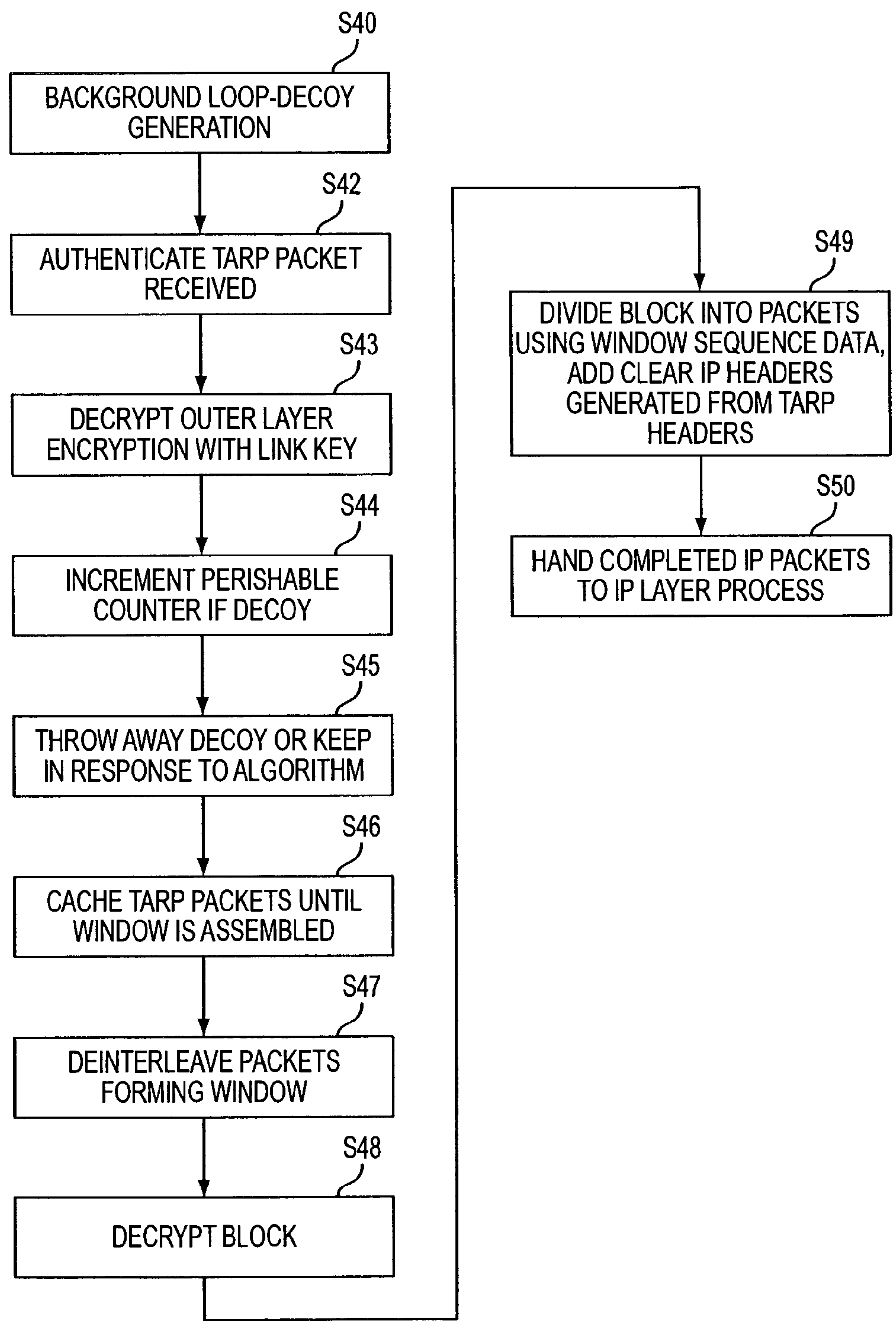

S40
BACKGROUND LOOP-DECOY GENERATION
S42
AUTHENTICATE TARP PACKET RECEIVED
S43
DECRYPT OUTER LAYER ENCRYPTION WITH LINK KEY
S44
INCREMENT PERISHABLE COUNTER IF DECOY
S45
THROW AWAY DECOY OR KEEP IN RESPONSE TO ALGORITHM
S46
CACHE TARP PACKETS UNTIL WINDOW IS ASSEMBLED
S47
DEINTERLEAVE PACKETS FORMING WINDOW
S48
DECRYPT BLOCK
S49
DIVIDE BLOCK INTO PACKETS USING WINDOW SEQUENCE DATA, ADD CLEAR IP HEADERS GENERATED FROM TARP HEADERS
S50
HAND COMPLETED IP PACKETS TO IP LAYER PROCESS

FIG. 7

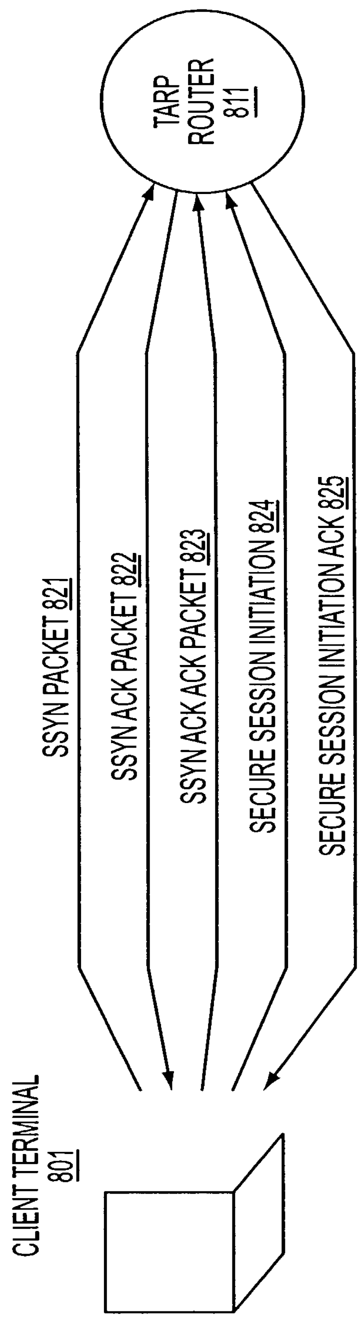
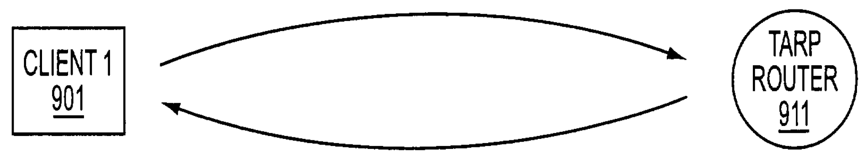

TARP ROUTER 811
SSYN PACKET 821
SSYN ACK PACKET 822
SSYN ACK ACK PACKET 823
SECURE SESSION INITIATION 824
SECURE SESSION INITIATION ACK 825
CLIENT TERMINAL 801

FIG. 8



TRANSMIT TABLE 921

131.218.204.98	,	131.218.204.65
131.218.204.221	,	131.218.204.97
131.218.204.139	,	131.218.204.186
131.218.204.12	,	131.218.204.55
⋮		⋮

RECEIVE TABLE 924

131.218.204.98	,	131.218.204.65
131.218.204.221	,	131.218.204.97
131.218.204.139	,	131.218.204.186
131.218.204.12	,	131.218.204.55
⋮		⋮

RECEIVE TABLE 922

131.218.204.161	,	131.218.204.89
131.218.204.66	,	131.218.204.212
131.218.204.201	,	131.218.204.127
131.218.204.119	,	131.218.204.49
⋮		⋮

TRANSMIT TABLE 923

131.218.204.161	,	131.218.204.89
131.218.204.66	,	131.218.204.212
131.218.204.201	,	131.218.204.127
131.218.204.119	,	131.218.204.49
⋮		⋮

FIG. 9

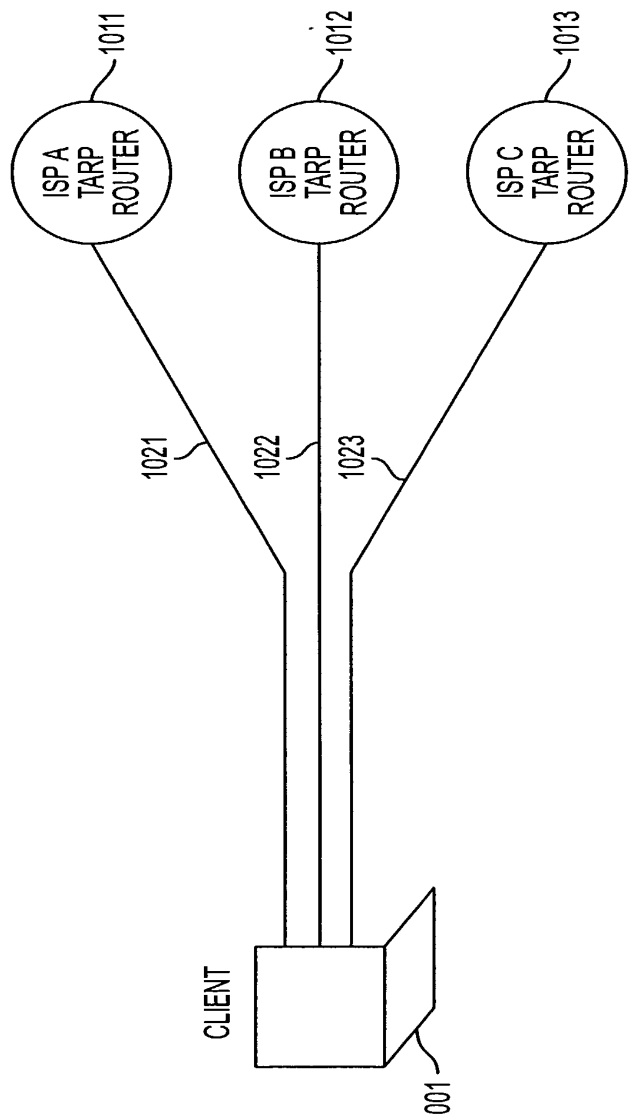

ISP A TARP ROUTER
1011
ISP B TARP ROUTER
1012
ISP C TARP ROUTER
1013
1021
1022
1023
CLIENT
1001

FIG. 10

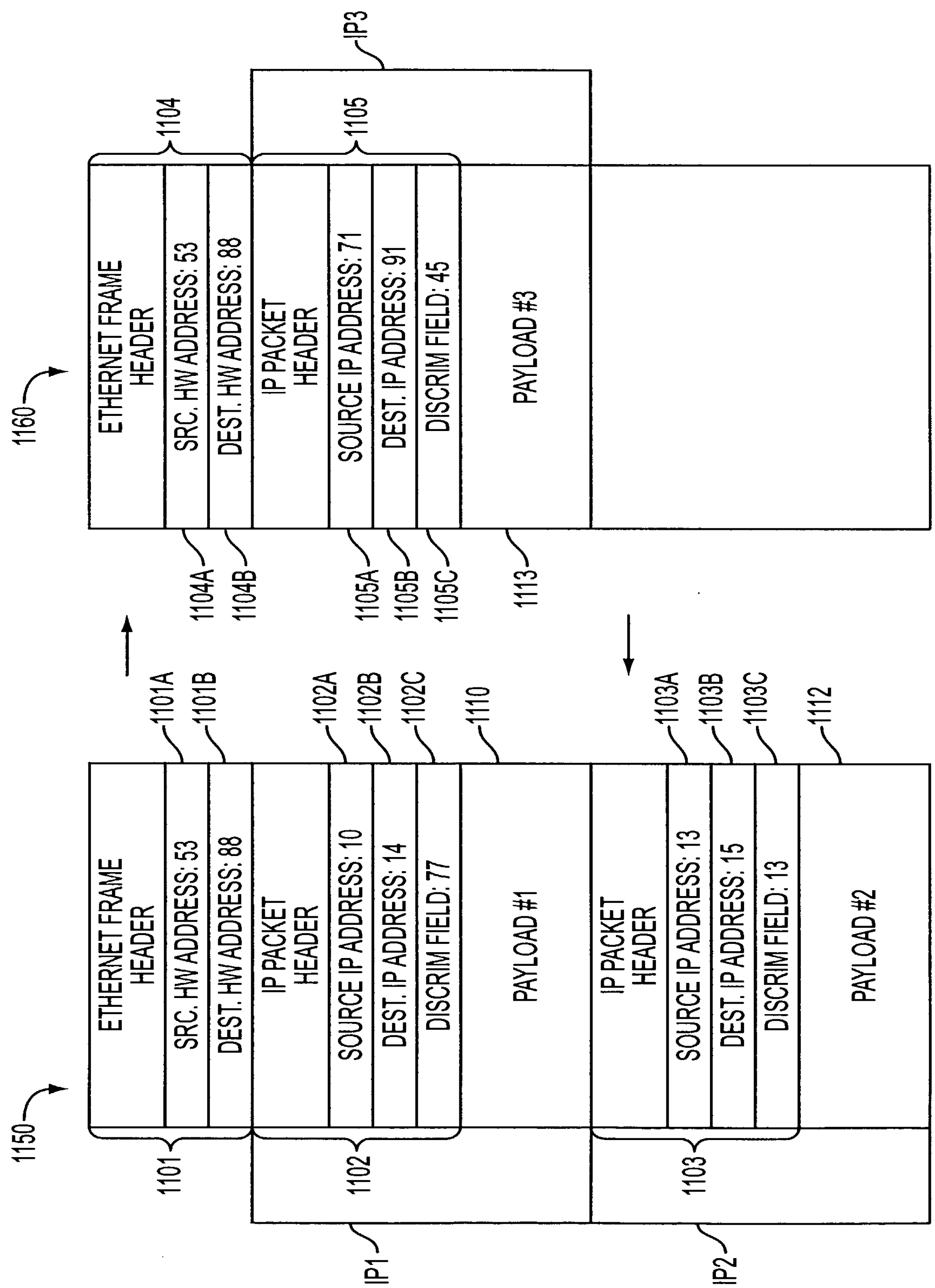

1150
1101
1102
1103
IP1
IP2
ETHERNET FRAME HEADER
SRC. HW ADDRESS: 53
DEST. HW ADDRESS: 88
IP PACKET HEADER
SOURCE IP ADDRESS: 10
DEST. IP ADDRESS: 14
DISCRIM FIELD: 77
PAYLOAD #1
IP PACKET HEADER
SOURCE IP ADDRESS: 13
DEST. IP ADDRESS: 15
DISCRIM FIELD: 13
PAYLOAD #2
1101A
1101B
1102A
1102B
1102C
1110
1103A
1103B
1103C
1112
1160
1104
1105
IP3
ETHERNET FRAME HEADER
SRC. HW ADDRESS: 53
DEST. HW ADDRESS: 88
IP PACKET HEADER
SOURCE IP ADDRESS: 71
DEST. IP ADDRESS: 91
DISCRIM FIELD: 45
PAYLOAD #3
1104A
1104B
1105A
1105B
1105C
1113

FIG. 11

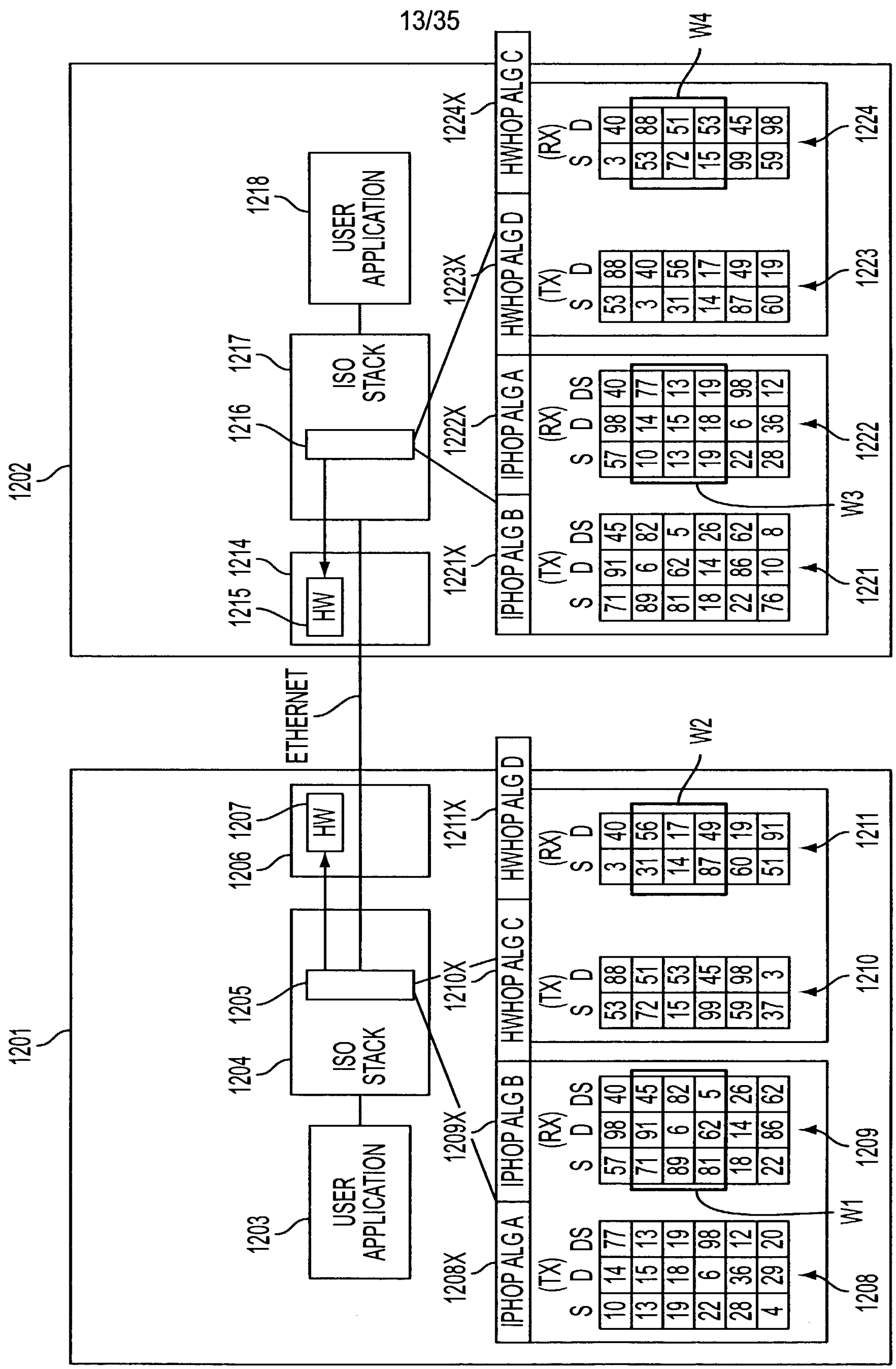
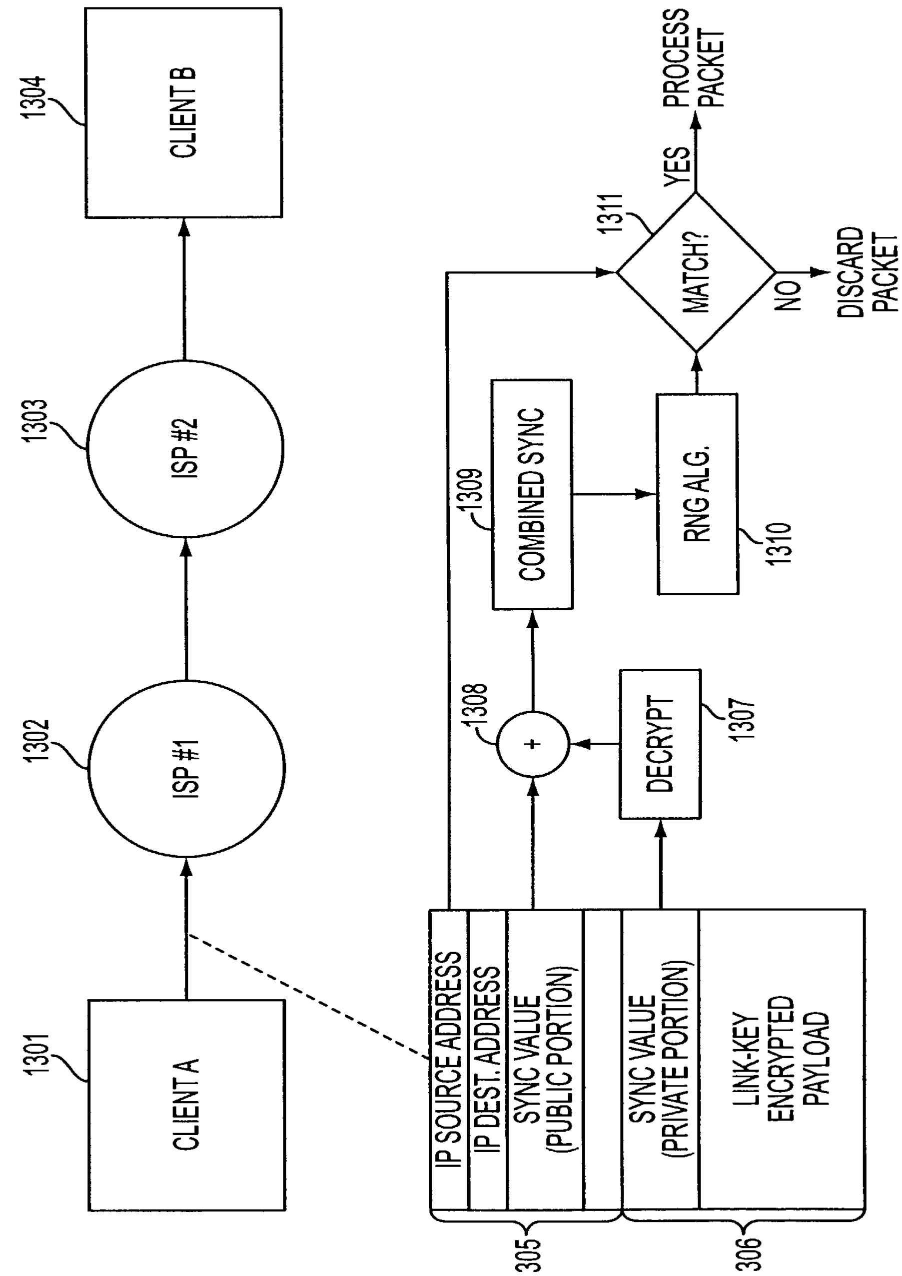

1201
1202
1203
USER APPLICATION
1204
ISO STACK
1205
1206
1207
HW
1208X
1209X
1210X
1211X
IPHOP ALG A
IPHOP ALG B
HWHOP ALG C
HWHOP ALG D
(TX)
(RX)
S D DS
S D
1208
1209
1210
1211
W1
W2
ETHERNET
1214
1215
1216
1217
1218
1221X
1222X
1223X
1224X
1221
1222
1223
1224
W3
W4

FIG. 12A

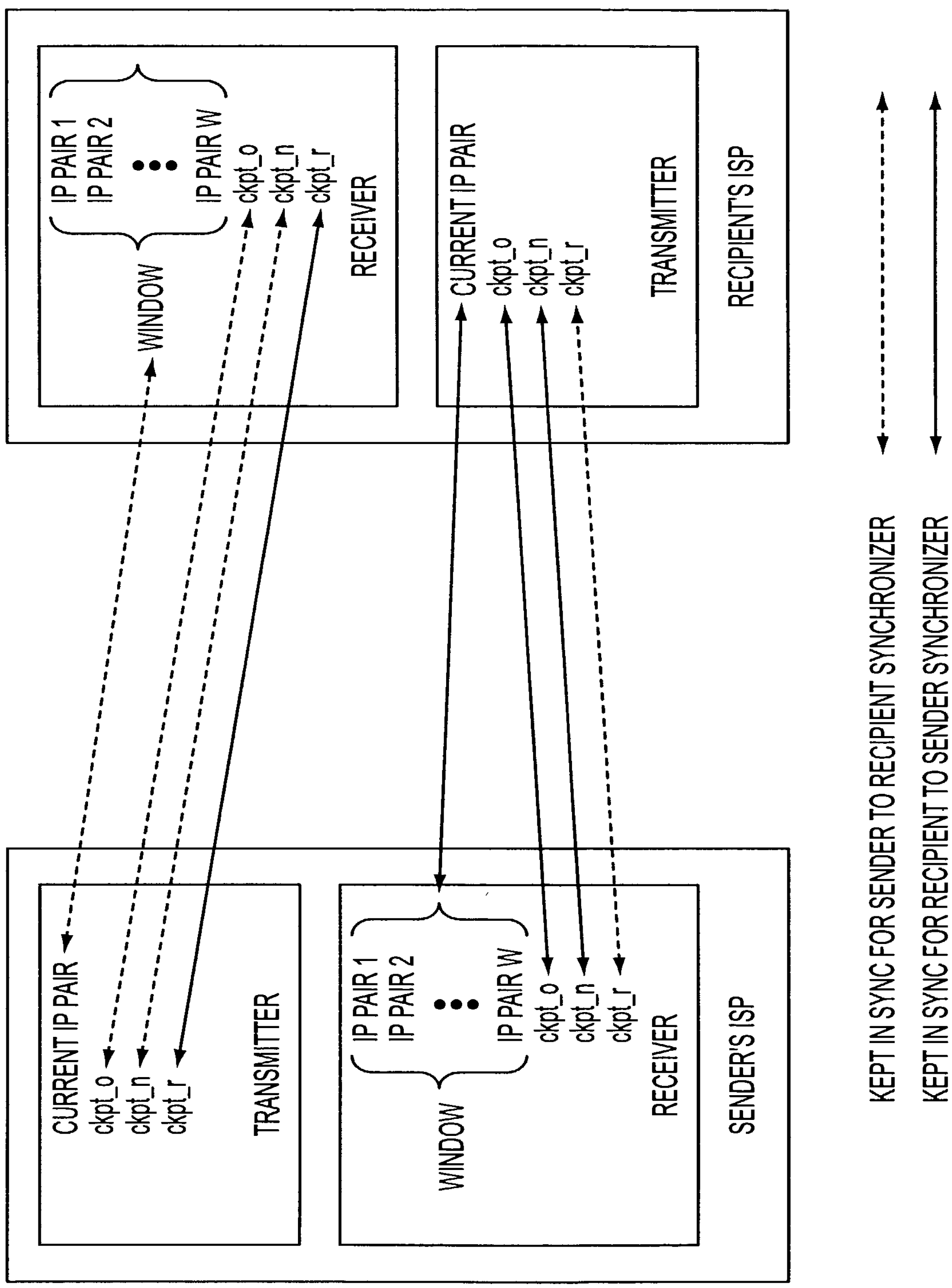
MODE OR EMBODIMENT	HARDWARE ADDRESSES	IP ADDRESSES	DISCRIMINATOR FIELD VALUES
1. PROMISCUOUS	SAME FOR ALL NODES OR COMPLETELY RANDOM	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
2. PROMISCUOUS PER VPN	FIXED FOR EACH VPN	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
3. HARDWARE HOPPING	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC

FIG. 12B



1301
CLIENT A
1302
ISP #1
1303
ISP #2
1304
CLIENT B
IP SOURCE ADDRESS
IP DEST. ADDRESS
SYNC VALUE (PUBLIC PORTION)
SYNC VALUE (PRIVATE PORTION)
LINK-KEY ENCRYPTED PAYLOAD
1305
1306
DECRYPT
1307
+
1308
COMBINED SYNC
1309
RNG ALG.
1310
MATCH?
1311
YES
PROCESS PACKET
NO
DISCARD PACKET

FIG. 13



WINDOW
IP PAIR 1
IP PAIR 2
IP PAIR W
ckpt_o
ckpt_n
ckpt_r
RECEIVER
CURRENT IP PAIR
ckpt_o
ckpt_n
ckpt_r
TRANSMITTER
RECIPIENT'S ISP
CURRENT IP PAIR
ckpt_o
ckpt_n
ckpt_r
TRANSMITTER
WINDOW
IP PAIR 1
IP PAIR 2
IP PAIR W
ckpt_o
ckpt_n
ckpt_r
RECEIVER
SENDER'S ISP
KEPT IN SYNC FOR SENDER TO RECIPIENT SYNCHRONIZER
KEPT IN SYNC FOR RECIPIENT TO SENDER SYNCHRONIZER

FIG. 14

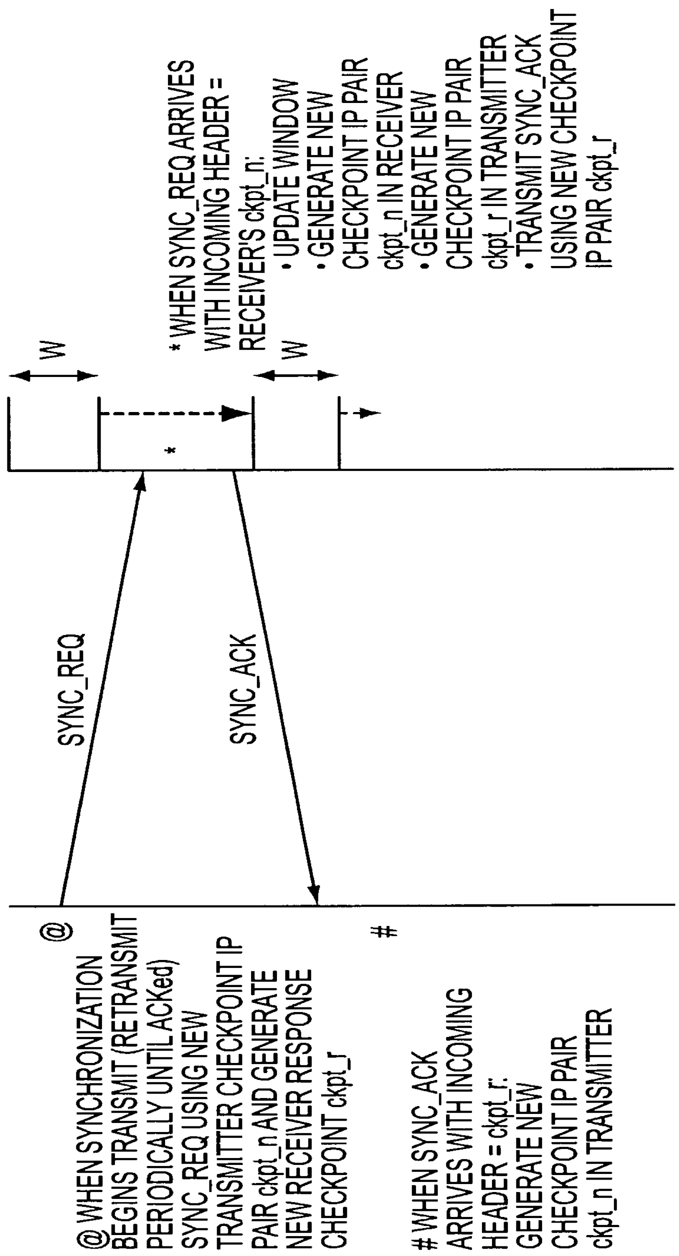

@
SYNC_REQ
SYNC_ACK
#
*
W
W
@ WHEN SYNCHRONIZATION BEGINS TRANSMIT (RETRANSMIT PERIODICALLY UNTIL ACKed) SYNC_REQ USING NEW TRANSMITTER CHECKPOINT IP PAIR ckpt_n AND GENERATE NEW RECEIVER RESPONSE CHECKPOINT ckpt_r
# WHEN SYNC_ACK ARRIVES WITH INCOMING HEADER = ckpt_r: GENERATE NEW CHECKPOINT IP PAIR ckpt_n IN TRANSMITTER
* WHEN SYNC_REQ ARRIVES WITH INCOMING HEADER = RECEIVER'S ckpt_n:
• UPDATE WINDOW
• GENERATE NEW CHECKPOINT IP PAIR ckpt_n IN RECEIVER
• GENERATE NEW CHECKPOINT IP PAIR ckpt_r IN TRANSMITTER
• TRANSMIT SYNC_ACK USING NEW CHECKPOINT IP PAIR ckpt_r

FIG. 15

10259494 .093002

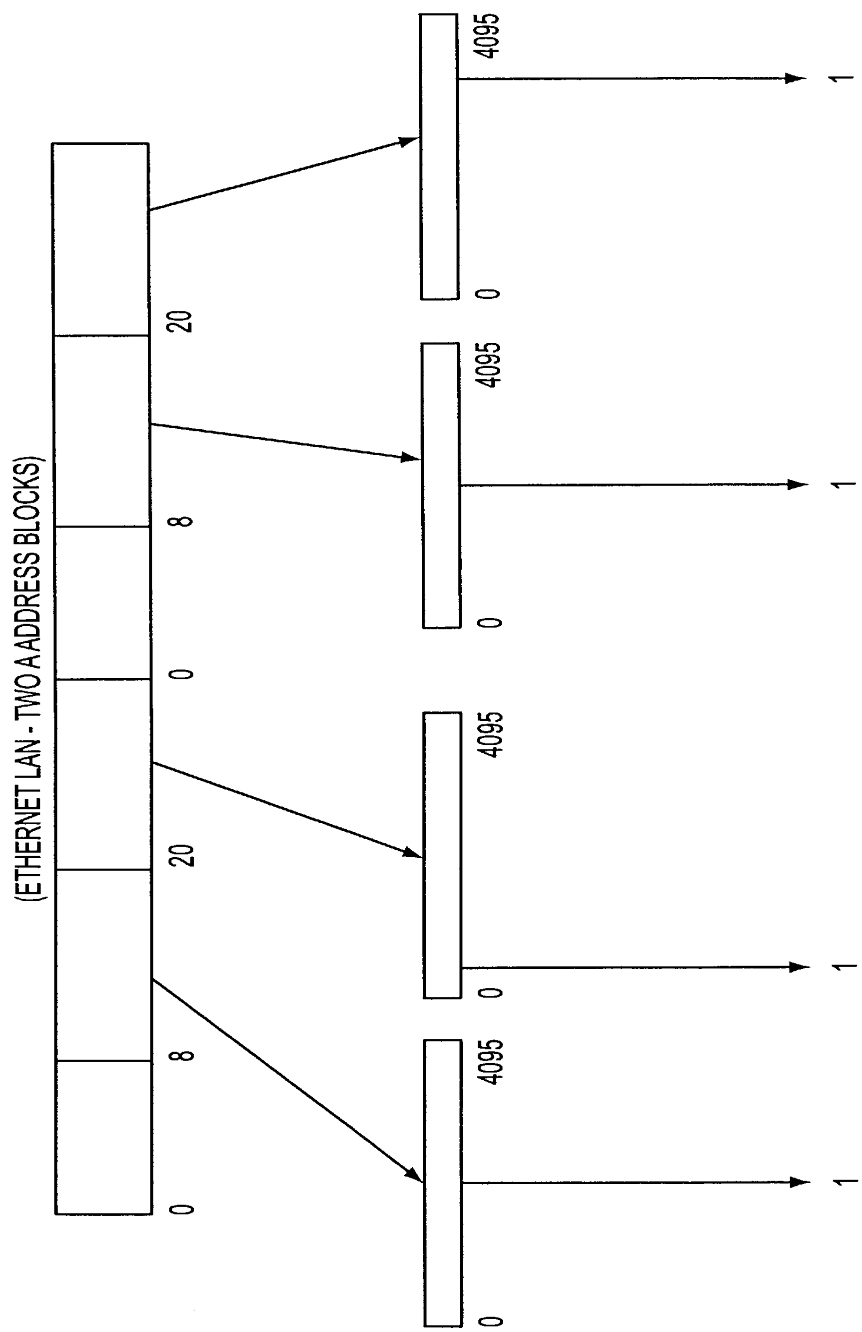


FIG. 16

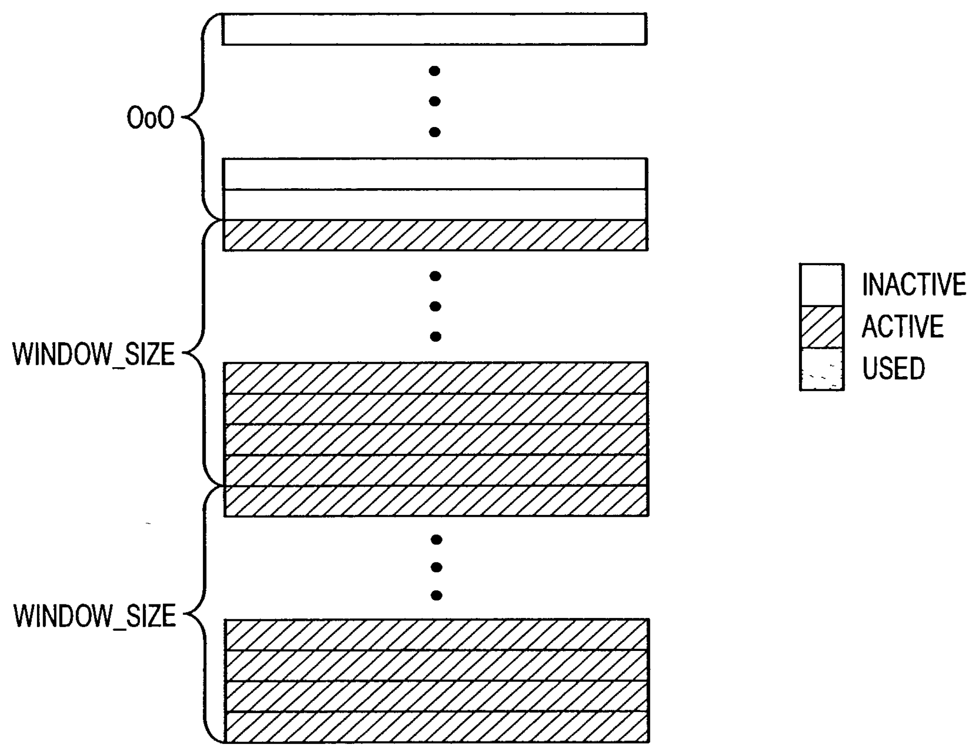


FIG. 17

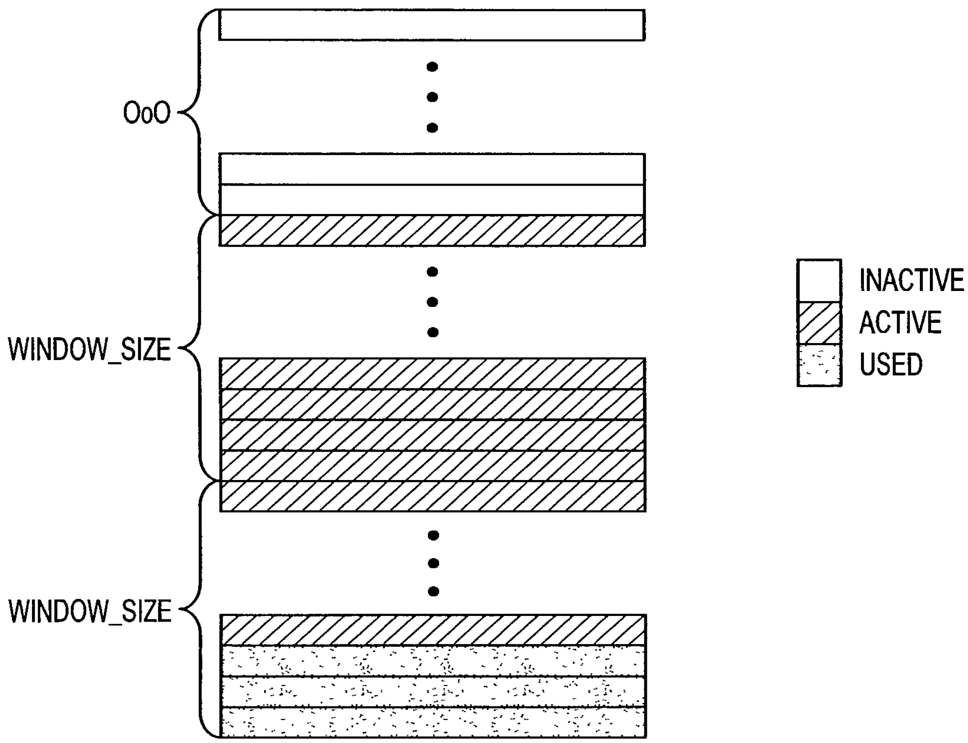


FIG. 18

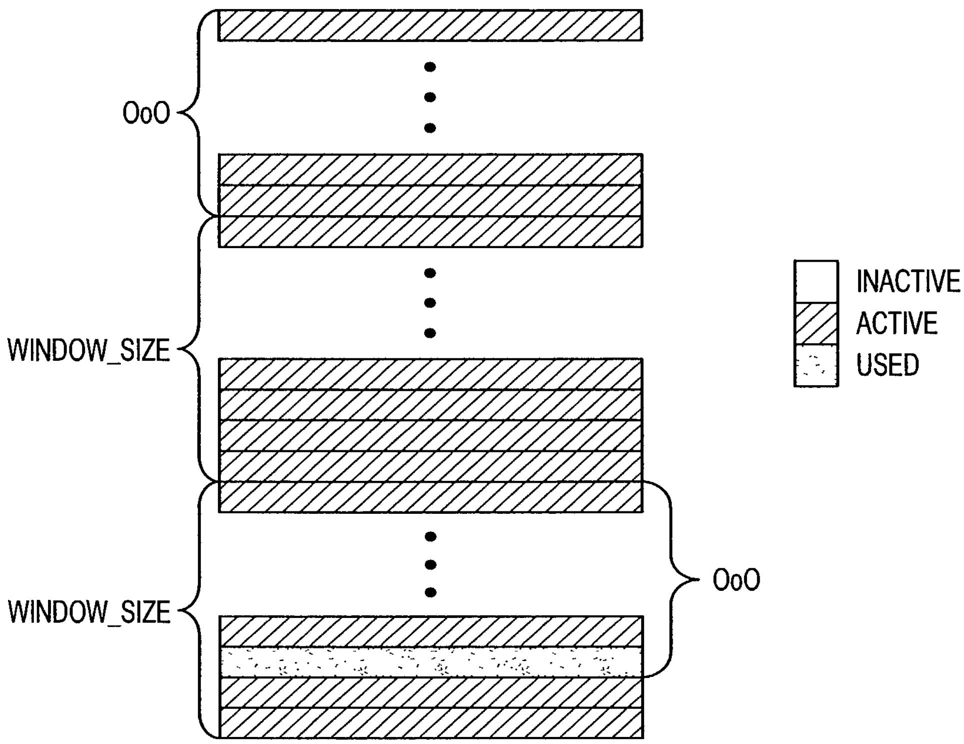


FIG. 19

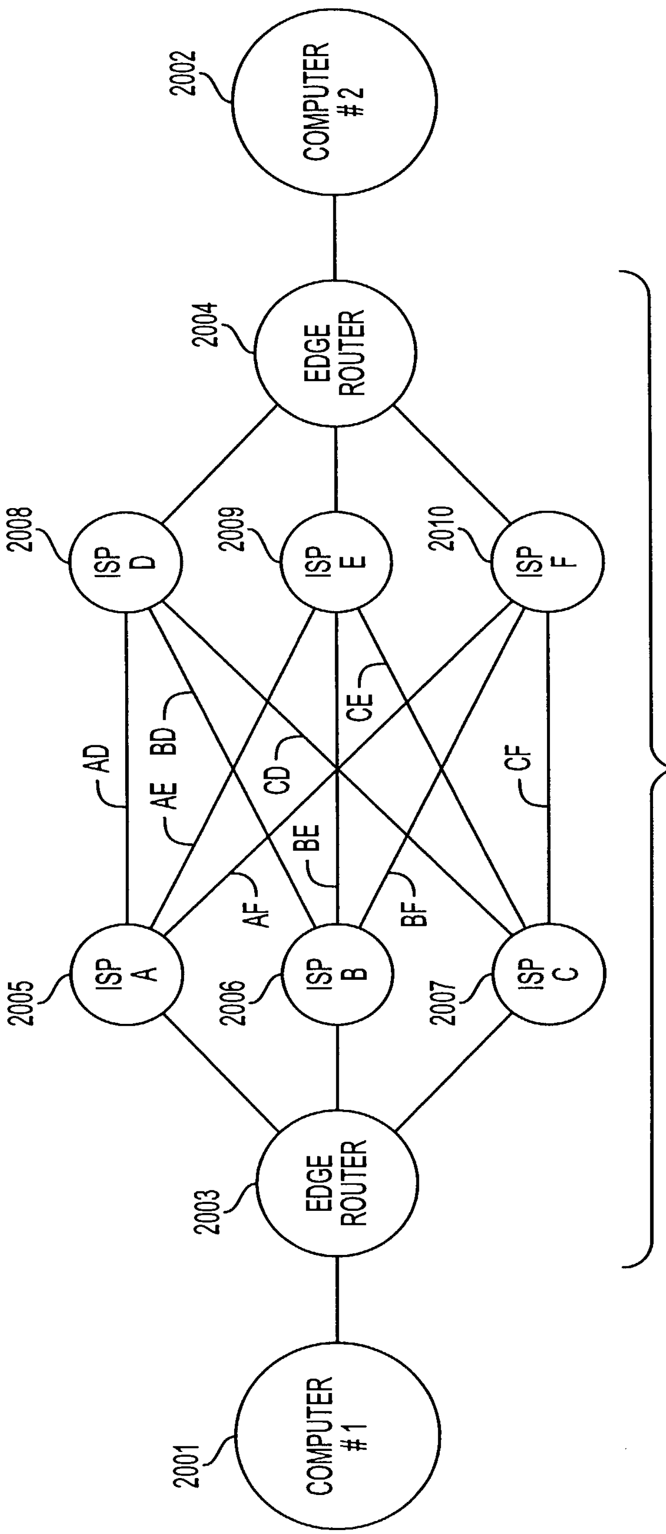

2001
COMPUTER #1
2002
COMPUTER #2
2003
EDGE ROUTER
2004
EDGE ROUTER
2005
ISP A
2006
ISP B
2007
ISP C
2008
ISP D
2009
ISP E
2010
ISP F
AD
AE
AF
BD
BE
BF
CD
CE
CF
2011

FIG. 20

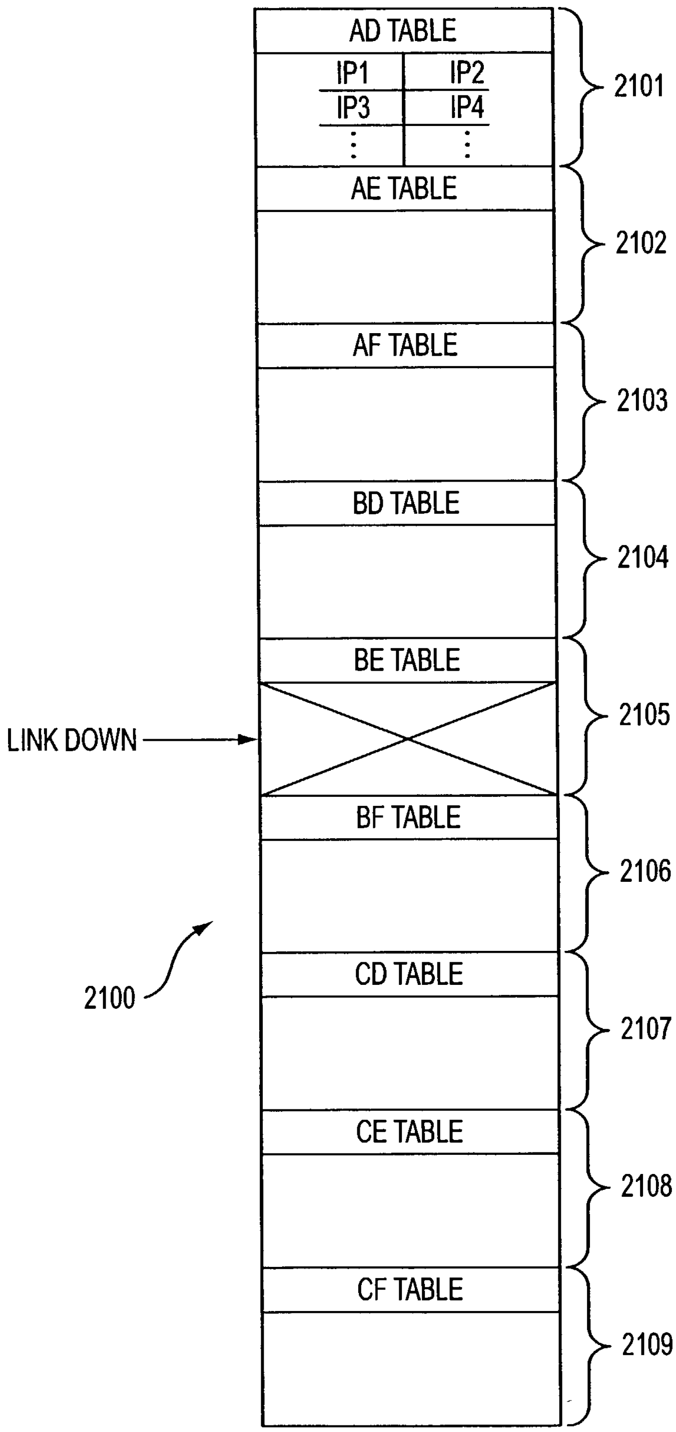
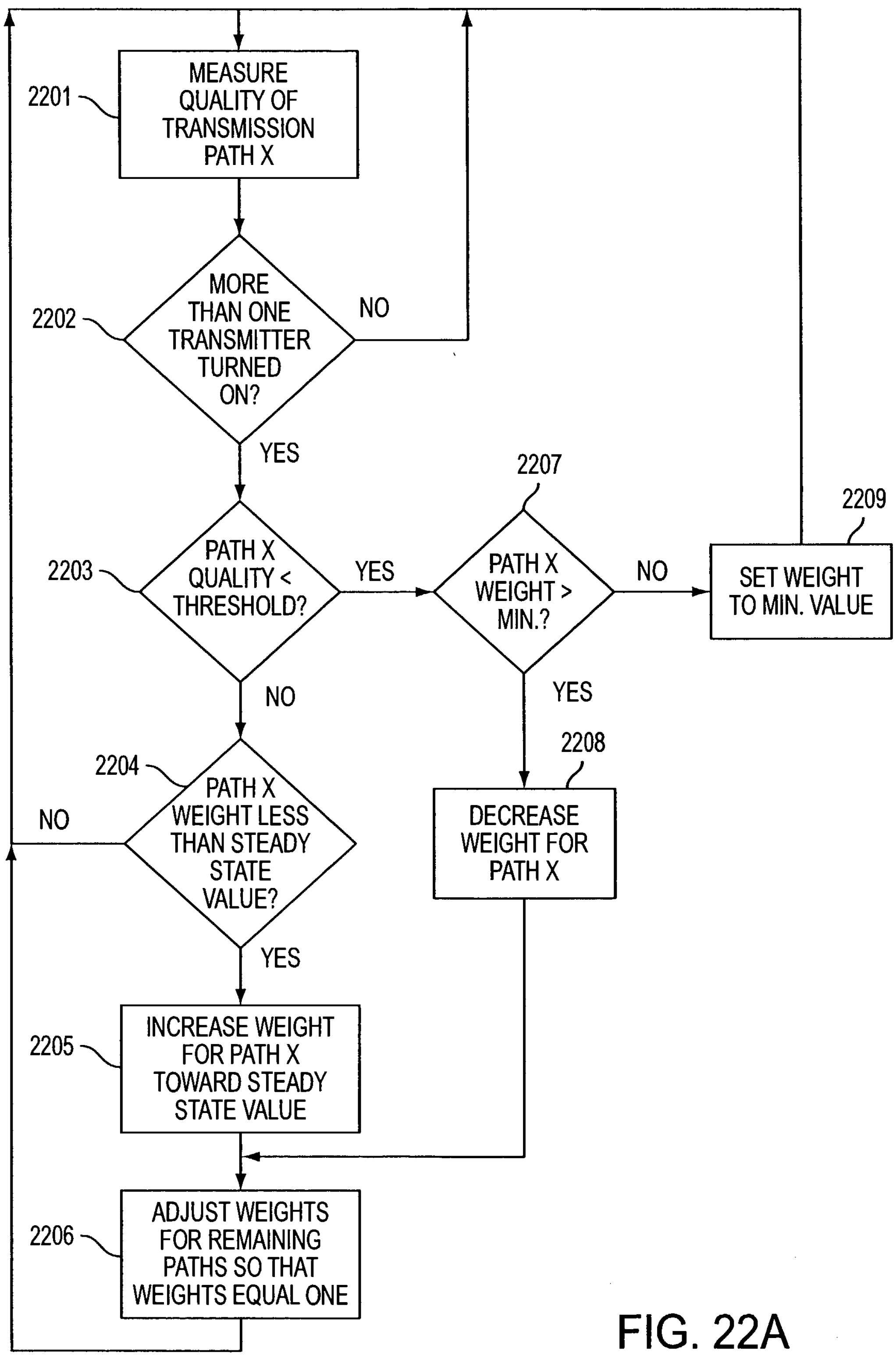

AD TABLE
IP1
IP2
IP3
IP4
AE TABLE
AF TABLE
BD TABLE
BE TABLE
BF TABLE
CD TABLE
CE TABLE
CF TABLE
LINK DOWN
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109

FIG. 21



MEASURE QUALITY OF TRANSMISSION PATH X
2201
MORE THAN ONE TRANSMITTER TURNED ON?
2202
NO
YES
PATH X QUALITY < THRESHOLD?
2203
YES
2207
PATH X WEIGHT > MIN.?
NO
2209
SET WEIGHT TO MIN. VALUE
NO
YES
2208
DECREASE WEIGHT FOR PATH X
2204
PATH X WEIGHT LESS THAN STEADY STATE VALUE?
NO
YES
INCREASE WEIGHT FOR PATH X TOWARD STEADY STATE VALUE
2205
ADJUST WEIGHTS FOR REMAINING PATHS SO THAT WEIGHTS EQUAL ONE
2206

FIG. 22A

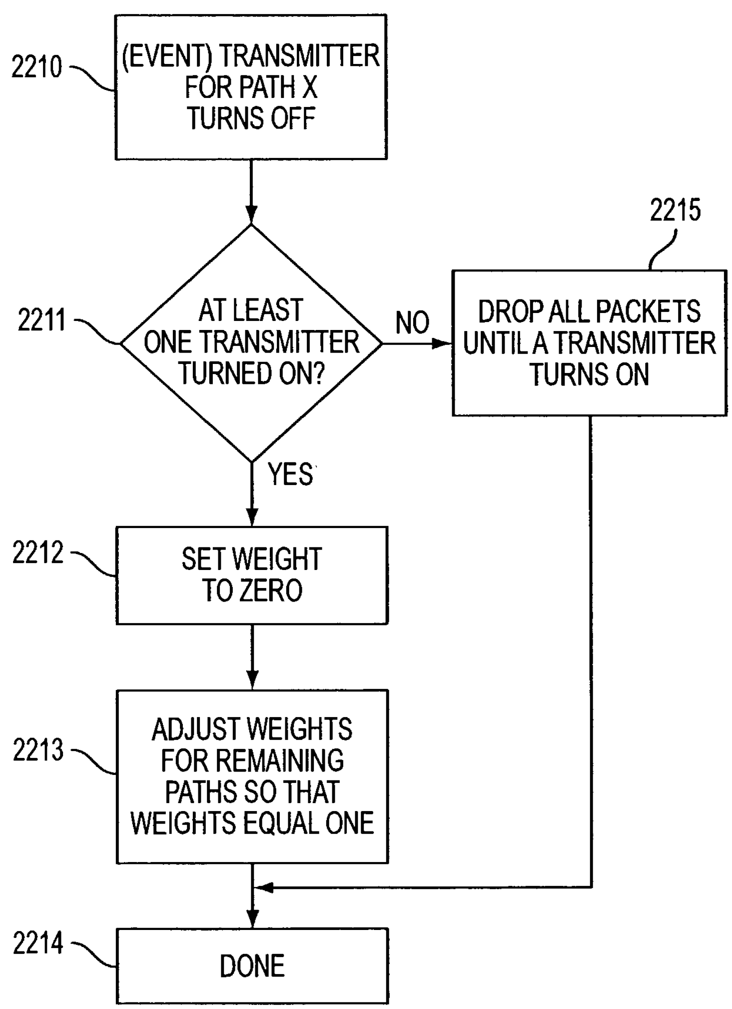

2210
(EVENT) TRANSMITTER FOR PATH X TURNS OFF
2211
AT LEAST ONE TRANSMITTER TURNED ON?
NO
2215
DROP ALL PACKETS UNTIL A TRANSMITTER TURNS ON
YES
2212
SET WEIGHT TO ZERO
2213
ADJUST WEIGHTS FOR REMAINING PATHS SO THAT WEIGHTS EQUAL ONE
2214
DONE

FIG. 22B

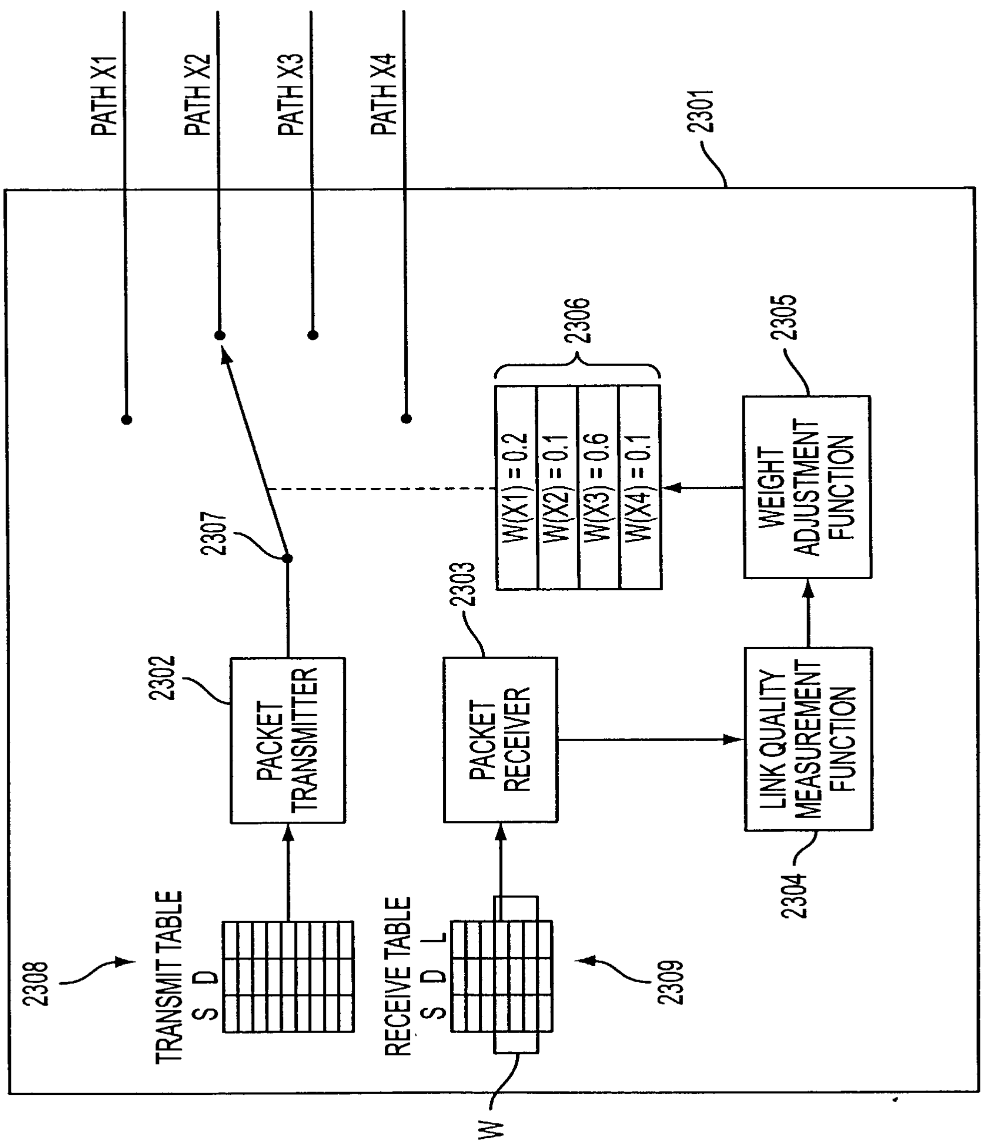

PATH X1
PATH X2
PATH X3
PATH X4
2301
2306
W(X1) = 0.2
W(X2) = 0.1
W(X3) = 0.6
W(X4) = 0.1
2305
WEIGHT ADJUSTMENT FUNCTION
2307
2303
2302
PACKET TRANSMITTER
PACKET RECEIVER
LINK QUALITY MEASUREMENT FUNCTION
2304
2308
TRANSMIT TABLE
S D
RECEIVE TABLE
S D L
2309
W

FIG. 23

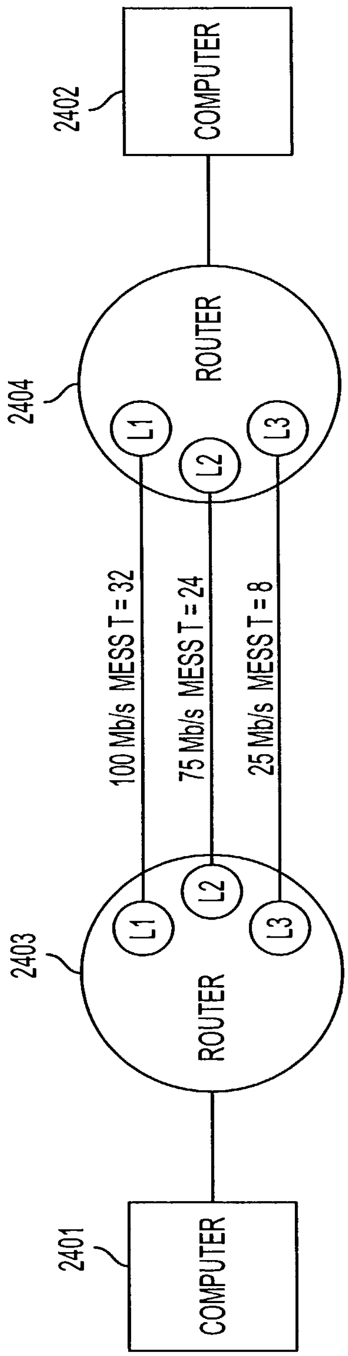

2402
COMPUTER
2404
ROUTER
L1
L2
L3
100 Mb/s MESS T = 32
75 Mb/s MESS T = 24
25 Mb/s MESS T = 8
2403
ROUTER
L1
L2
L3
2401
COMPUTER

FIG. 24

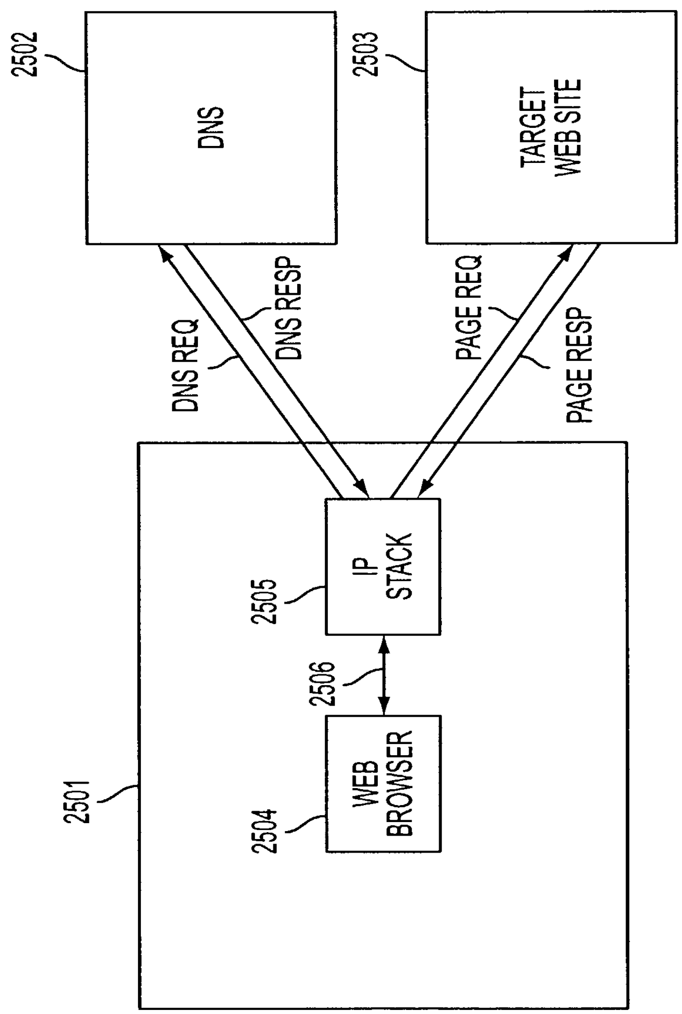


FIG. 25
(PRIOR ART)

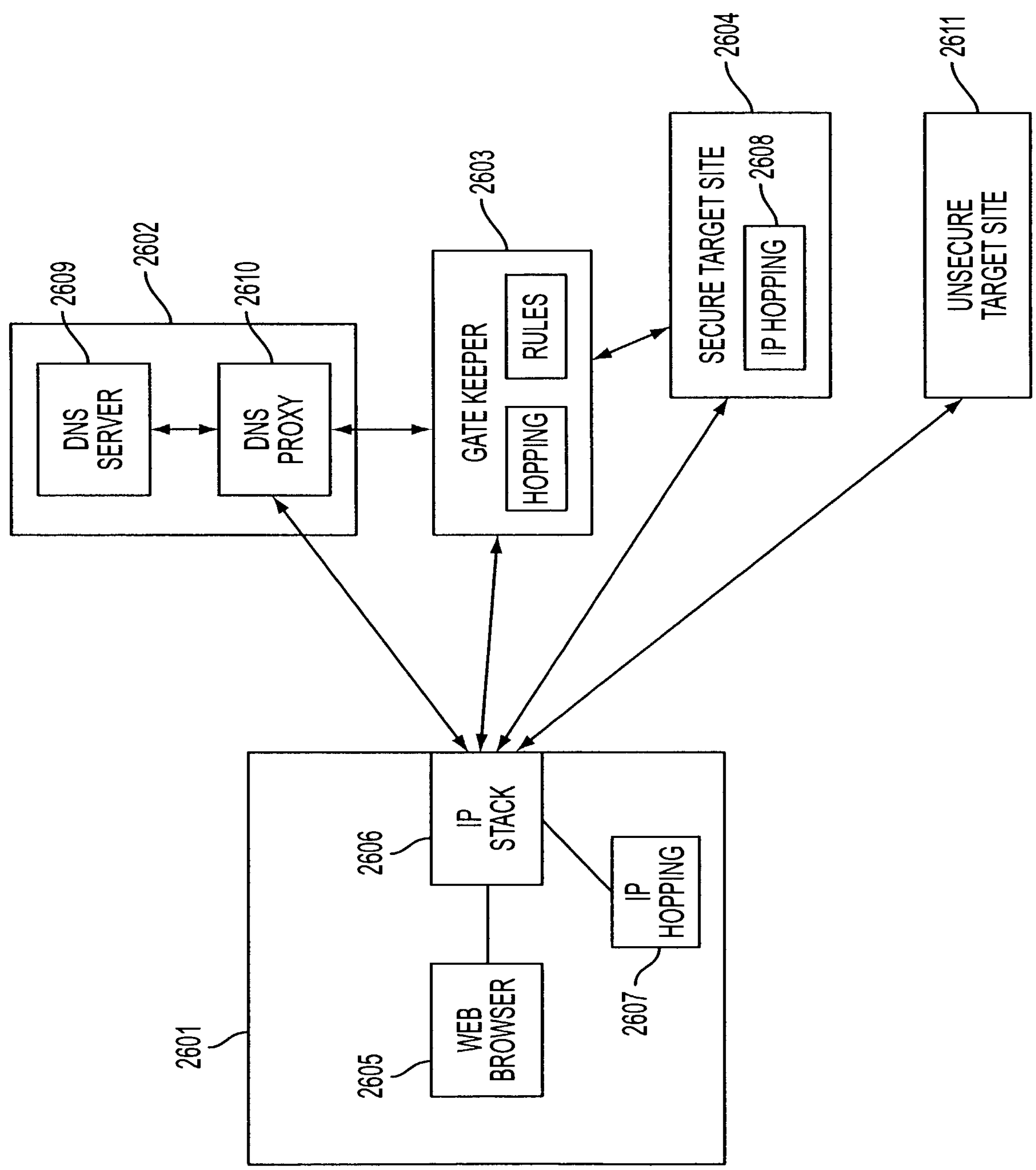

2601
WEB BROWSER
2605
IP STACK
2606
IP HOPPING
2607
DNS SERVER
2609
DNS PROXY
2610
2602
GATE KEEPER
HOPPING
RULES
2603
SECURE TARGET SITE
IP HOPPING
2608
2604
UNSECURE TARGET SITE
2611

FIG. 26

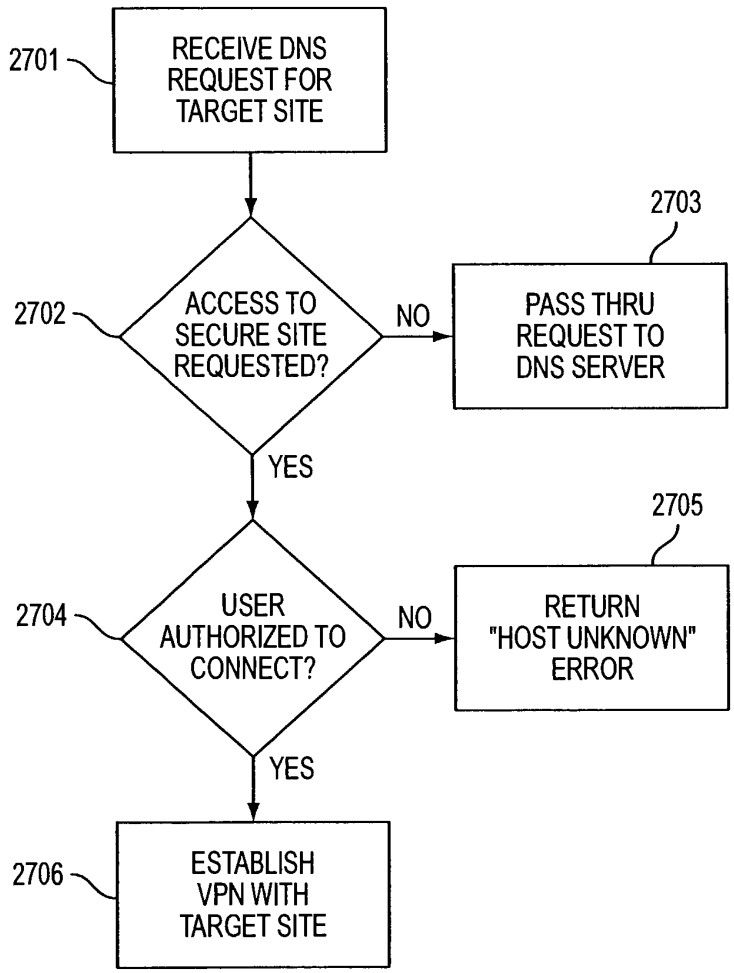

2701
RECEIVE DNS REQUEST FOR TARGET SITE
2702
ACCESS TO SECURE SITE REQUESTED?
NO
2703
PASS THRU REQUEST TO DNS SERVER
YES
2704
USER AUTHORIZED TO CONNECT?
NO
2705
RETURN "HOST UNKNOWN" ERROR
YES
2706
ESTABLISH VPN WITH TARGET SITE

FIG. 27

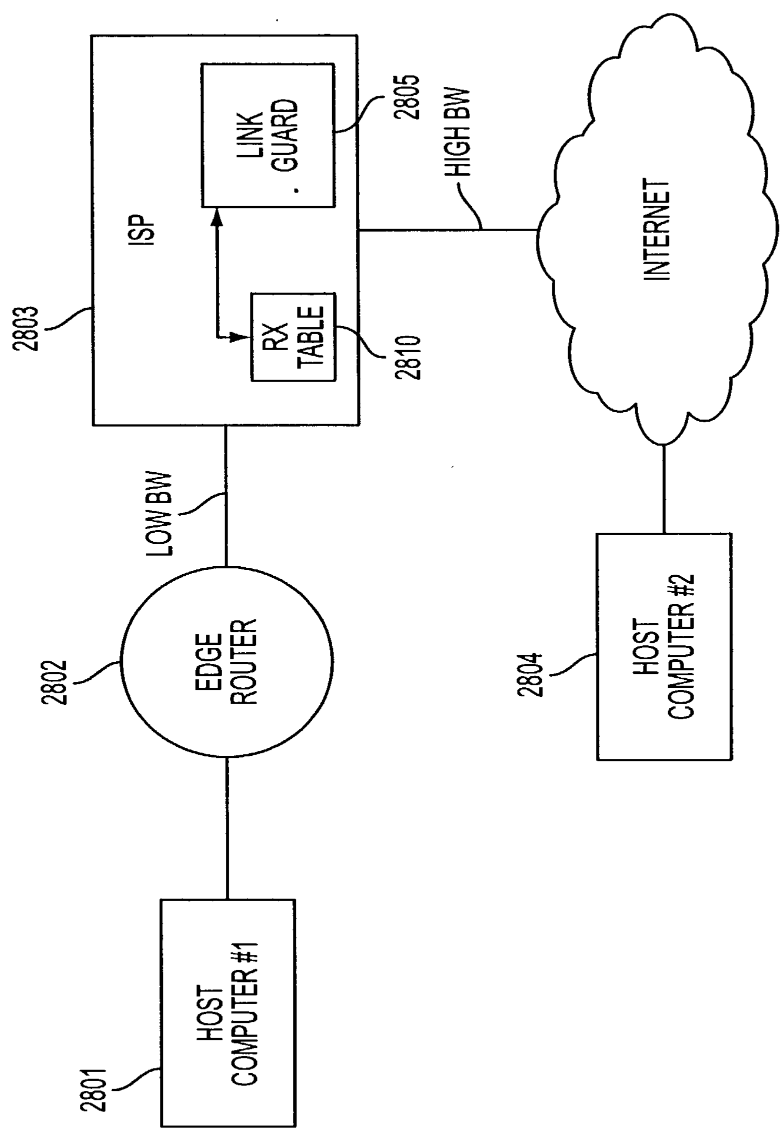
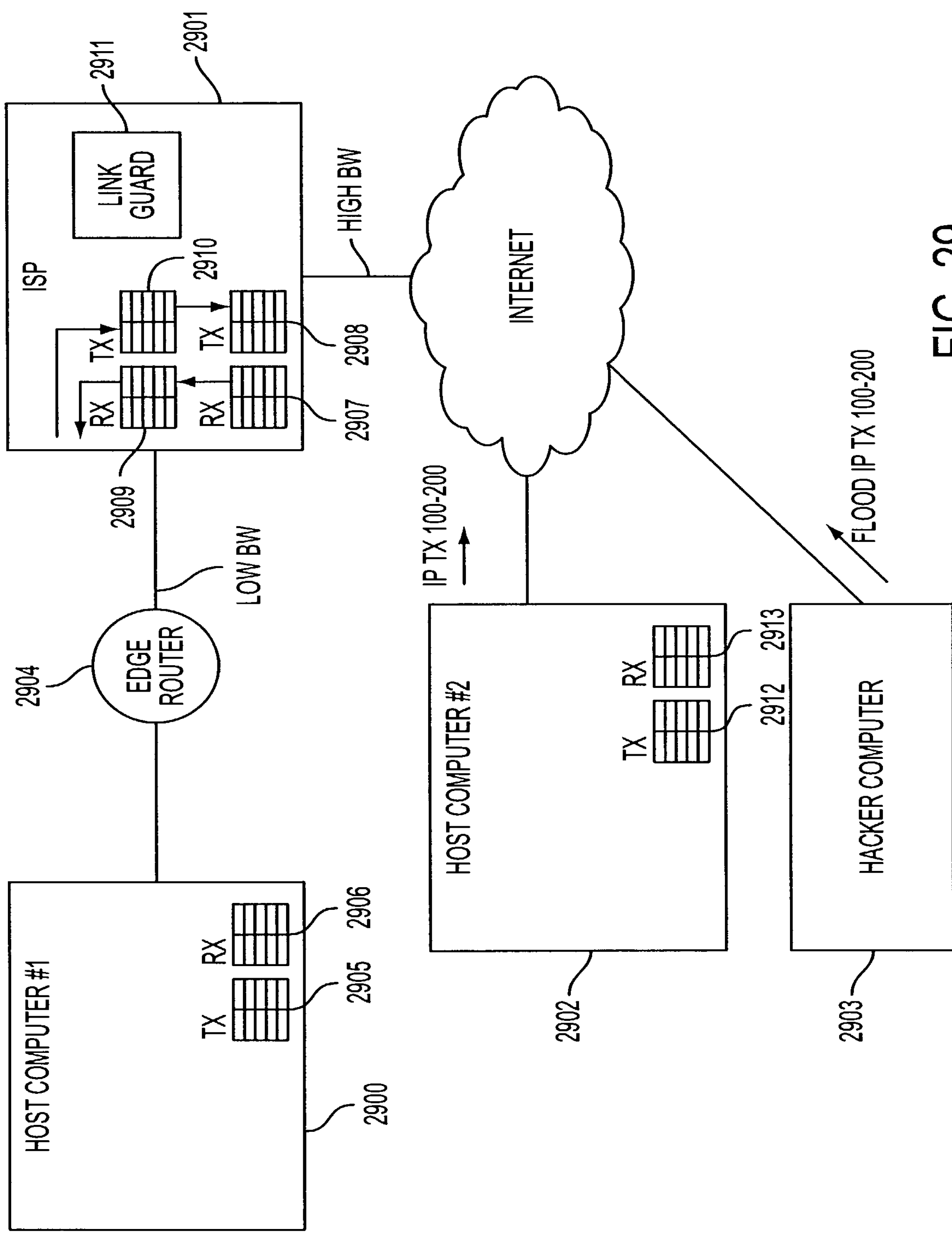

ISP
LINK
. GUARD
2805
RX
TABLE
2810
2803
HIGH BW
INTERNET
LOW BW
EDGE
ROUTER
2802
HOST
COMPUTER #2
2804
HOST
COMPUTER #1
2801

FIG. 28



HOST COMPUTER #1
TX
RX
2905
2906
2900
EDGE ROUTER
2904
LOW BW
ISP
LINK GUARD
2911
2901
TX
RX
TX
RX
2909
2910
2908
2907
HIGH BW
INTERNET
IP TX 100-200
FLOOD IP TX 100-200
HOST COMPUTER #2
TX
RX
2912
2913
2902
HACKER COMPUTER
2903

FIG. 29

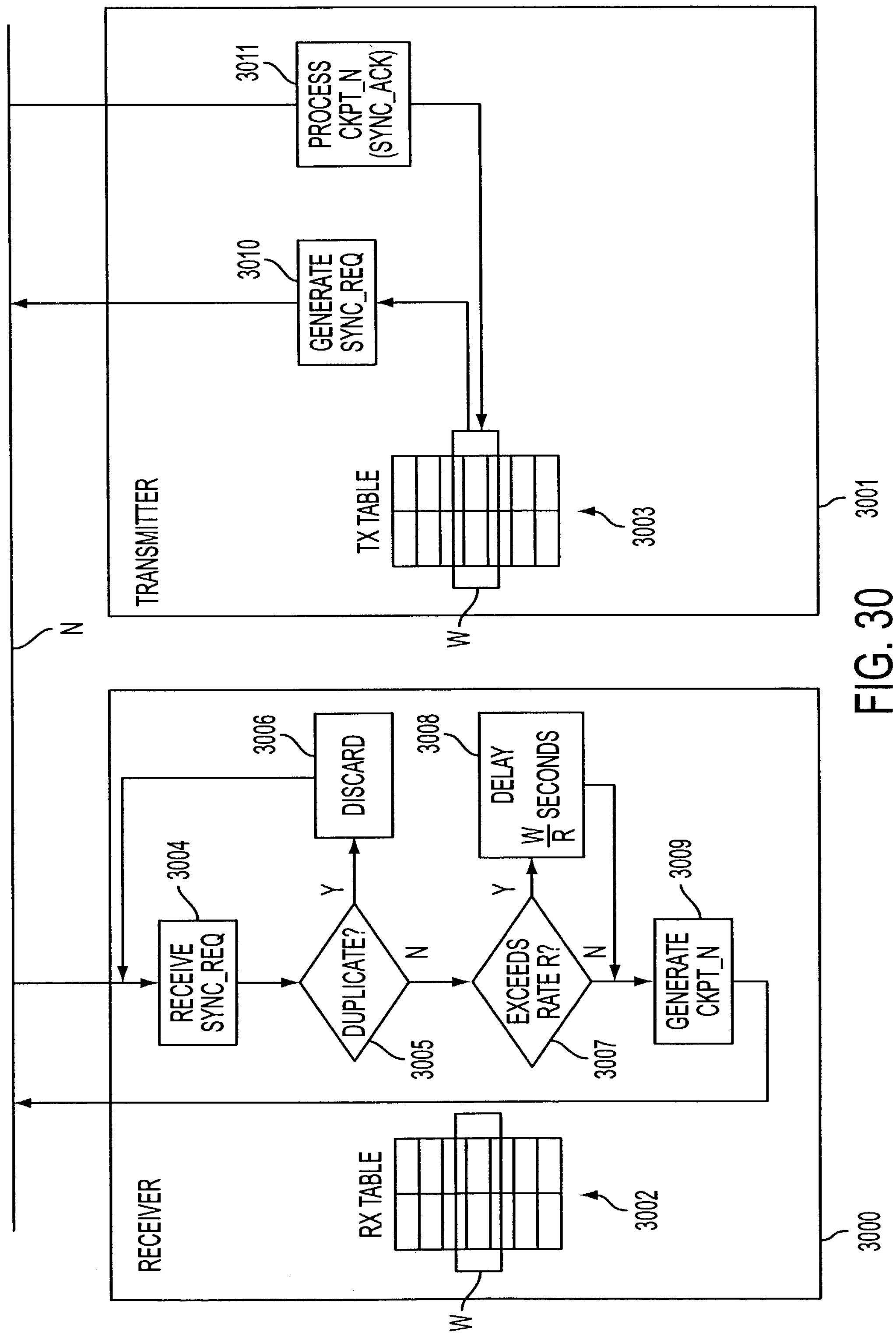

TRANSMITTER
3011
PROCESS CKPT_N (SYNC_ACK)
3010
GENERATE SYNC_REQ
TX TABLE
3003
3001
W
N
RECEIVER
RX TABLE
3002
3000
W
3004
RECEIVE SYNC_REQ
DUPLICATE?
3005
Y
N
3006
DISCARD
EXCEEDS RATE R?
3007
Y
N
3008
DELAY W/R SECONDS
3009
GENERATE CKPT_N

FIG. 30

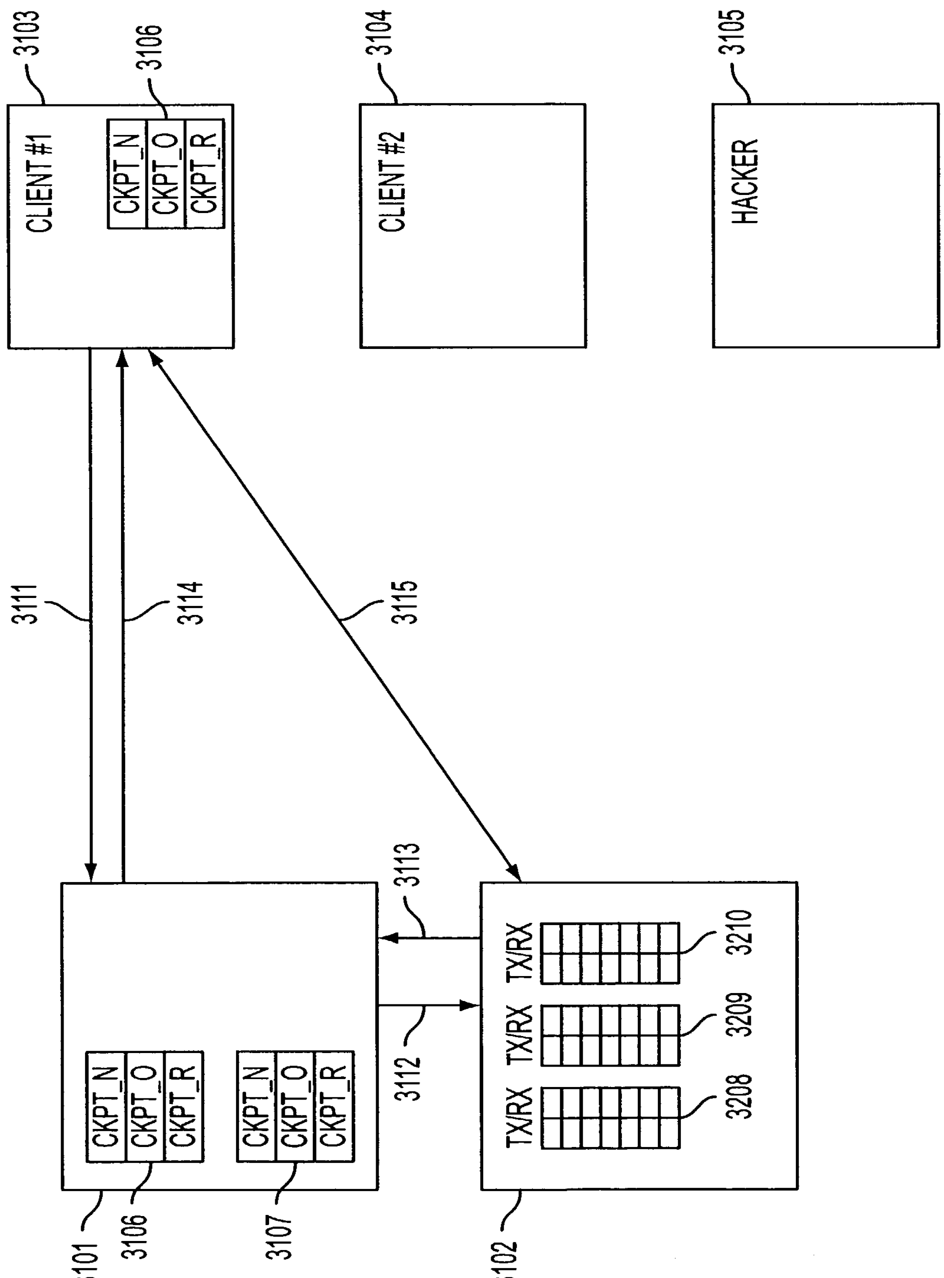

3103
3106
CLIENT #1
CKPT_N
CKPT_O
CKPT_R
3104
CLIENT #2
3105
HACKER
3111
3114
3115
3113
3112
CKPT_N
CKPT_O
CKPT_R
CKPT_N
CKPT_O
CKPT_R
TX/RX
TX/RX
TX/RX
3210
3209
3208
3101
3106
3107
3102

FIG. 31

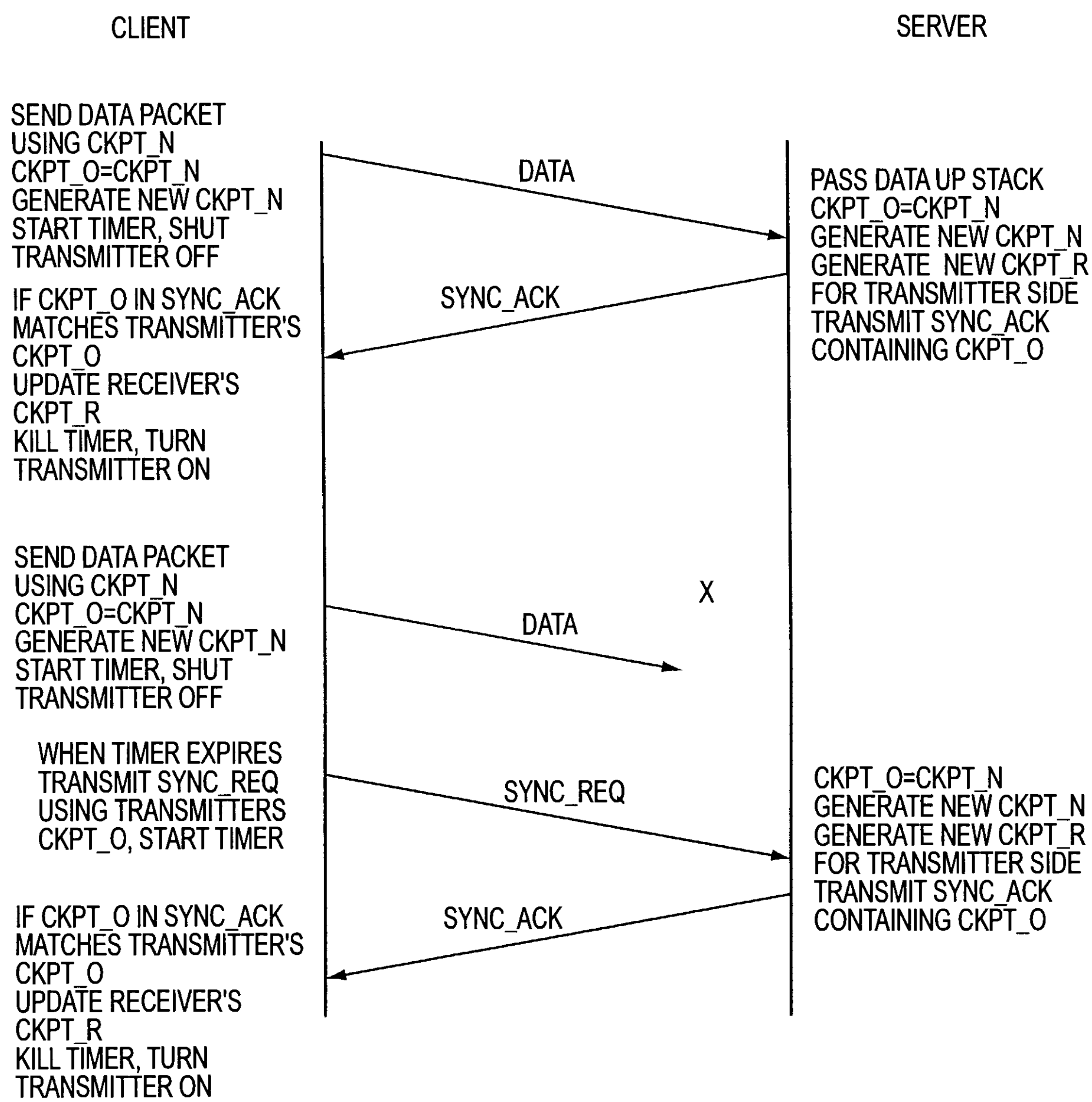

CLIENT
SERVER
SEND DATA PACKET
USING CKPT_N
CKPT_O=CKPT_N
GENERATE NEW CKPT_N
START TIMER, SHUT
TRANSMITTER OFF
DATA
PASS DATA UP STACK
CKPT_O=CKPT_N
GENERATE NEW CKPT_N
GENERATE NEW CKPT_R
FOR TRANSMITTER SIDE
TRANSMIT SYNC_ACK
CONTAINING CKPT_O
SYNC_ACK
IF CKPT_O IN SYNC_ACK
MATCHES TRANSMITTER'S
CKPT_O
UPDATE RECEIVER'S
CKPT_R
KILL TIMER, TURN
TRANSMITTER ON
SEND DATA PACKET
USING CKPT_N
CKPT_O=CKPT_N
GENERATE NEW CKPT_N
START TIMER, SHUT
TRANSMITTER OFF
DATA
X
WHEN TIMER EXPIRES
TRANSMIT SYNC_REQ
USING TRANSMITTERS
CKPT_O, START TIMER
SYNC_REQ
CKPT_O=CKPT_N
GENERATE NEW CKPT_N
GENERATE NEW CKPT_R
FOR TRANSMITTER SIDE
TRANSMIT SYNC_ACK
CONTAINING CKPT_O
SYNC_ACK
IF CKPT_O IN SYNC_ACK
MATCHES TRANSMITTER'S
CKPT_O
UPDATE RECEIVER'S
CKPT_R
KILL TIMER, TURN
TRANSMITTER ON

FIG. 32

Attorney Docket No. 00479.85672

JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

■ is attached hereto.

☐ was filed on ______ as Application Serial Number ________ and was amended on ________ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed Under 35 U.S.C. 119	Certified Copy Attached
				Yes / No	Yes / No

Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s)	Filing Date (MM/DD/YYYY)	☐ Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

Attorney Docket No. 00479.85672

We hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial Number	Date of Filing (Day, Month, Year)	Status Patented, Pending, Abandoned
09/429,643	10/29/99	Pending

Power of Attorney

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Altherr, Reg. No. 31,810, Donald W. Banner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.
Eleventh Floor
1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature________________________ Date________________

Full Name of Joint Inventor: MUNGER (Family Name) Edmund (First Given Name) Colby (Second Given Name)

Residence: 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship: U.S.

Post Office Address: 1101 Opaca Court, Crownsville, Maryland 21032

Attorney Docket No. 00479.85672

Signature________________________ Date________________

Full Name of
Joint Inventor SCHMIDT Douglas Charles
Family Name First Given Name Second Given Name

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.
Post Office
Address 230 Oak Court, Severna Park, Maryland 21146

Signature [signature] Date 2/14/00

Full Name of
Joint Inventor SHORT Robert Dunham, III
Family Name First Given Name Second Given Name

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.
Post Office
Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature [signature] Date 2/14/2000

Full Name of
Joint Inventor LARSON Victor
Family Name First Given Name Second Given Name

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.
Post Office
Address 12026 Lisa Marie Court, Fairfax, Virginia 22033

Attorney Docket No. 00479.85672

Signature [signature] Date 2/14/2000

Full Name of Joint Inventor WILLIAMSON Michael

Family Name First Given Name Second Given Name

Residence 26203 Ocala Circle, South Riding, Virginia 20152

Citizenship U.S.

Post Office Address 26203 Ocala Circle, South Riding, Virginia 20152

LAW OFFICES
BANNER & WITCOFF, LTD.
1001 G STREET, N.W.
WASHINGTON, D.C. 20001-4597
(202) 508-9100

Attorney Docket No. 00479.85672

JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

■ is attached hereto.

☐ was filed on ______ as Application Serial Number ________ and was amended on ________ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed Under 35 U.S.C. 119	Certified Copy Attached
				Yes / No	Yes / No

Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s)	Filing Date (MM/DD/YYYY)	☐ Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

We hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial Number	Date of Filing (Day, Month, Year)	Status: Patented, Pending, Abandoned
09/429,643	10/29/99	Pending

Power of Attorney

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Altherr, Reg. No. 31,810, Donald W. Banner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.
Eleventh Floor
1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature__________ Date__________

Full Name of Joint Inventor: MUNGER (Family Name) — Edmund (First Given Name) — Colby (Second Given Name)

Residence: 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship: U.S.

Post Office Address: 1101 Opaca Court, Crownsville, Maryland 21032

Attorney Docket No. 00479.85672

Signature [signature] Date 2/14/00

Full Name of
Joint Inventor SCHMIDT Douglas Charles
Family Name First Given Name Second Given Name

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.
Post Office
Address 230 Oak Court, Severna Park, Maryland 21146

Signature Date

Full Name of
Joint Inventor SHORT Robert Dunham, III
Family Name First Given Name Second Given Name

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.
Post Office
Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature Date

Full Name of
Joint Inventor LARSON Victor
Family Name First Given Name Second Given Name

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.
Post Office
Address 12026 Lisa Marie Court, Fairfax, Virginia 22033

Attorney Docket No. 00479.85672

Signature________________________ Date________________________

Full Name of Joint Inventor: WILLIAMSON (Family Name) | Michael (First Given Name) | (Second Given Name)

Residence: 26203 Ocala Circle, South Riding, Virginia 20152

Citizenship: U.S.

Post Office Address: 26203 Ocala Circle, South Riding, Virginia 20152

LAW OFFICES
BANNER & WITCOFF, LTD.
1001 G STREET, N.W.
WASHINGTON, D.C. 20001-4597
(202) 508-9100

Attorney Docket No. 00479.85672

JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

■ is attached hereto.

☐ was filed on ______ as Application Serial Number ________ and was amended on ________ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed Under 35 U.S.C. 119	Certified Copy Attached
				Yes / No	Yes / No

Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s)	Filing Date (MM/DD/YYYY)	☐ Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

Attorney Docket No. 00479.85672

We hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT International filing date of this application:

Application Serial Number	Date of Filing (Day, Month, Year)	Status: Patented, Pending, Abandoned
09/429,643	10/29/99	Pending

Power of Attorney

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Altherr, Reg. No. 31,810, Donald W. Banner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.
Eleventh Floor
1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature: Edmund Colby Munger Date: 15 FEB 2000

Full Name of Joint Inventor: MUNGER (Family Name) Edmund (First Given Name) Colby (Second Given Name)

Residence: 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship: U.S.

Post Office Address: 1101 Opaca Court, Crownsville, Maryland 21032

Attorney Docket No. 00479.85672

Signature______________________ Date______________________

Full Name of Joint Inventor: SCHMIDT (Family Name), Douglas (First Given Name), Charles (Second Given Name)

Residence: 230 Oak Court, Severna Park, Maryland 21146

Citizenship: U.S.

Post Office Address: 230 Oak Court, Severna Park, Maryland 21146

Signature______________________ Date______________________

Full Name of Joint Inventor: SHORT (Family Name), Robert (First Given Name), Dunham, III (Second Given Name)

Residence: 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship: U.S.

Post Office Address: 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature______________________ Date______________________

Full Name of Joint Inventor: LARSON (Family Name), Victor (First Given Name), (Second Given Name)

Residence: 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship: U.S.

Post Office Address: 12026 Lisa Marie Court, Fairfax, Virginia 22033

Attorney Docket No. 00479.85672

Signature________________________ Date________________

Full Name of Joint Inventor: WILLIAMSON (Family Name) — Michael (First Given Name) — (Second Given Name)

Residence: 26203 Ocala Circle, South Riding, Virginia 20152

Citizenship: U.S.

Post Office Address: 26203 Ocala Circle, South Riding, Virginia 20152

LAW OFFICES
BANNER & WITCOFF, LTD.
1001 G STREET, N.W.
WASHINGTON, D.C. 20001-4597
(202) 508-9100

10/259494 U.S. PTO 09/30/02

U.S. UTILITY Patent Application

PATENT NUMBER and ISSUE DATE

APPL NUM	FILING DATE	CLASS	SUBCLASS	GAU	EXAMINER
10259494	09/30/2002	709	100	2153	Lim

**APPLICANTS: Munger Edmund; Schmidt Douglas; Short Robert; Larson Victor; Williamson Michael;

2153

**CONTINUING DATA VERIFIED:

This application is a DIV of 09/504,783 02/15/2000 now U.S. Patent No 6,502,135
which is a CIP of 09/429,643 10/29/1999
which claims benefit of 60/106,261 10/30/1998
and claims benefit of 60/137,704 06/07/1999

** FOREIGN APPLICATIONS VERIFIED:

BEST AVAILABLE COPY

PG-PUB	DO NOT PUBLISH ☐	RESCIND ☐

Foreign priority claimed ☐ yes ☑ no
35 USC 119 conditions met ☐ yes ☑ no
Verified and Acknowledged Examiners's initials KL

ATTORNEY DOCKET NO 000479.00082

TITLE : Agile network protocol for secure communications with assured system availability

U.S.DEPT. OF COMM./PAT.& TM-PTO-436L(Rev. 12-94)

NOTICE OF ALLOWANCE MAILED

Assistant Examiner

CLAIMS ALLOWED: Total Claims | Print Claim for O.G

ISSUE FEE: Amount Due | Date Paid

DRAWING: Sheets Drwg. | Figs.Drwg. | Print Fig.

Primary Examiner

Application Examiner

☐ TERMINAL DISCLAMER

PREPARED FOR ISSUE

WARNING: The information disclosed herein may be restricted. Unauthorized disclosure may be prohibited by the United States Code Title 35, Sections 122, 181 and 368, Possession outside the U.S. Patent & Trademark Office is restricted to authorized employees and contractors only.

FILED WITH: ☐ DISK (CRF) ☐ CD-ROM (Attached in pocket on right inside flap)

BEST AVAILABLE COPY

SEARCH

Class	Sub.	Date	Exmr.
709	217 218 219 229	6/22/04	KC
713	201		

SEARCH NOTES

(List databases searched. Attach search strategy inside.)

	Date	Exmr.
East	2/22/04 6/18/04	KC

INTERFERENCE SEARCHED

Class	Sub.	Date	Exmr.

ISSUE SLIP STAPLE AREA (for additional cross-references)

ISSUING CLASSIFICATION

ORIGINAL		CROSS REFERENCE(S)							
CLASS	SUBCLASS	CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)						
INTERNATIONAL CLASSIFICATION									
	/								
	/								
	/								
	/								
	/								

∧ Continued on Issue Slip Inside File Jacket

INDEX OF CLAIMS

✓ .... Rejected – (Through numeral) ... Canceled N .... Non-elected A .... Appeal
= .... Allowed + ... Restricted I .... Interference O .... Objected

Claim Final	Claim Original	Date	Claim Final	Claim Original	Date	Claim Final	Claim Original	Date
	1			51			101	
	2			52			102	
	3			53			103	
	4			54			104	
	5			55			105	
	6			56			106	
	7			57			107	
	8			58			108	
	9			59			109	
	10			60			110	
	11			61			111	
	12			62			112	
	13			63			113	
	14			64			114	
	15			65			115	
	16			66			116	
	17			67			117	
	18			68			118	
	19			69			119	
	20			70			120	
	21			71			121	
	22			72			122	
	23			73			123	
	24			74			124	
	25			75			125	
	26			76			126	
	27			77			127	
	28			78			128	
	29			79			129	
	30			80			130	
	31			81			131	
	32			82			132	
	33			83			133	
	34			84			134	
	35			85			135	
	36			86			136	
	37			87			137	
	38			88			138	
	39			89			139	
	40			90			140	
	41			91			141	
	42			92			142	
	43			93			143	
	44			94			144	
	45			95			145	
	46			96			146	
	47			97			147	
	48			98			148	
	49			99			149	
	50			100			150	

If more than 150 claims or 9 actions staple additional sheet here

10259494 .093002

Please type a plus sign (+) inside this box → [+]

MODIFIED PTO/SB/05 (03-0
Approved for use through 10/31/2002. OMB 0651-0
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

J1129 U.S. PTO 09/30/02

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 C.F.R. 1.53(b))

Attorney Docket No.	000479.00082
First Inventor	Edmond Colby Munger
Title	IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY
Express Mail Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. ☒ Fee Transmittal Form (e.g., PTO/SB/17)
   (Submit an original and a duplicate for fee processing)
2. ☐ Applicant claims small entity status. See 37 CFR 1.27.
3. ☒ Specification [*Total Pages* 76]
   (preferred arrangement set forth below)
   - Descriptive title of the Invention
   - Cross Reference to Related Applications
   - Statement Regarding Fed sponsored R & D
   - Reference to sequence listing, a table, or a computer program listing appendix
   - Background of the Invention
   - Brief Summary of the Invention
   - Brief Description of the Drawings *(if filed)*
   - Detailed Description
   - Claim(s)
   - Abstract of the Disclosure
4. ☒ Drawing(s) *(35 U.S.C.113)* [*Total Sheets* 35]
   a. ☒ Formal; or
   b. ☐ Informal
5. Oath or Declaration [*Total Pages* 12]
   a. ☐ Newly executed (original or copy); or
   b. ☒ Copy from a prior application (37 CFR 1.63 (d))
   (for a continuation/divisional with Box 18 completed)
   i. ☐ DELETION OF INVENTOR(S)
   Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
6. ☒ Application Data Sheet. See 37 CFR 1.76
7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program *(Appendix)*
8. Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. ☐ Computer Readable Form (CRF)
   b. Specification Sequence Listing on:
   i. ☐ CD-ROM or CD-R (2 copies); or
   ii. ☐ paper
   c. ☐ Statements verifying identity of above copies

J1033 U.S. PTO 10/259494 09/30/02

ACCOMPANYING APPLICATIONS PARTS

9. ☐ Assignment Papers (cover sheet & document(s))
10. ☐ 37 C.F.R.§3.73(b) Statement *(when there is an assignee)* ☐ Power of Attorney
11. ☐ English Translation Document *(if applicable)*
12. ☒ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
13. ☐ Preliminary Amendment
14. ☒ Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
15. ☐ Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. ☐ Nonpublication Request under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. ☐ Other: _____

18. If a CONTINUING APPLICATION, *check appropriate box, and supply the requisite information below and in a preliminary amendment, or in an Application Data Sheet under 37 CFR 1.76:*

☐ Continuation ☒ Divisional ☐ Continuation-in-part (CIP) of prior application No: 09 / 504,783

Prior application information: *Examiner* Krisna Lim *Group / Art Unit:* 2153

For CONTINUATION or DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.

17. CORRESPONDENCE ADDRESS

☒ *Customer Number or Bar Code Label* 22907 *(Insert Customer No. or Attach bar code label here)* or ☐ *Correspondence address below*

Name					
Address					
City		*State*		*Zip Code*	
Country		*Telephone*		*Fax*	

Name (Print/Type)	Ross A. Dannenberg	*Registration No. (Attorney/Agent)*	49,024
Signature	Ross A. Dannenberg	*Date*	September 30, 2002

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

Approved [illegible] use through 10/31/2002. OMB 0651-003[illegible]
U.S. Patent and Tradem[illegible]ce: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1[illegible] persons are required to respond to a collection of information [illegible] it displays a valid OMB control number.

FEE TRANSMITTAL for FY 2002

Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 1076

Complete if Known	
Application Number	TBA
Filing Date	September 30, 2002
First Named Inventor	Edmond Colby Munger
Examiner Name	TBA
Group / Art Unit	2153
Attorney Docket No.	000479.00082

METHOD OF PAYMENT (check all that apply)

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number	19-0733
Deposit Account Name	Banner & Witcoff, Ltd.

The Commissioner is authorized to: *(check all that apply)*

☒ Charge fee(s) indicated below ☒ Credit any overpayments

☐ Charge any additional fee(s) during the pendency of this application

☐ Charge fee(s) indicated below, **except for the filing fee** to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
101	740	201	370	Utility filing fee	740
106	330	206	165	Design filing fee	
107	510	207	255	Plant filing fee	
108	740	208	370	Reissue filing fee	
114	160	214	80	Provisional filing fee	

SUBTOTAL (1) ($) 740

2. EXTRA CLAIM FEES

			Extra Claims		Fee from below		Fee Paid
Total Claims	20	-20 ** =	0	X		=	0
Independent Claims	7	-3 ** =	4	X	84	=	336
Multiple Dependent				X		=	0

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
103	18	203	9	Claims in excess of 20
102	84	202	42	Independent claims in excess of 3
104	280	204	140	Multiple dependent claim, if not paid
109	84	209	42	** Reissue independent claims over original patent
110	18	210	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 336

**or number previously paid, if greater; For Reissues, see above

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	400	216	200	Extension for reply within second month	
117	920	217	460	Extension for reply within third month	
118	1,440	218	720	Extension for reply within fourth month	
128	1,960	228	980	Extension for reply within fifth month	
119	320	219	160	Notice of Appeal	
120	320	220	160	Filing a brief in support of an appeal	
121	280	221	140	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive – unavoidable	
141	1,280	241	640	Petition to revive – unintentional	
142	1,280	242	640	Utility issue fee (or reissue)	
143	460	243	230	Design issue fee	
144	620	244	310	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Processing fee under 37 CFR 1.17 (q)	
126	180	126	180	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	740	246	370	Filing a submission after final rejection (37 CFR § 1.129(a))	
149	740	249	370	For each additional invention to be examined (37 CFR § 1.129(b))	
179	740	279	370	Request for Continued Examination (RCE)	
169	900	169	900	Request for expedited examination of a design application	

Other fee (specify) _____

*Reduced by Basic Filing Fee Paid **SUBTOTAL (3)** ($) 0

SUBMITTED BY				Complete *(if applicable)*	
Name (Print/Type)	Ross A. Dannenberg	*Registration No. Attorney/Agent)*	49,024	*Telephone*	(202) 508-9153
Signature				*Date*	September 30, 2002

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

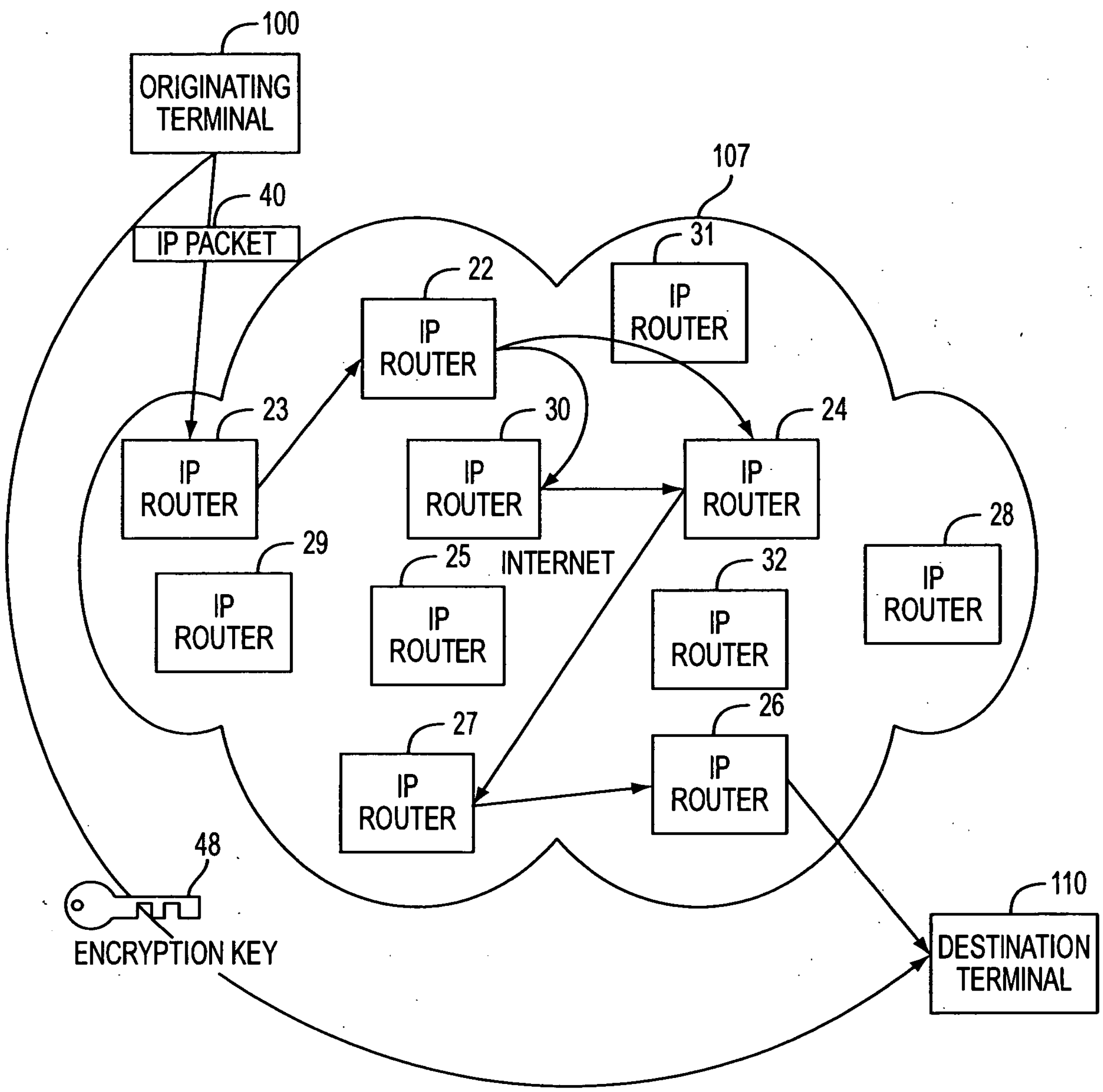
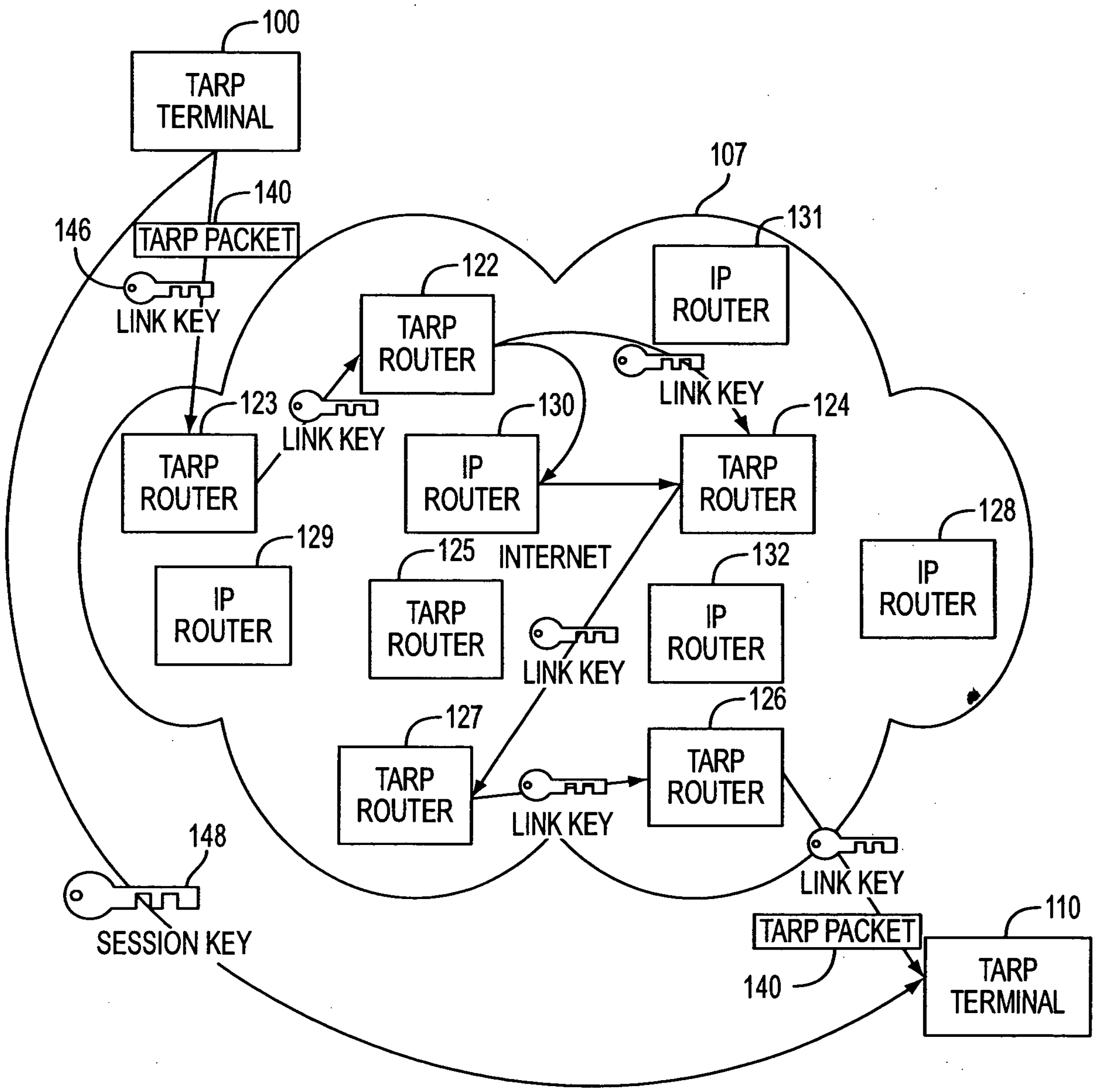
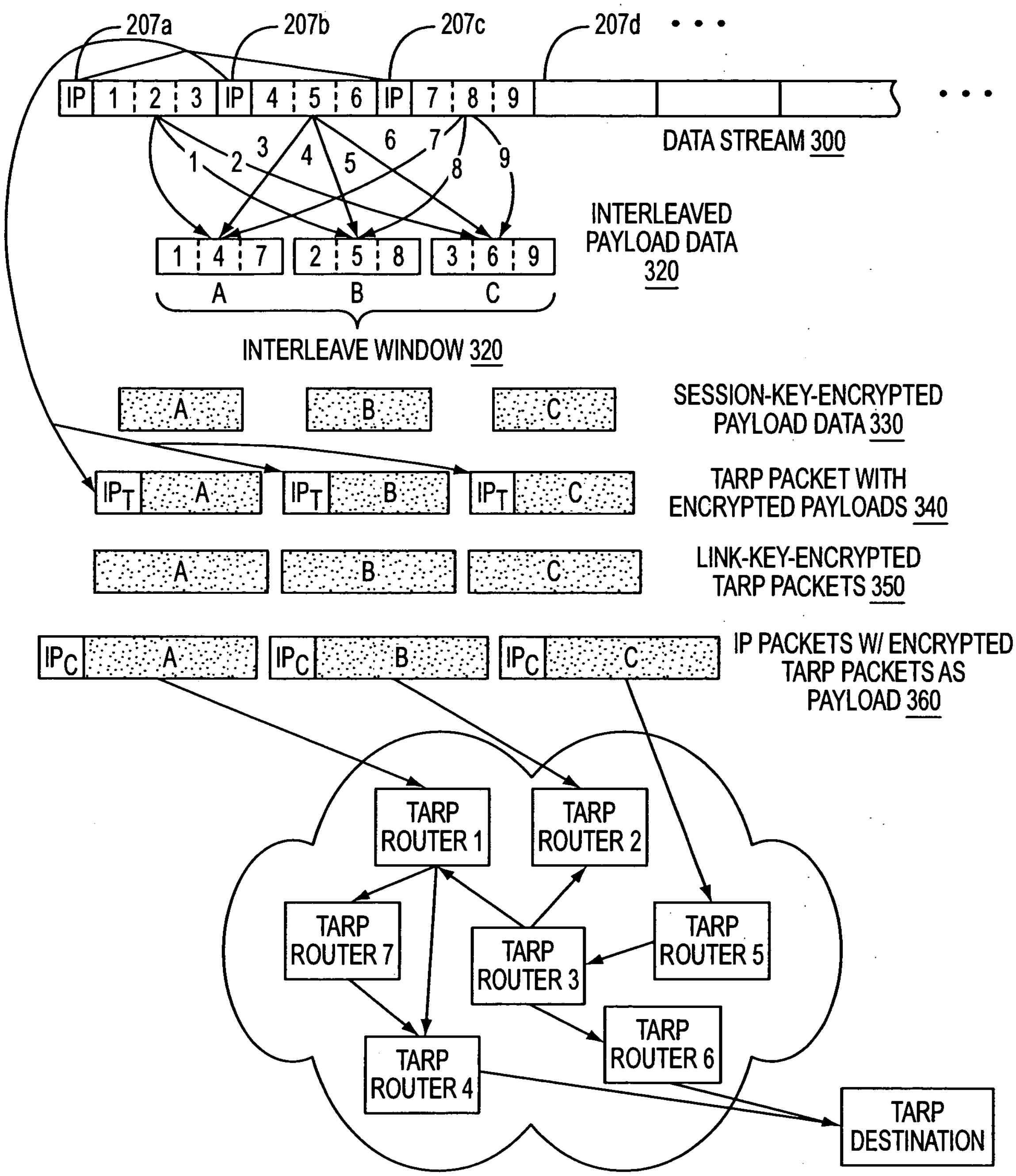

100
ORIGINATING TERMINAL
40
IP PACKET
107
31
IP ROUTER
22
IP ROUTER
23
IP ROUTER
30
IP ROUTER
24
IP ROUTER
28
IP ROUTER
29
IP ROUTER
25
IP ROUTER
INTERNET
32
IP ROUTER
26
IP ROUTER
27
IP ROUTER
48
ENCRYPTION KEY
110
DESTINATION TERMINAL

FIG. 1



100
TARP TERMINAL
107
140
TARP PACKET
146
LINK KEY
131
IP ROUTER
122
TARP ROUTER
LINK KEY
123
LINK KEY
130
124
TARP ROUTER
IP ROUTER
TARP ROUTER
129
128
125
INTERNET
132
IP ROUTER
TARP ROUTER
IP ROUTER
IP ROUTER
LINK KEY
126
127
TARP ROUTER
TARP ROUTER
LINK KEY
148
LINK KEY
SESSION KEY
110
TARP PACKET
140
TARP TERMINAL

FIG. 2



207a
207b
207c
207d
IP 1 2 3 IP 4 5 6 IP 7 8 9
DATA STREAM 300
1 2 3 4 5 6 7 8 9
1 4 7
2 5 8
3 6 9
A
B
C
INTERLEAVED PAYLOAD DATA 320
INTERLEAVE WINDOW 320
SESSION-KEY-ENCRYPTED PAYLOAD DATA 330
IPT A
IPT B
IPT C
TARP PACKET WITH ENCRYPTED PAYLOADS 340
LINK-KEY-ENCRYPTED TARP PACKETS 350
IPC A
IPC B
IPC C
IP PACKETS W/ ENCRYPTED TARP PACKETS AS PAYLOAD 360
TARP ROUTER 1
TARP ROUTER 2
TARP ROUTER 7
TARP ROUTER 3
TARP ROUTER 5
TARP ROUTER 6
TARP ROUTER 4
TARP DESTINATION

FIG. 3A

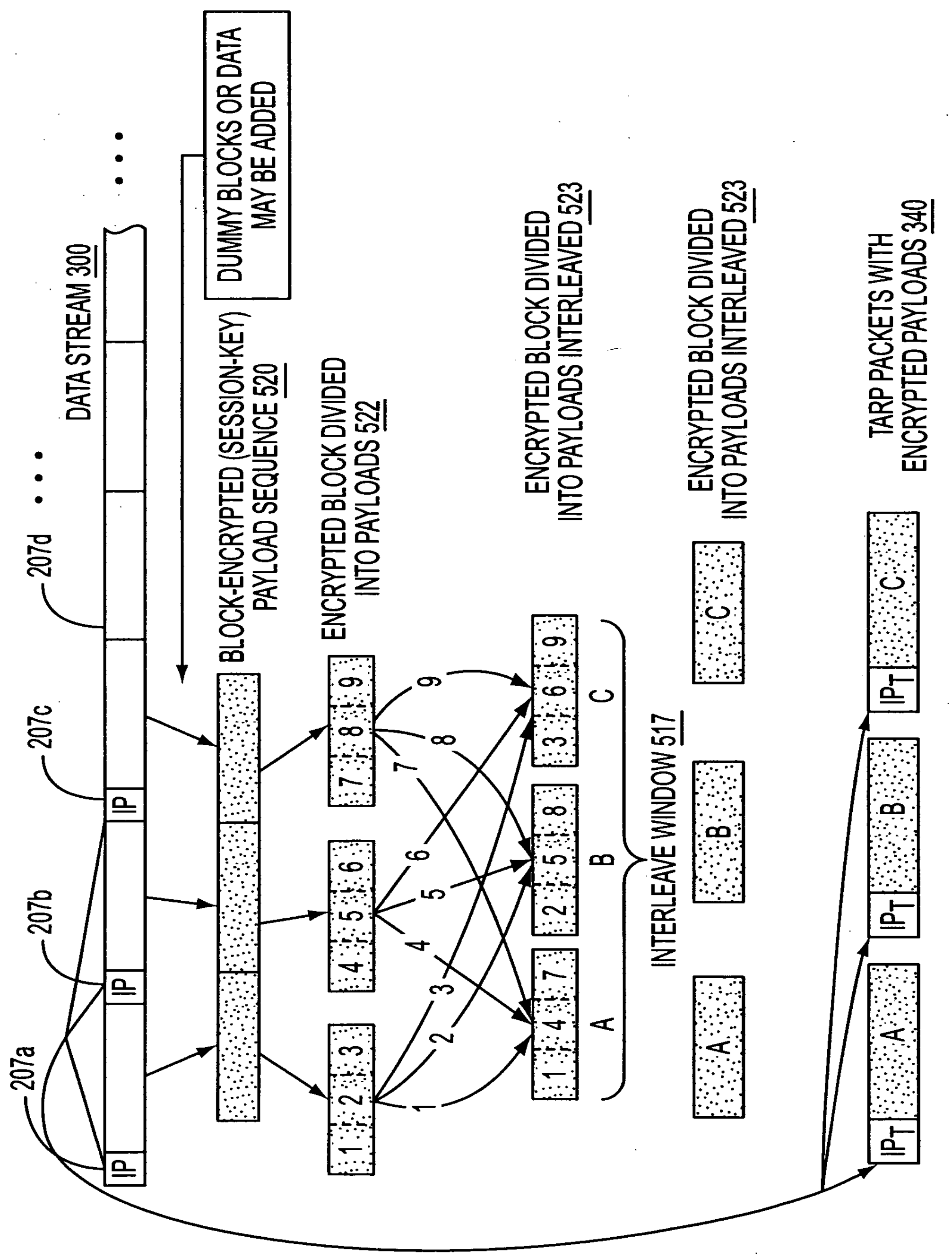

DATA STREAM 300
207a
207b
207c
207d
IP
IP
IP
DUMMY BLOCKS OR DATA MAY BE ADDED
BLOCK-ENCRYPTED (SESSION-KEY) PAYLOAD SEQUENCE 520
ENCRYPTED BLOCK DIVIDED INTO PAYLOADS 522
ENCRYPTED BLOCK DIVIDED INTO PAYLOADS INTERLEAVED 523
ENCRYPTED BLOCK DIVIDED INTO PAYLOADS INTERLEAVED 523
TARP PACKETS WITH ENCRYPTED PAYLOADS 340
INTERLEAVE WINDOW 517
A
B
C
IPT
1 2 3 4 5 6 7 8 9

FIG. 3B

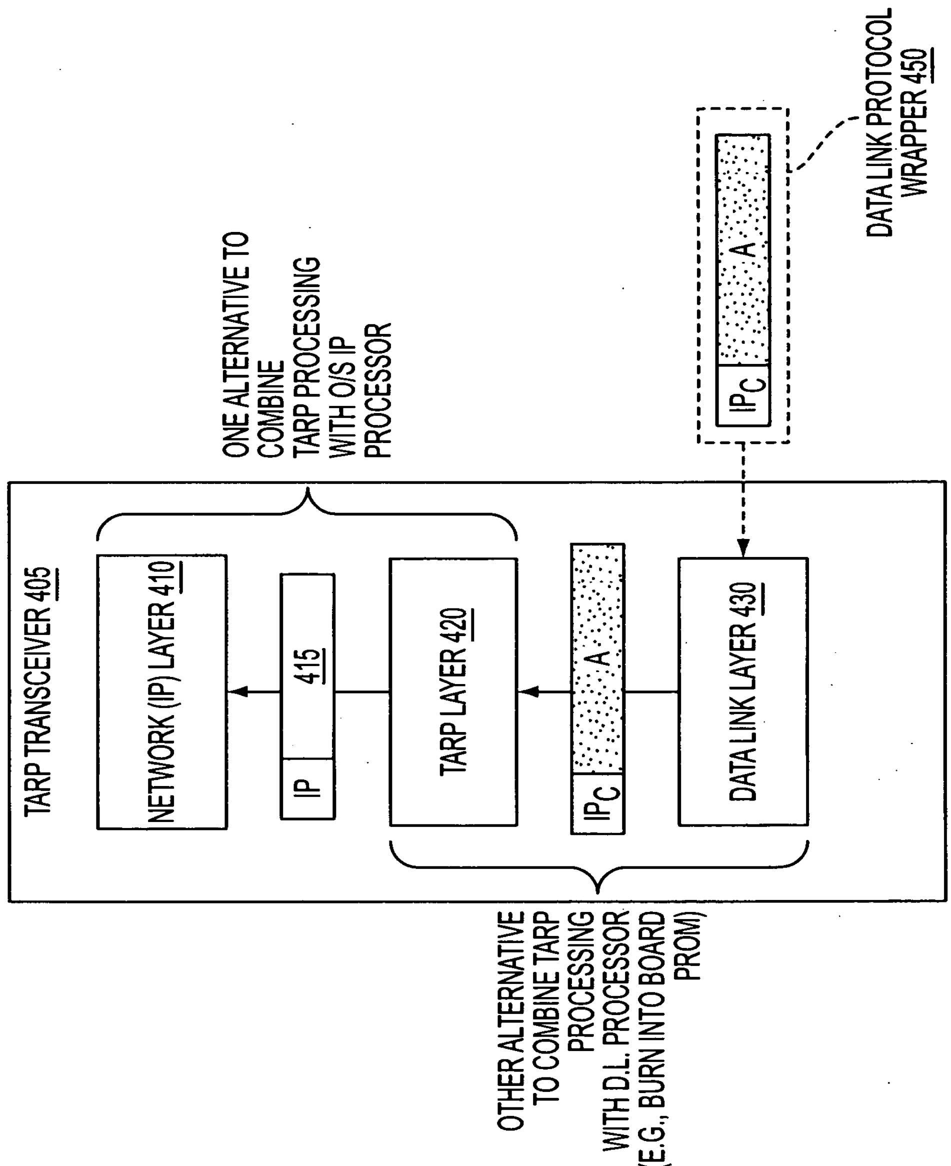

TARP TRANSCEIVER 405
NETWORK (IP) LAYER 410
IP
415
TARP LAYER 420
IPC
A
DATA LINK LAYER 430
ONE ALTERNATIVE TO COMBINE TARP PROCESSING WITH O/S IP PROCESSOR
OTHER ALTERNATIVE TO COMBINE TARP PROCESSING WITH D.L. PROCESSOR (E.G., BURN INTO BOARD PROM)
DATA LINK PROTOCOL WRAPPER 450

FIG. 4

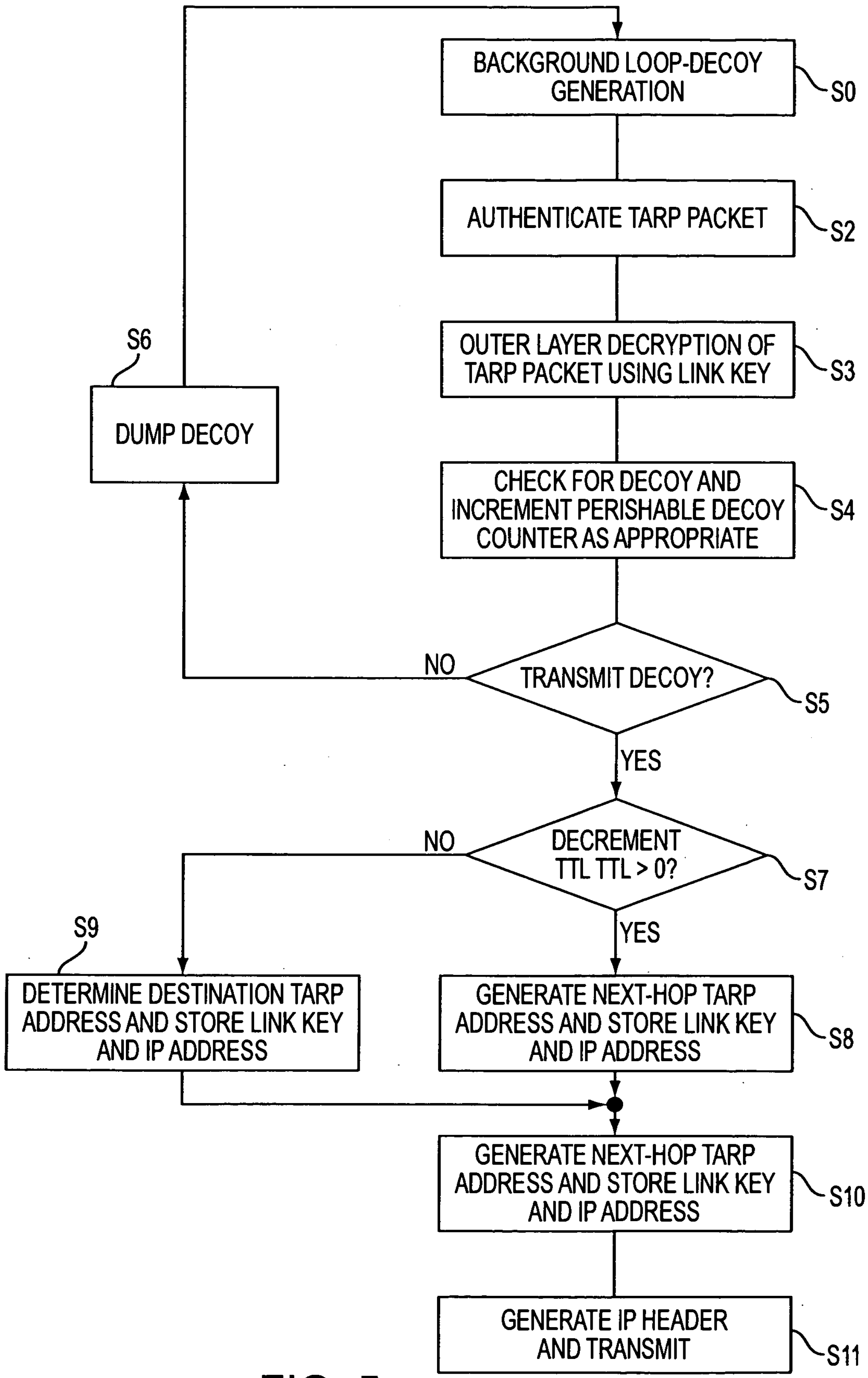

BACKGROUND LOOP-DECOY GENERATION
S0
AUTHENTICATE TARP PACKET
S2
OUTER LAYER DECRYPTION OF TARP PACKET USING LINK KEY
S3
CHECK FOR DECOY AND INCREMENT PERISHABLE DECOY COUNTER AS APPROPRIATE
S4
S6
DUMP DECOY
NO
TRANSMIT DECOY?
S5
YES
NO
DECREMENT TTL TTL > 0?
S7
YES
S9
DETERMINE DESTINATION TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
GENERATE NEXT-HOP TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
S8
GENERATE NEXT-HOP TARP ADDRESS AND STORE LINK KEY AND IP ADDRESS
S10
GENERATE IP HEADER AND TRANSMIT
S11

FIG. 5

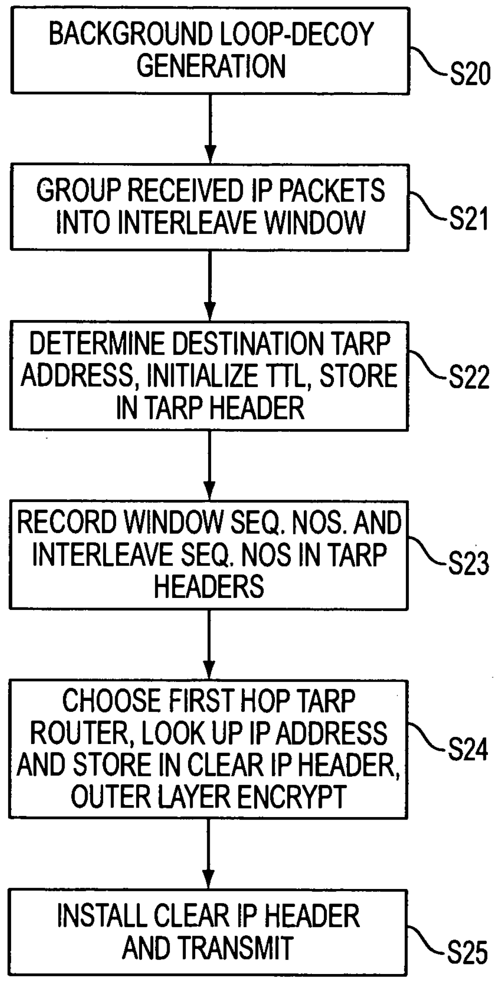

BACKGROUND LOOP-DECOY GENERATION
S20
GROUP RECEIVED IP PACKETS INTO INTERLEAVE WINDOW
S21
DETERMINE DESTINATION TARP ADDRESS, INITIALIZE TTL, STORE IN TARP HEADER
S22
RECORD WINDOW SEQ. NOS. AND INTERLEAVE SEQ. NOS IN TARP HEADERS
S23
CHOOSE FIRST HOP TARP ROUTER, LOOK UP IP ADDRESS AND STORE IN CLEAR IP HEADER, OUTER LAYER ENCRYPT
S24
INSTALL CLEAR IP HEADER AND TRANSMIT
S25

FIG. 6

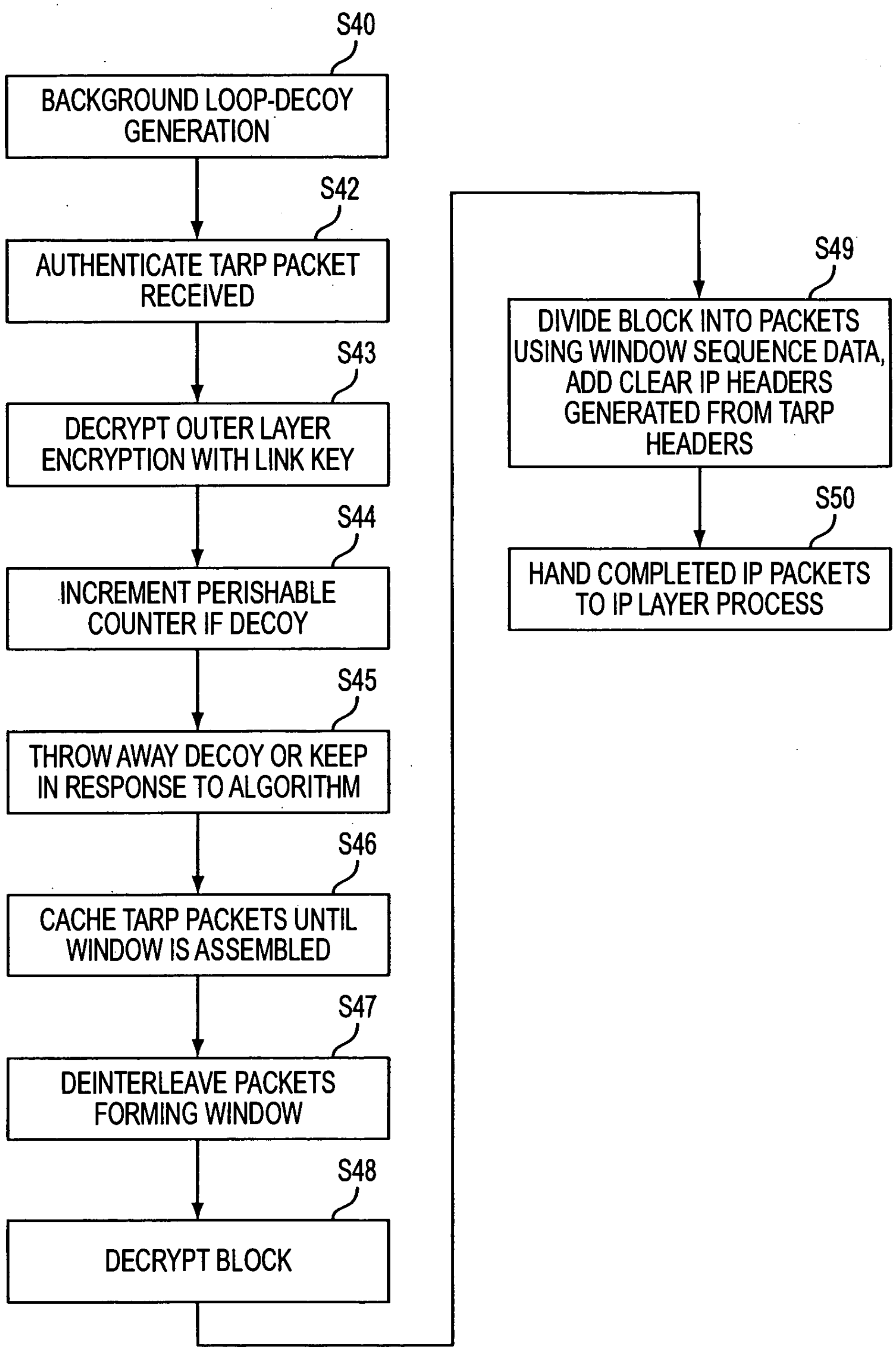

S40
BACKGROUND LOOP-DECOY GENERATION
S42
AUTHENTICATE TARP PACKET RECEIVED
S43
DECRYPT OUTER LAYER ENCRYPTION WITH LINK KEY
S44
INCREMENT PERISHABLE COUNTER IF DECOY
S45
THROW AWAY DECOY OR KEEP IN RESPONSE TO ALGORITHM
S46
CACHE TARP PACKETS UNTIL WINDOW IS ASSEMBLED
S47
DEINTERLEAVE PACKETS FORMING WINDOW
S48
DECRYPT BLOCK
S49
DIVIDE BLOCK INTO PACKETS USING WINDOW SEQUENCE DATA, ADD CLEAR IP HEADERS GENERATED FROM TARP HEADERS
S50
HAND COMPLETED IP PACKETS TO IP LAYER PROCESS

FIG. 7

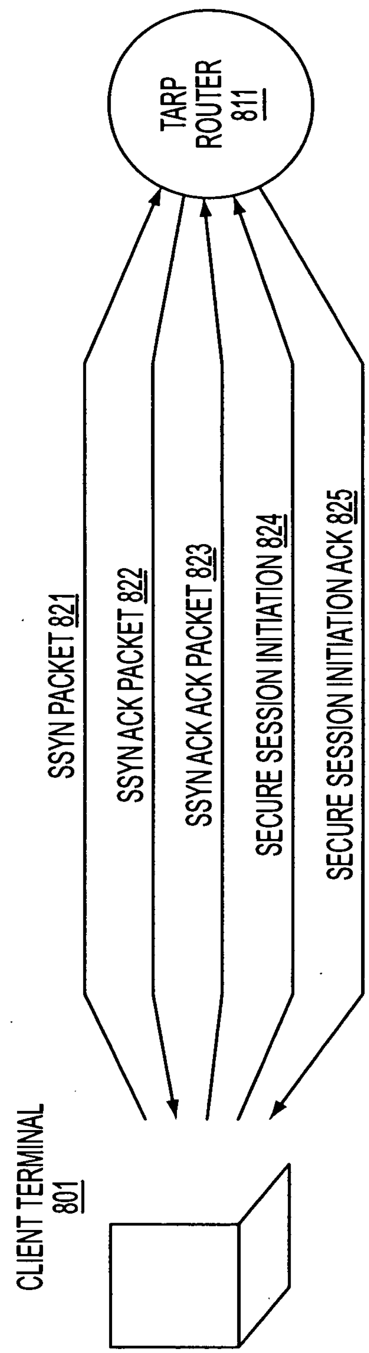
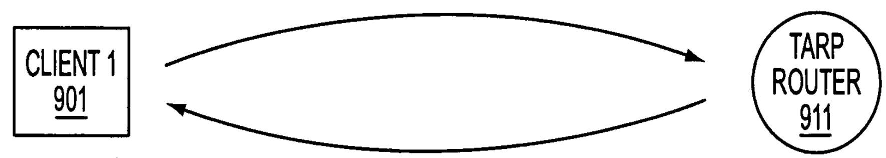

TARP ROUTER 811
SSYN PACKET 821
SSYN ACK PACKET 822
SSYN ACK ACK PACKET 823
SECURE SESSION INITIATION 824
SECURE SESSION INITIATION ACK 825
CLIENT TERMINAL 801

FIG. 8



TRANSMIT TABLE 921

131.218.204.98	,	131.218.204.65
131.218.204.221	,	131.218.204.97
131.218.204.139	,	131.218.204.186
131.218.204.12	,	131.218.204.55
⋮		⋮

RECEIVE TABLE 924

131.218.204.98	,	131.218.204.65
131.218.204.221	,	131.218.204.97
131.218.204.139	,	131.218.204.186
131.218.204.12	,	131.218.204.55
⋮		⋮

RECEIVE TABLE 922

131.218.204.161	,	131.218.204.89
131.218.204.66	,	131.218.204.212
131.218.204.201	,	131.218.204.127
131.218.204.119	,	131.218.204.49
⋮		⋮

TRANSMIT TABLE 923

131.218.204.161	,	131.218.204.89
131.218.204.66	,	131.218.204.212
131.218.204.201	,	131.218.204.127
131.218.204.119	,	131.218.204.49
⋮		⋮

FIG. 9

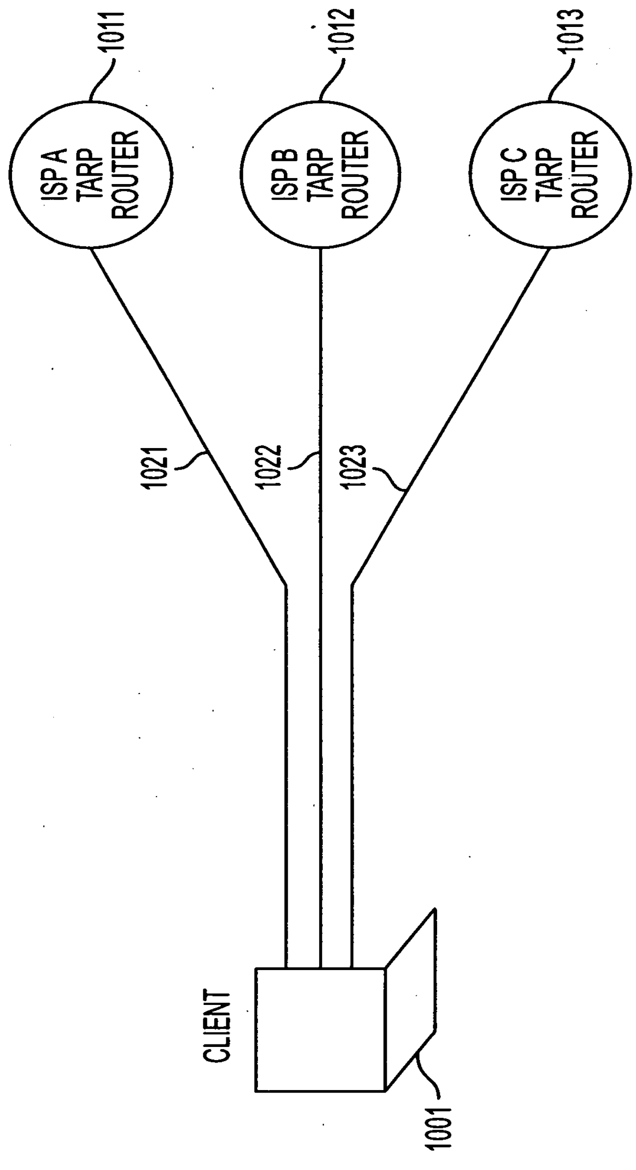

ISP A TARP ROUTER
1011
ISP B TARP ROUTER
1012
ISP C TARP ROUTER
1013
1021
1022
1023
CLIENT
1001

FIG. 10

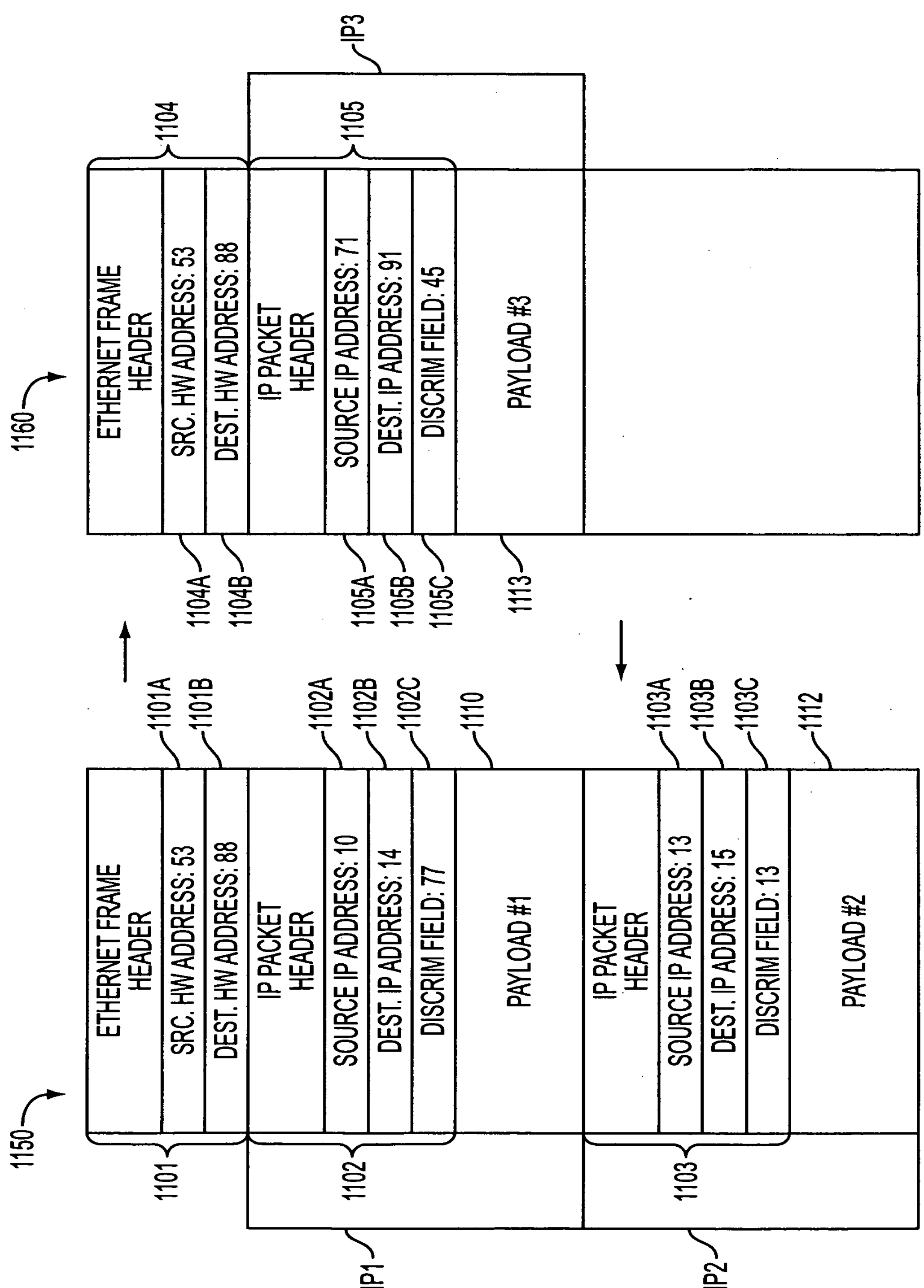

1150
1101
1101A
1101B
ETHERNET FRAME HEADER
SRC. HW ADDRESS: 53
DEST. HW ADDRESS: 88
1102
1102A
1102B
1102C
IP PACKET HEADER
SOURCE IP ADDRESS: 10
DEST. IP ADDRESS: 14
DISCRIM FIELD: 77
1110
PAYLOAD #1
IP1
1103
1103A
1103B
1103C
IP PACKET HEADER
SOURCE IP ADDRESS: 13
DEST. IP ADDRESS: 15
DISCRIM FIELD: 13
1112
PAYLOAD #2
IP2
1160
1104
1104A
1104B
ETHERNET FRAME HEADER
SRC. HW ADDRESS: 53
DEST. HW ADDRESS: 88
1105
1105A
1105B
1105C
IP PACKET HEADER
SOURCE IP ADDRESS: 71
DEST. IP ADDRESS: 91
DISCRIM FIELD: 45
1113
PAYLOAD #3
IP3

FIG. 11

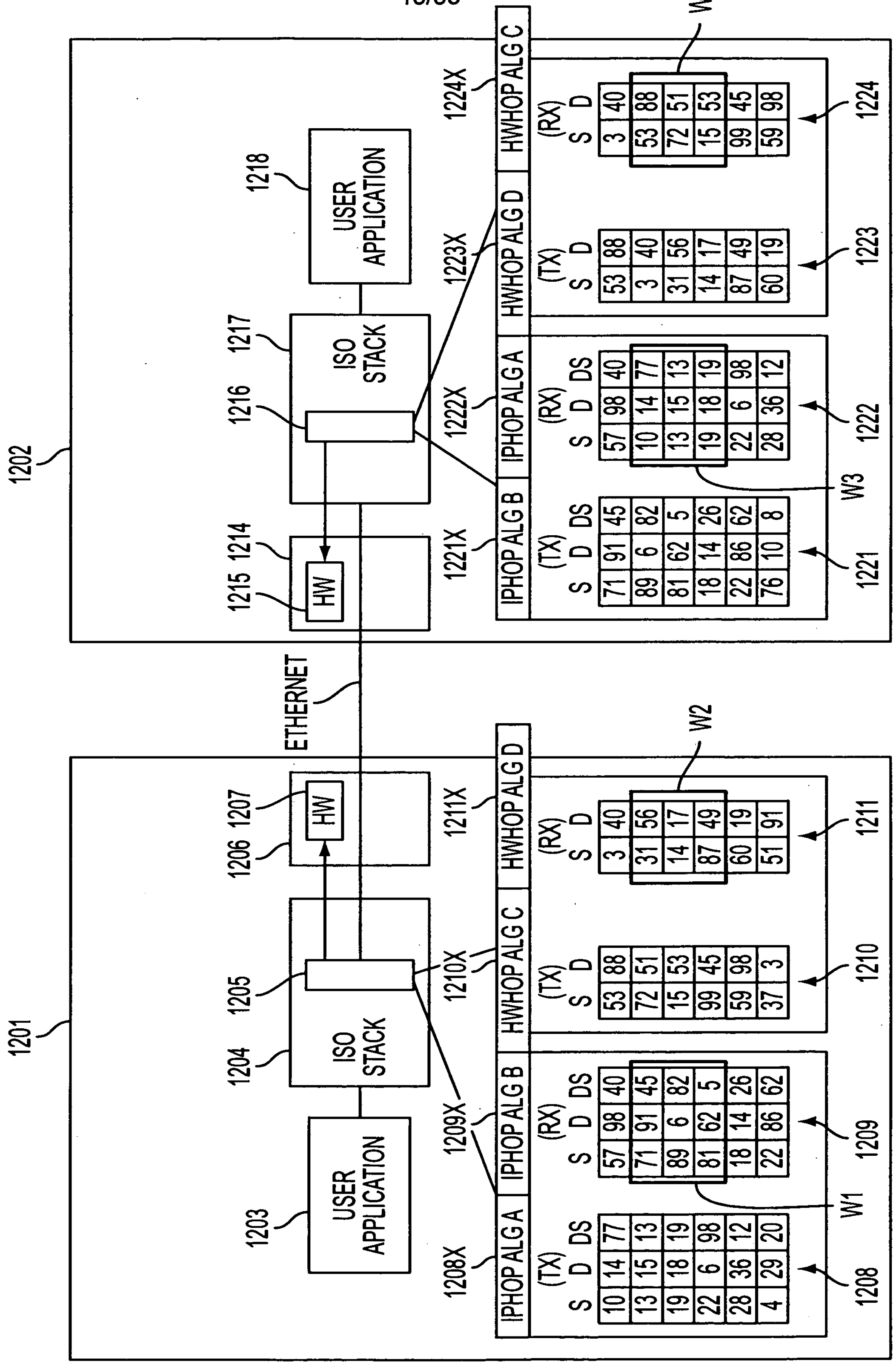

1201
1202
1203
USER APPLICATION
1204
ISO STACK
1205
1206 1207
HW
ETHERNET
1214 1215
HW
1216
1217
ISO STACK
1218
USER APPLICATION
1208X
IPHOP ALG A
1209X
IPHOP ALG B
1210X
HWHOP ALG C
1211X
HWHOP ALG D
1221X
IPHOP ALG B
1222X
IPHOP ALG A
1223X
HWHOP ALG D
1224X
HWHOP ALG C
(TX)
(RX)
S D DS
S D
1208
1209
1210
1211
1221
1222
1223
1224
W1
W2
W3
W4

FIG. 12A

MODE OR EMBODIMENT	HARDWARE ADDRESSES	IP ADDRESSES	DISCRIMINATOR FIELD VALUES
1. PROMISCUOUS	SAME FOR ALL NODES OR COMPLETELY RANDOM	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
2. PROMISCUOUS PER VPN	FIXED FOR EACH VPN	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC
3. HARDWARE HOPPING	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC	CAN BE VARIED IN SYNC

FIG. 12B

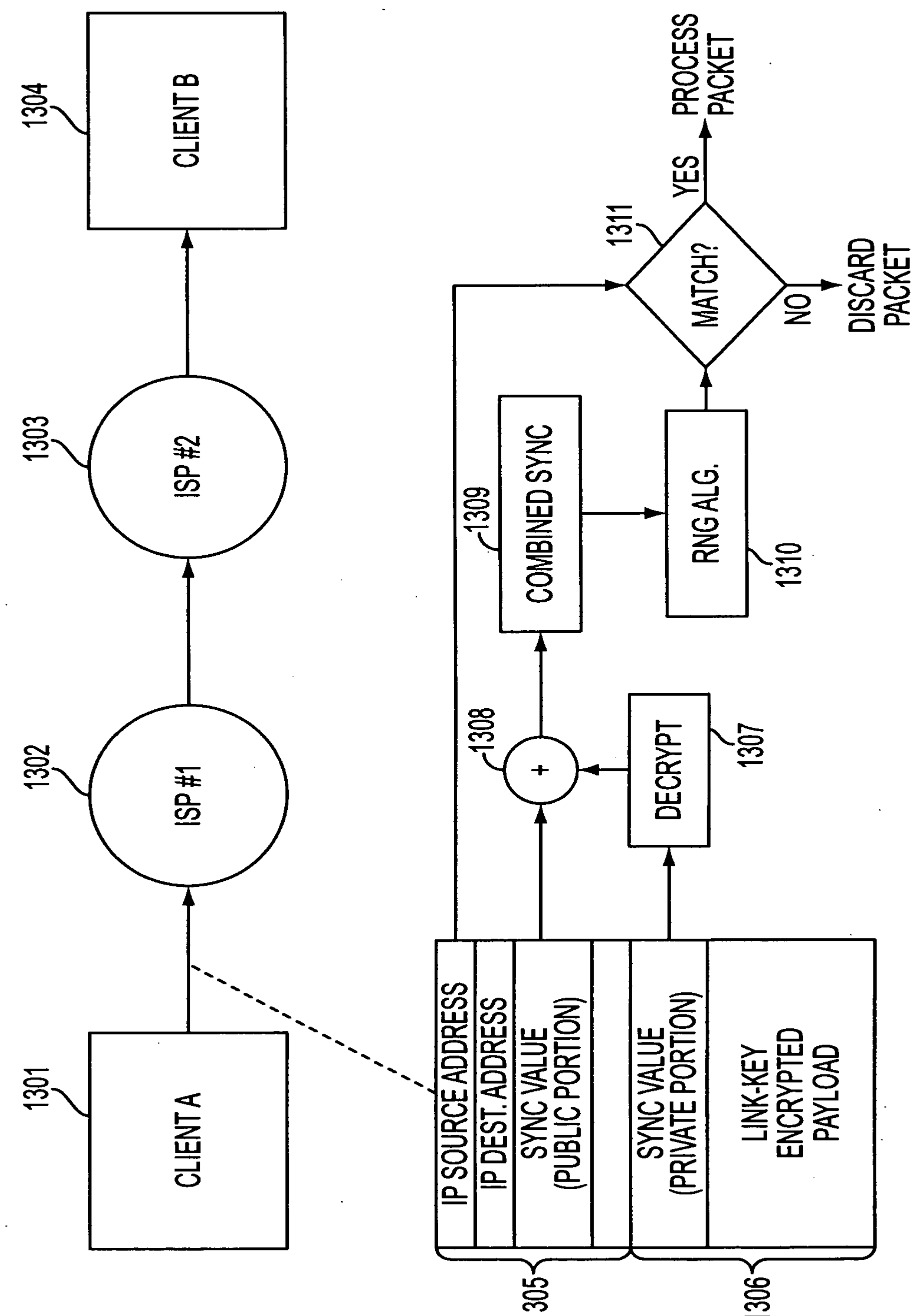

1301
CLIENT A
1302
ISP #1
1303
ISP #2
1304
CLIENT B
IP SOURCE ADDRESS
IP DEST. ADDRESS
SYNC VALUE
(PUBLIC PORTION)
SYNC VALUE
(PRIVATE PORTION)
LINK-KEY
ENCRYPTED
PAYLOAD
1305
1306
DECRYPT
1307
+
1308
COMBINED SYNC
1309
RNG ALG.
1310
MATCH?
1311
YES
PROCESS
PACKET
NO
DISCARD
PACKET

FIG. 13

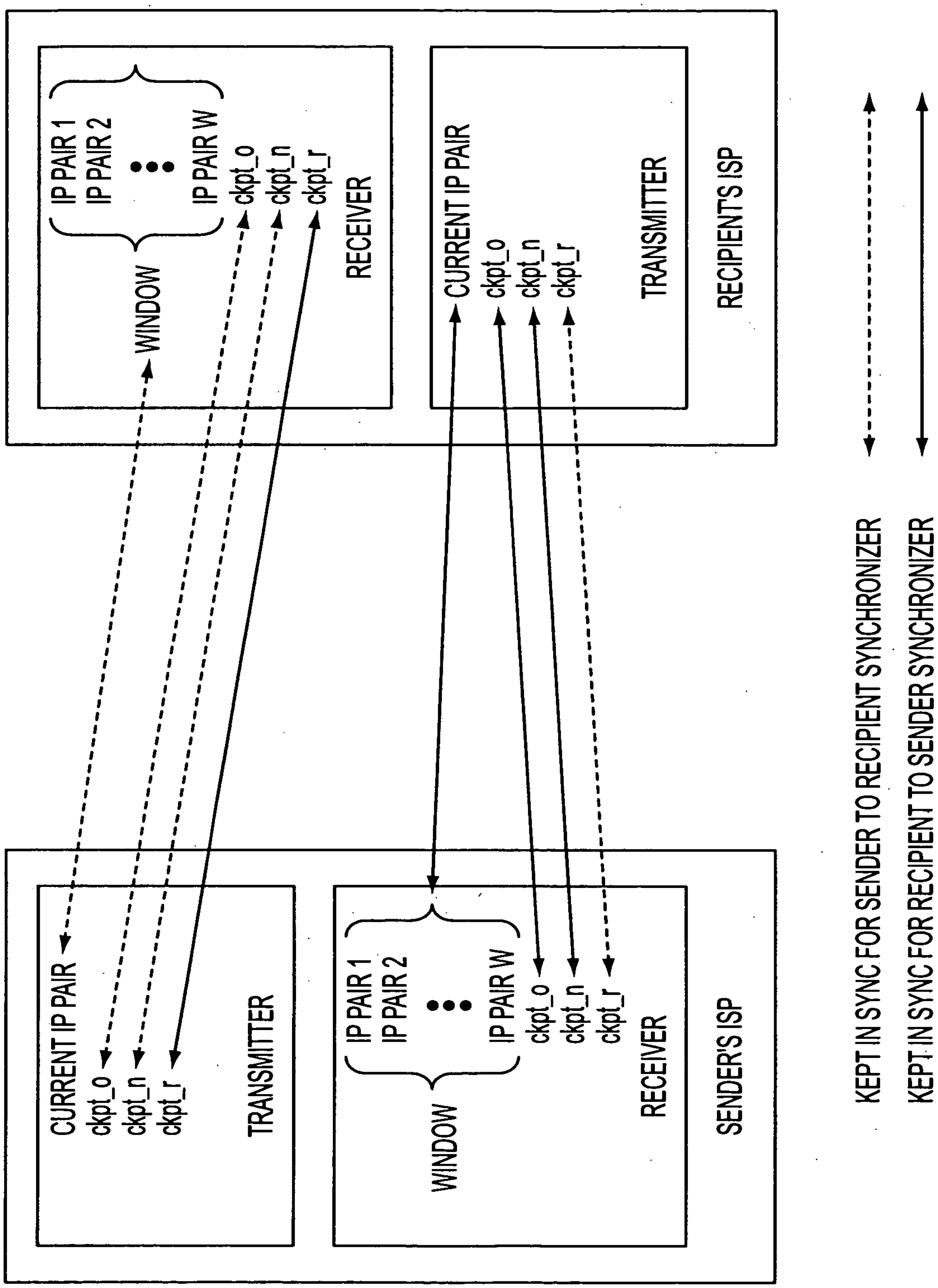

WINDOW
IP PAIR 1
IP PAIR 2
IP PAIR W
ckpt_o
ckpt_n
ckpt_r
RECEIVER
CURRENT IP PAIR
ckpt_o
ckpt_n
ckpt_r
TRANSMITTER
RECIPIENT'S ISP
CURRENT IP PAIR
ckpt_o
ckpt_n
ckpt_r
TRANSMITTER
WINDOW
IP PAIR 1
IP PAIR 2
IP PAIR W
ckpt_o
ckpt_n
ckpt_r
RECEIVER
SENDER'S ISP
KEPT IN SYNC FOR SENDER TO RECIPIENT SYNCHRONIZER
KEPT IN SYNC FOR RECIPIENT TO SENDER SYNCHRONIZER

FIG. 14

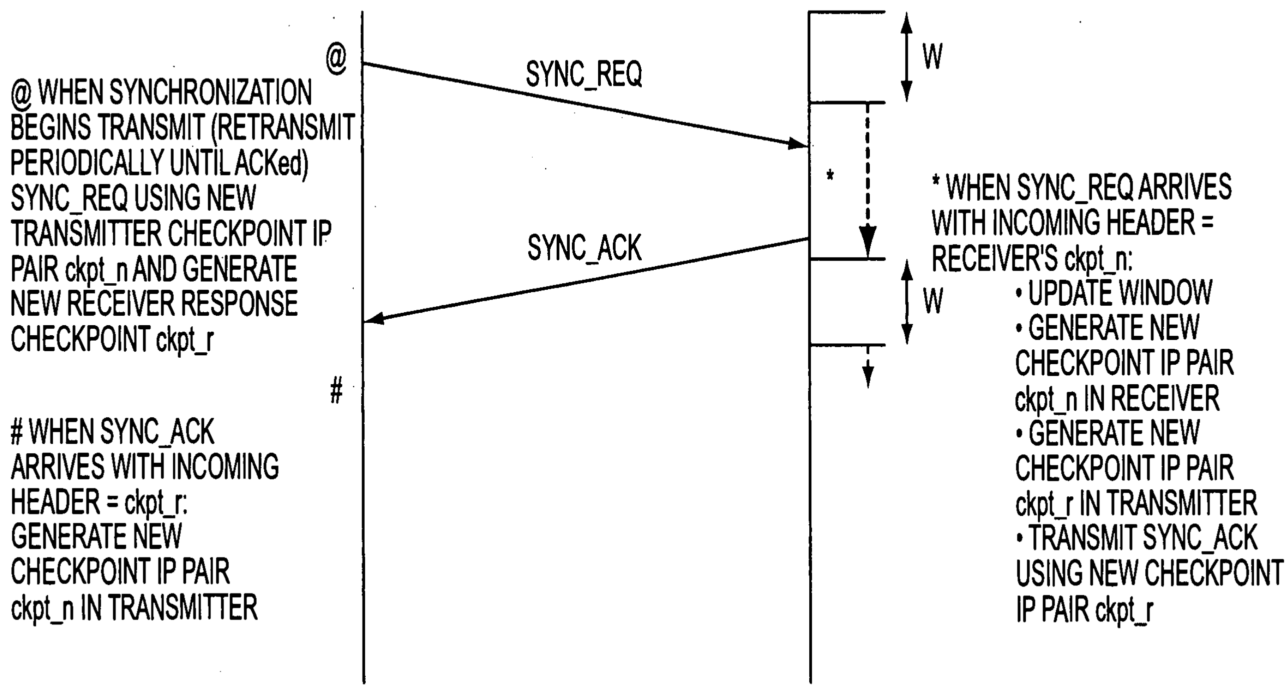

@ WHEN SYNCHRONIZATION BEGINS TRANSMIT (RETRANSMIT PERIODICALLY UNTIL ACKed) SYNC_REQ USING NEW TRANSMITTER CHECKPOINT IP PAIR ckpt_n AND GENERATE NEW RECEIVER RESPONSE CHECKPOINT ckpt_r
# WHEN SYNC_ACK ARRIVES WITH INCOMING HEADER = ckpt_r: GENERATE NEW CHECKPOINT IP PAIR ckpt_n IN TRANSMITTER
@
#
SYNC_REQ
SYNC_ACK
W
W
*
* WHEN SYNC_REQ ARRIVES WITH INCOMING HEADER = RECEIVER'S ckpt_n:
• UPDATE WINDOW
• GENERATE NEW CHECKPOINT IP PAIR ckpt_n IN RECEIVER
• GENERATE NEW CHECKPOINT IP PAIR ckpt_r IN TRANSMITTER
• TRANSMIT SYNC_ACK USING NEW CHECKPOINT IP PAIR ckpt_r

FIG. 15

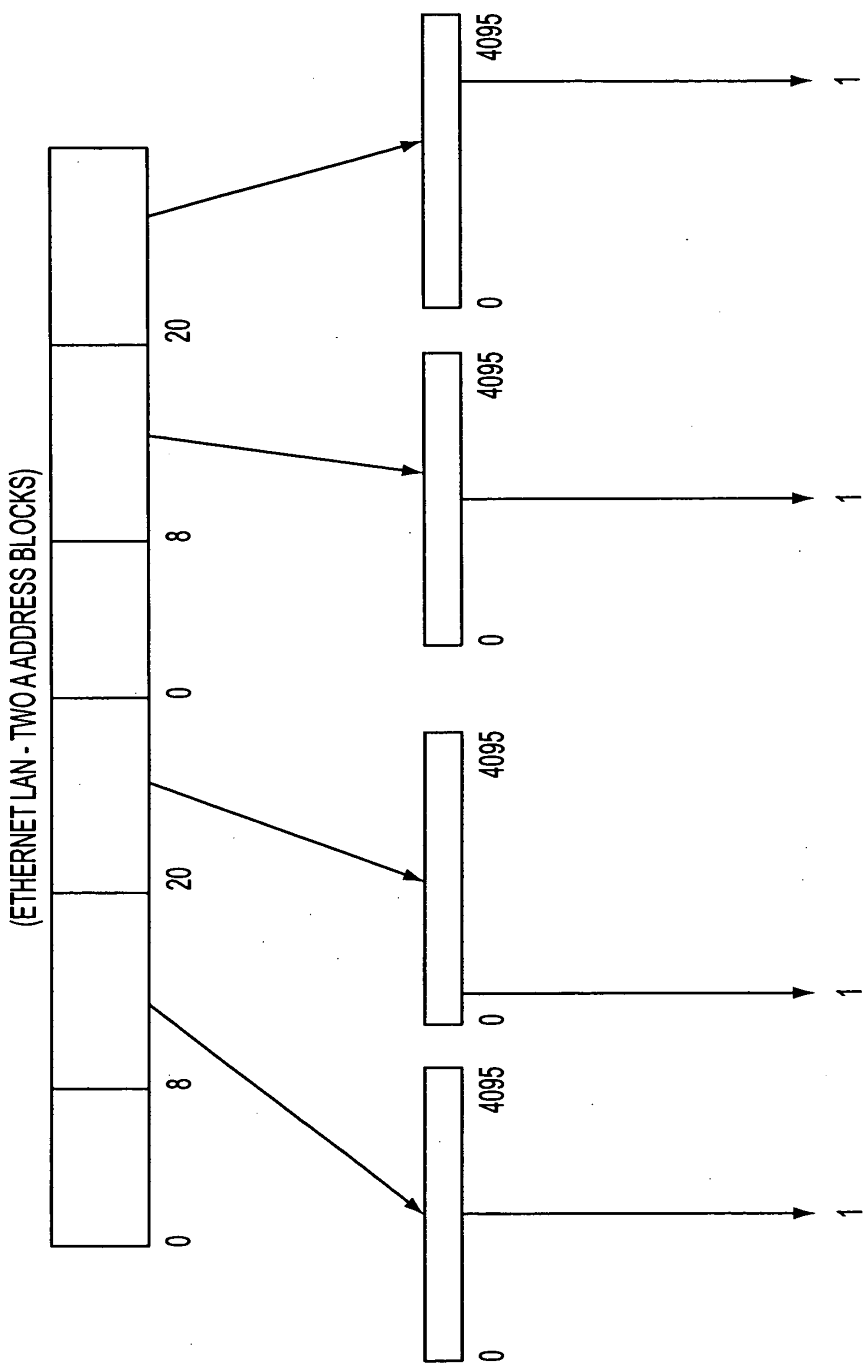

(ETHERNET LAN - TWO A ADDRESS BLOCKS)
0
8
20
0
8
20
0
4095
0
4095
0
4095
0
4095
1
1
1
1

FIG. 16

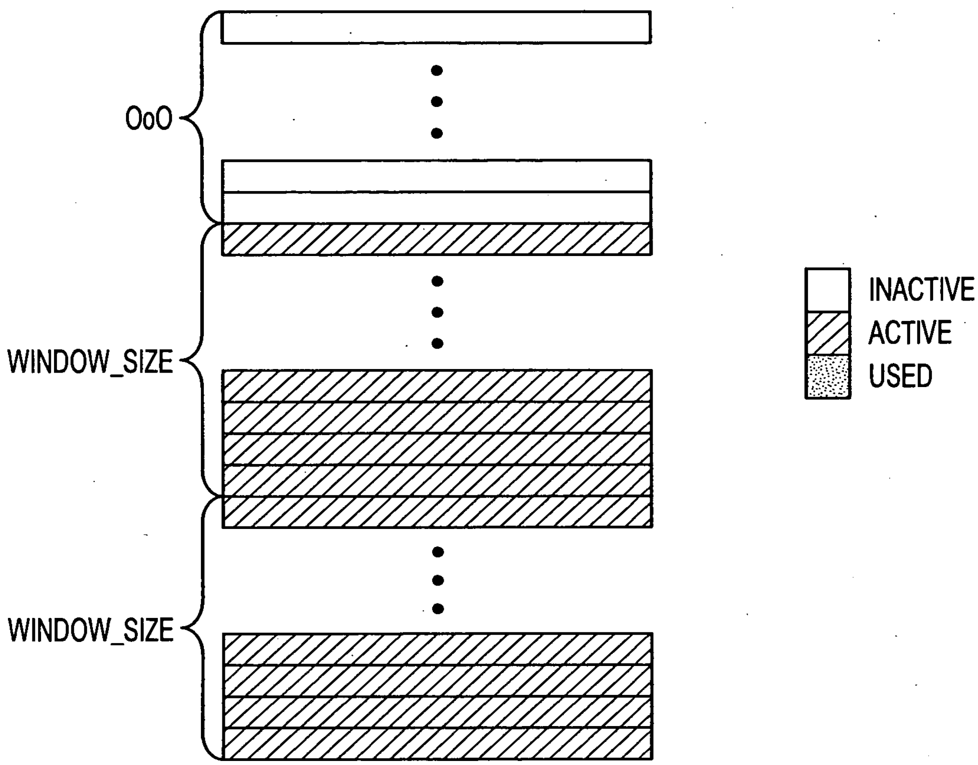


FIG. 17

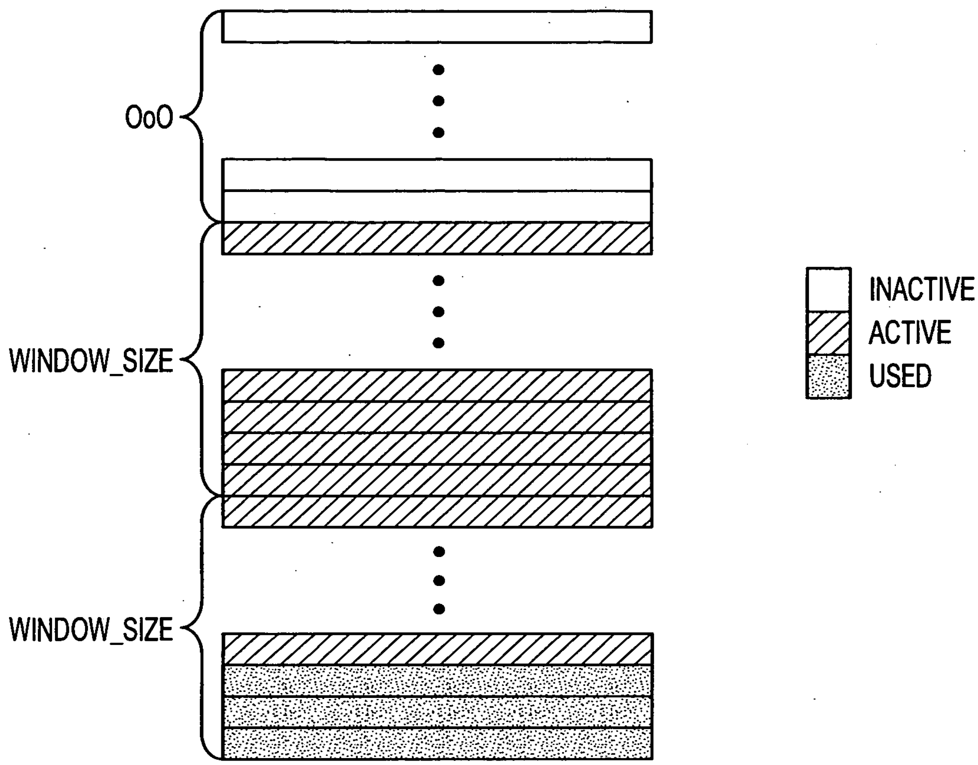


FIG. 18

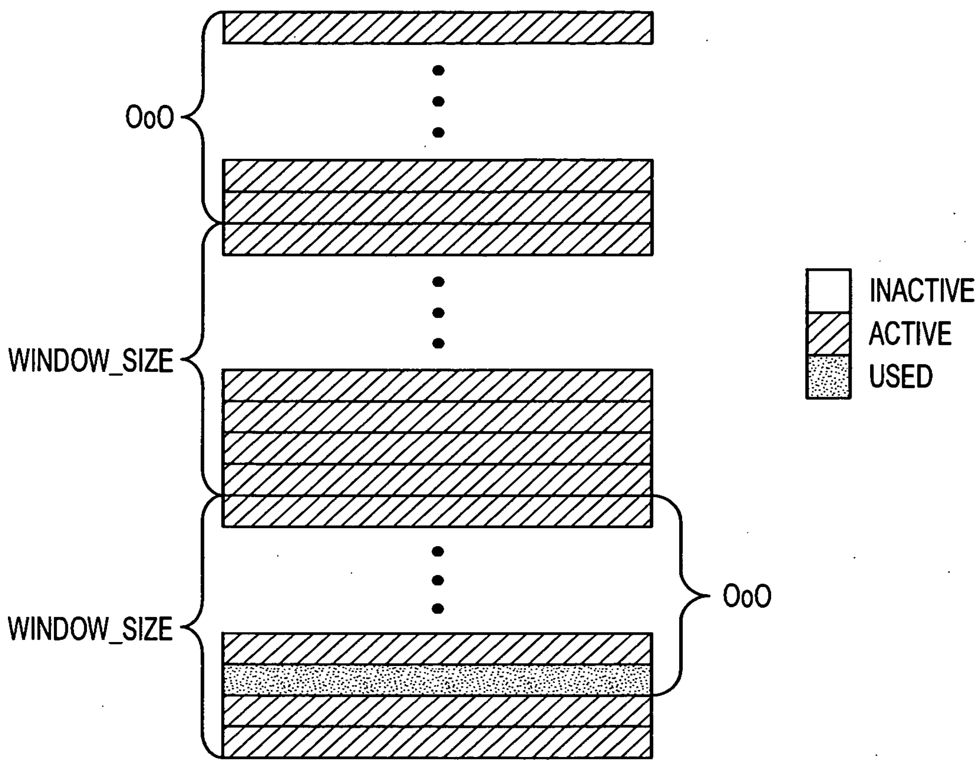
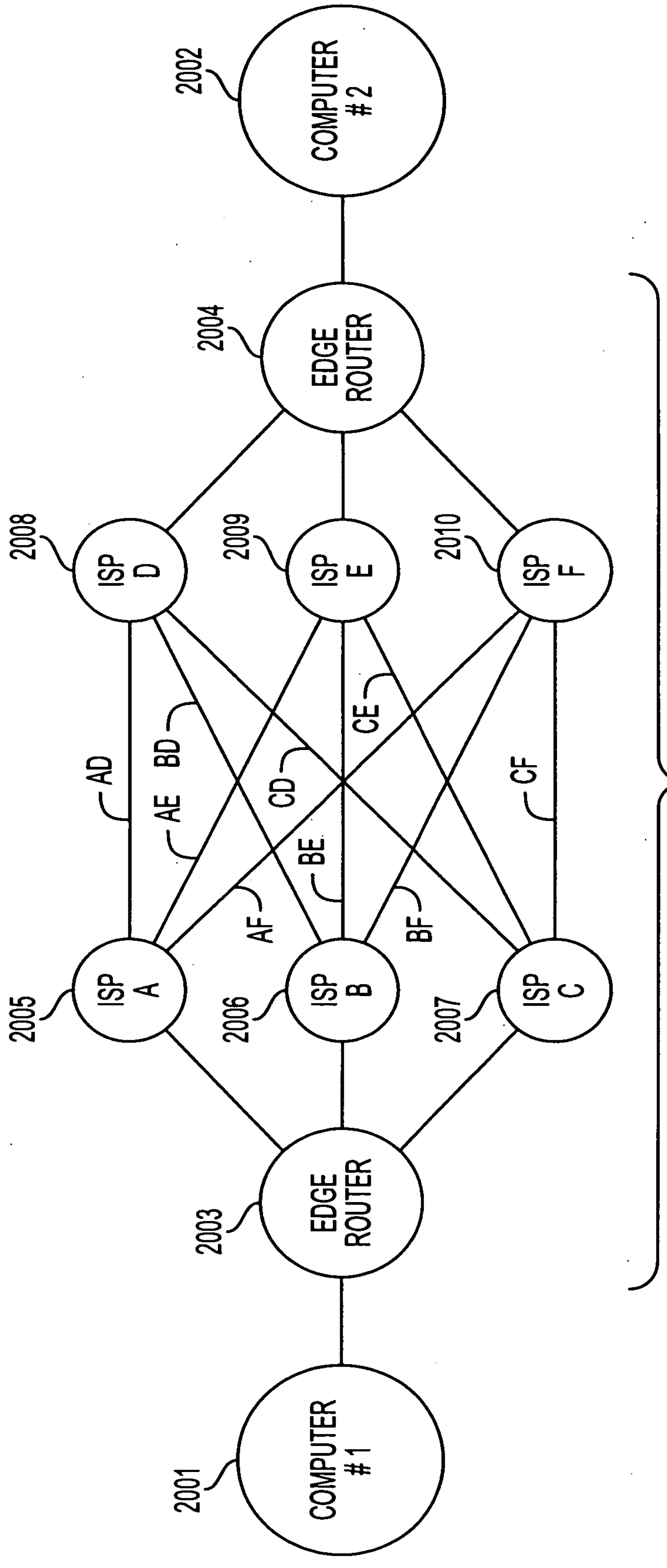


FIG. 19



2002
COMPUTER #2
2004
EDGE ROUTER
2008
ISP D
2009
ISP E
2010
ISP F
AD
AE
BD
CD
BE
CE
CF
AF
BF
2005
ISP A
2006
ISP B
2007
ISP C
2003
EDGE ROUTER
2001
COMPUTER #1
2011

FIG. 20

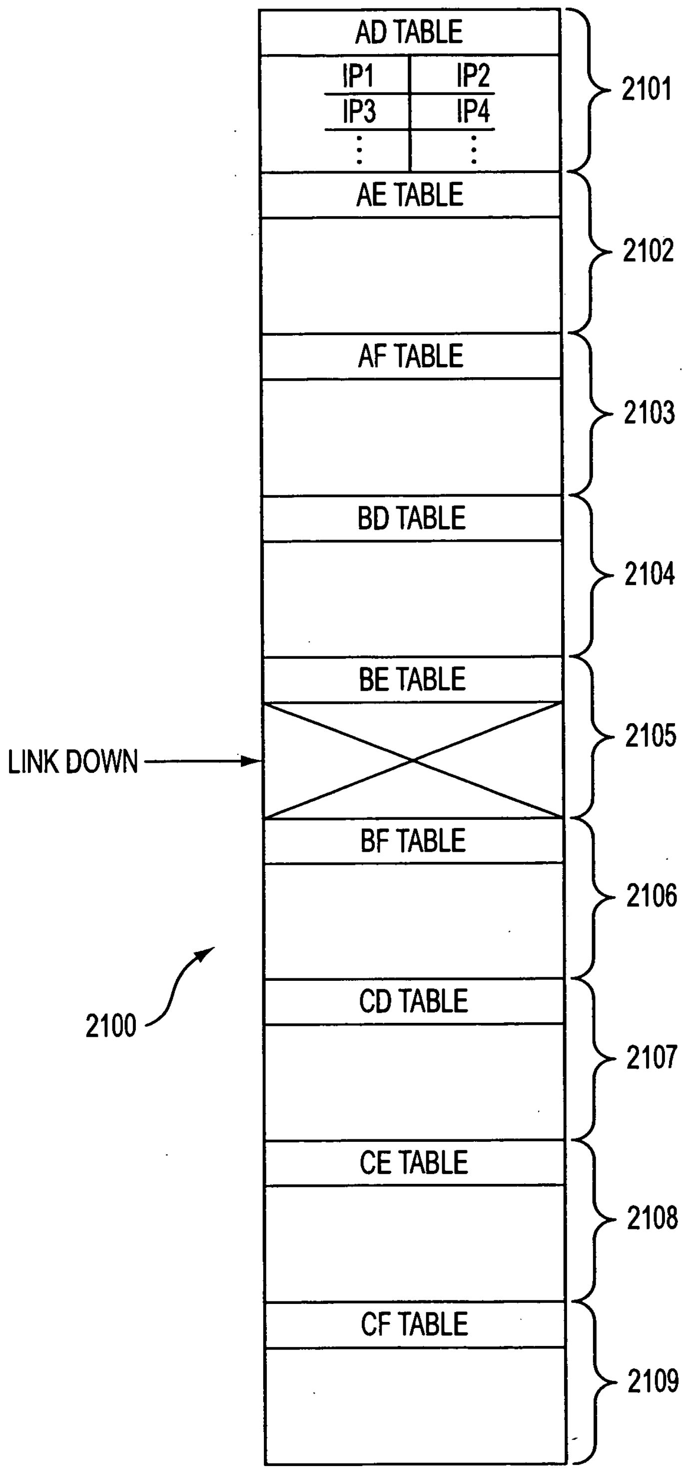

AD TABLE
IP1
IP2
IP3
IP4
AE TABLE
AF TABLE
BD TABLE
BE TABLE
BF TABLE
CD TABLE
CE TABLE
CF TABLE
LINK DOWN
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109

FIG. 21

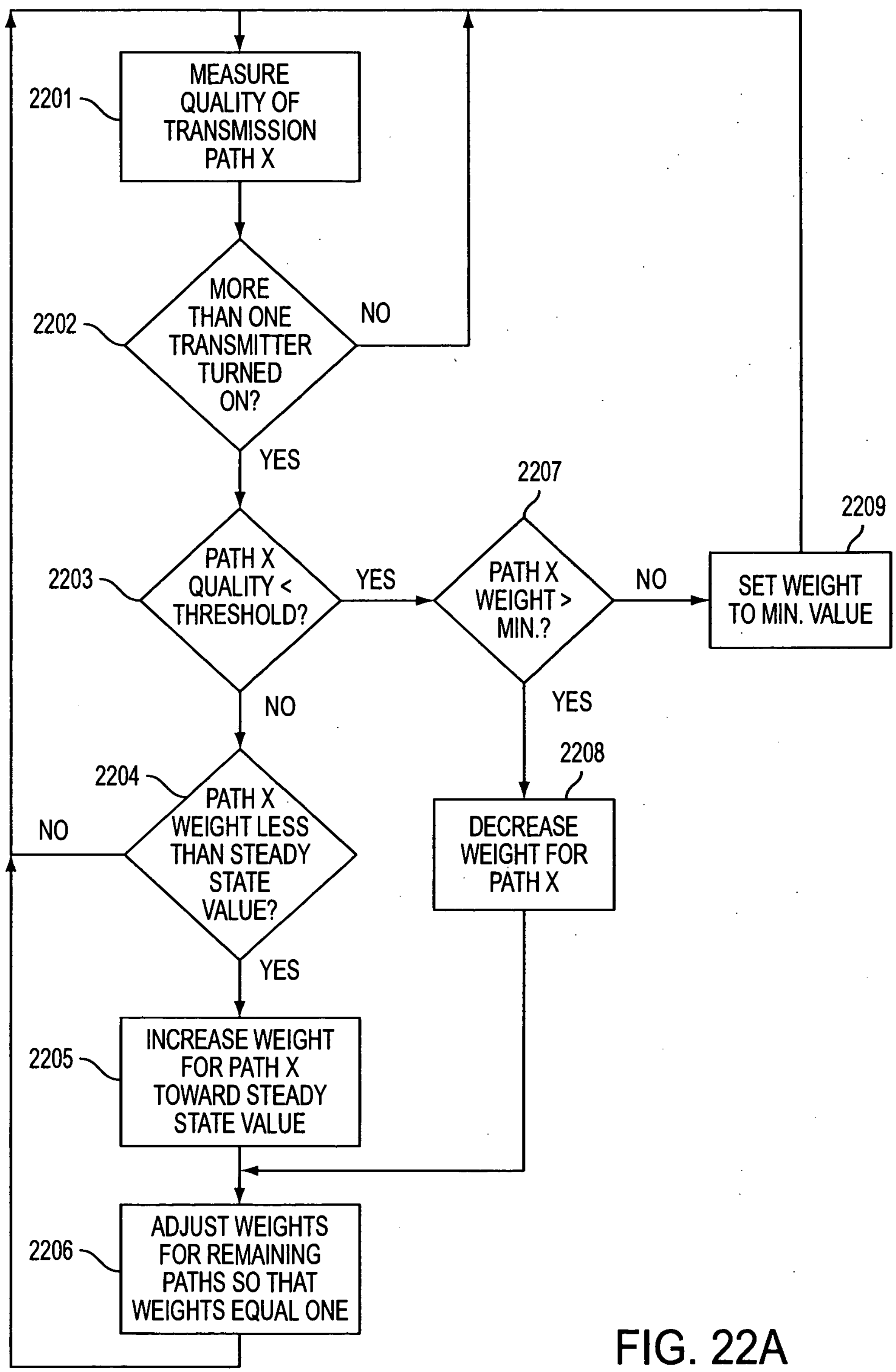

2201
MEASURE QUALITY OF TRANSMISSION PATH X
2202
MORE THAN ONE TRANSMITTER TURNED ON?
NO
YES
2203
PATH X QUALITY < THRESHOLD?
YES
NO
2207
PATH X WEIGHT > MIN.?
NO
YES
2209
SET WEIGHT TO MIN. VALUE
2208
DECREASE WEIGHT FOR PATH X
2204
PATH X WEIGHT LESS THAN STEADY STATE VALUE?
NO
YES
2205
INCREASE WEIGHT FOR PATH X TOWARD STEADY STATE VALUE
2206
ADJUST WEIGHTS FOR REMAINING PATHS SO THAT WEIGHTS EQUAL ONE

FIG. 22A

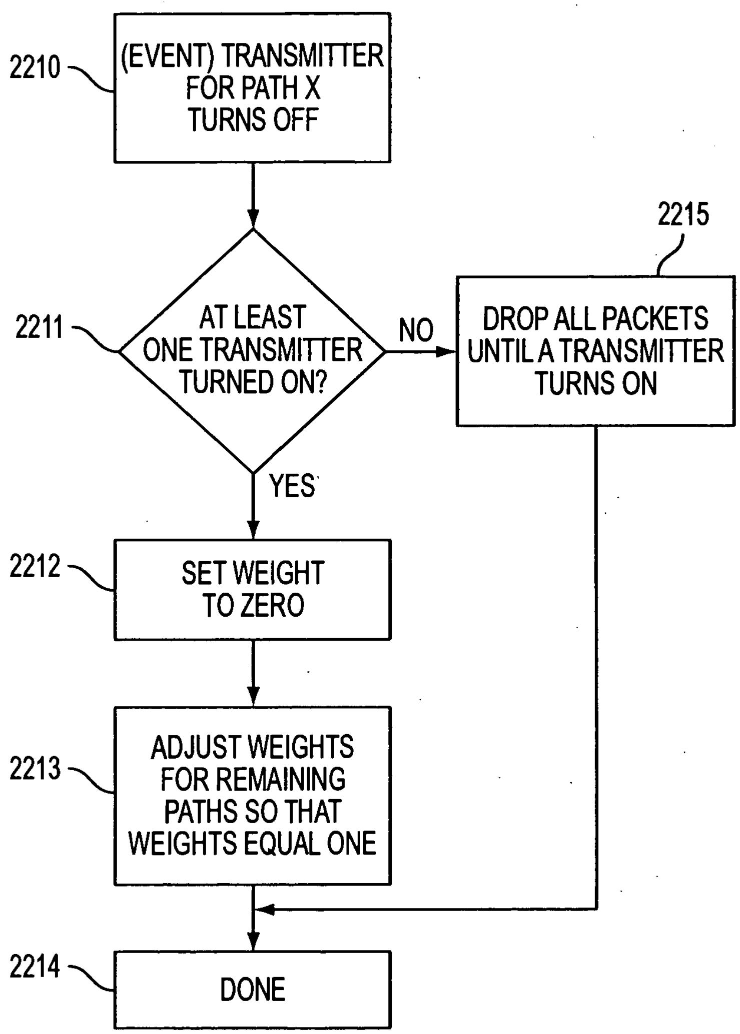

2210
(EVENT) TRANSMITTER FOR PATH X TURNS OFF
2211
AT LEAST ONE TRANSMITTER TURNED ON?
NO
YES
2215
DROP ALL PACKETS UNTIL A TRANSMITTER TURNS ON
2212
SET WEIGHT TO ZERO
2213
ADJUST WEIGHTS FOR REMAINING PATHS SO THAT WEIGHTS EQUAL ONE
2214
DONE

FIG. 22B

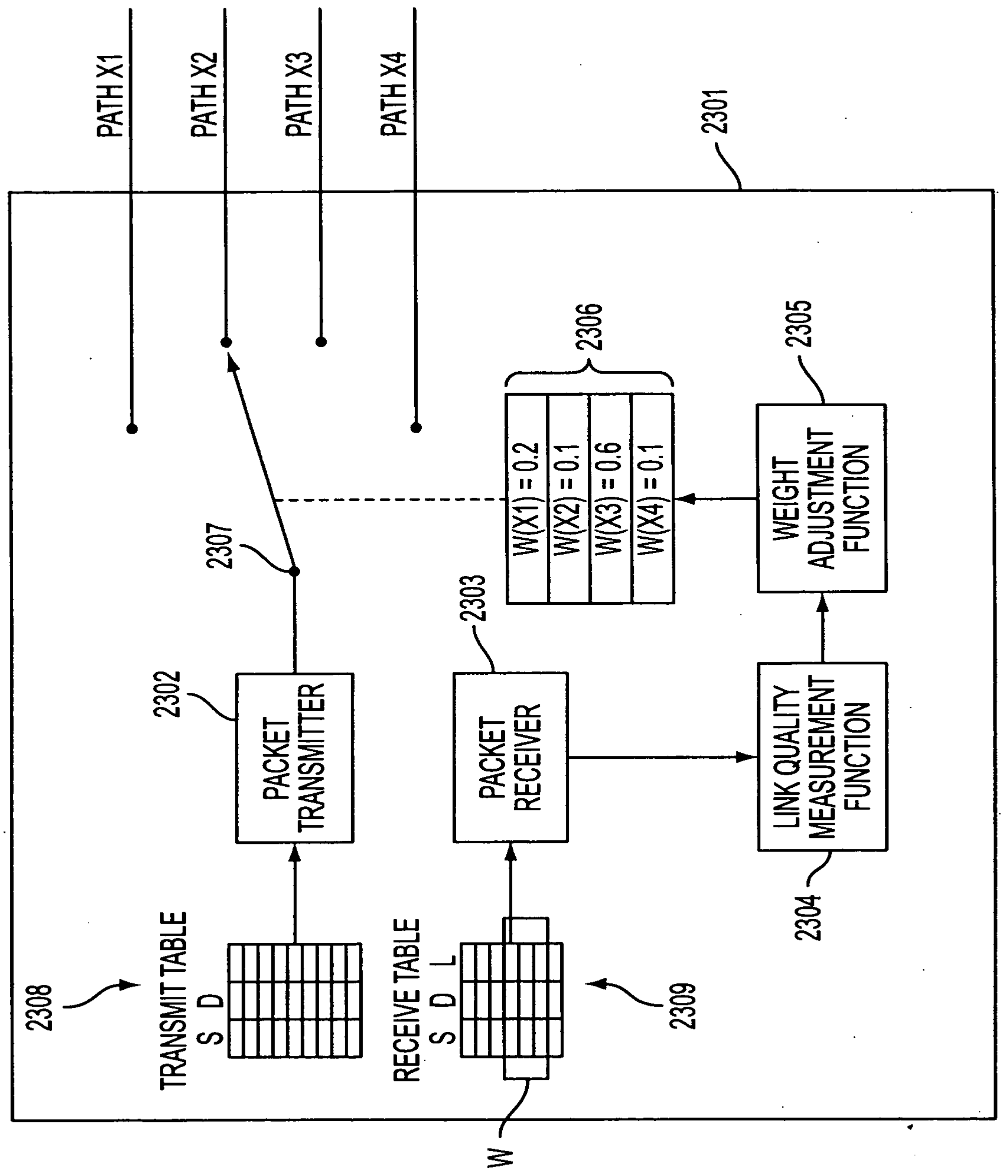
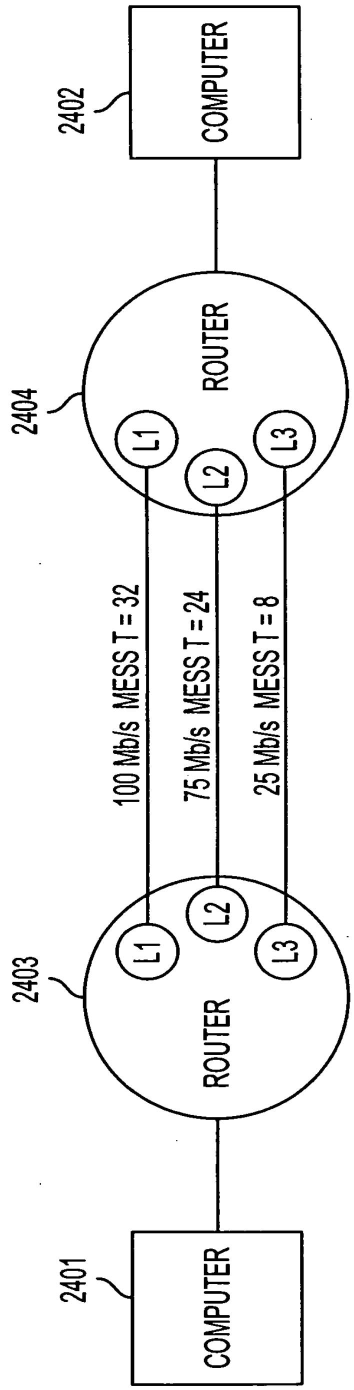

PATH X1
PATH X2
PATH X3
PATH X4
2301
2306
2305
2307
W(X1) = 0.2
W(X2) = 0.1
W(X3) = 0.6
W(X4) = 0.1
WEIGHT ADJUSTMENT FUNCTION
2303
2302
PACKET TRANSMITTER
PACKET RECEIVER
LINK QUALITY MEASUREMENT FUNCTION
2304
2308
TRANSMIT TABLE
S D
RECEIVE TABLE
S D L
2309
W

FIG. 23



2401
COMPUTER
2403
ROUTER
L1
L2
L3
100 Mb/s MESS T = 32
75 Mb/s MESS T = 24
25 Mb/s MESS T = 8
2404
ROUTER
L1
L2
L3
2402
COMPUTER

FIG. 24

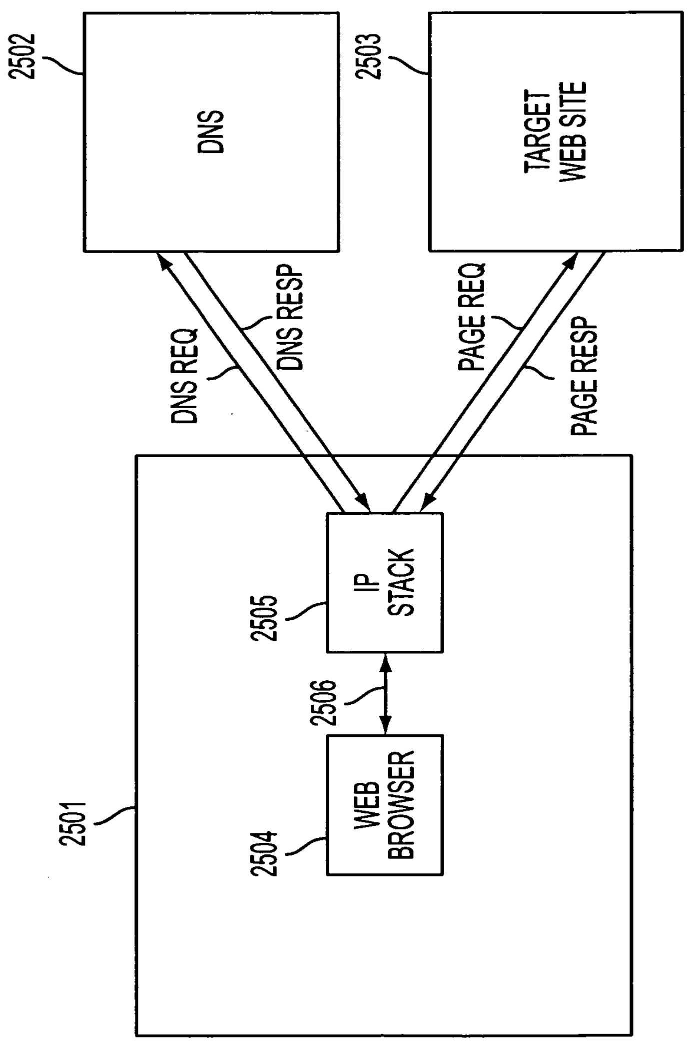


FIG. 25
(PRIOR ART)

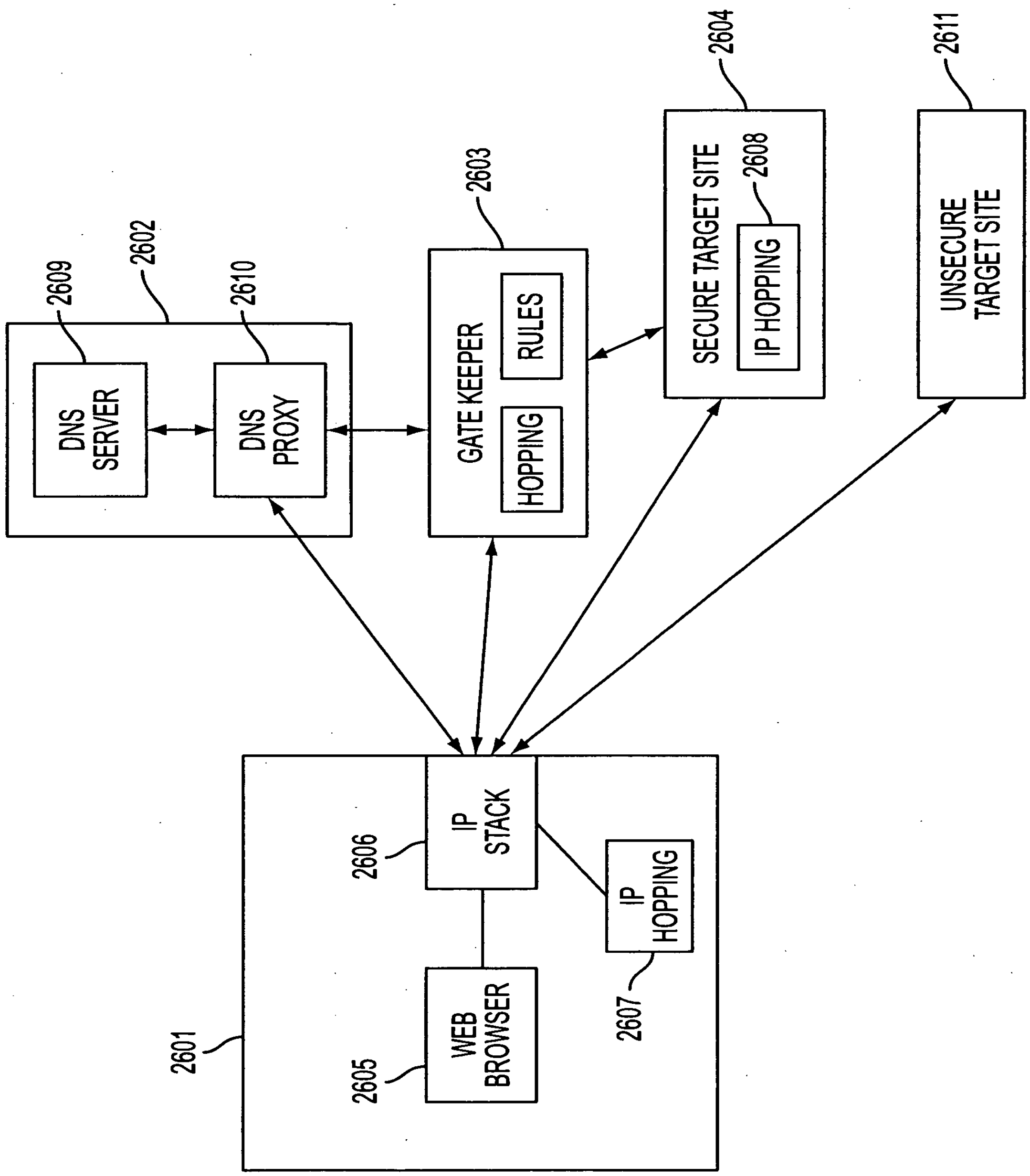

2602
2609
DNS SERVER
2610
DNS PROXY
2603
GATE KEEPER
HOPPING
RULES
2604
SECURE TARGET SITE
2608
IP HOPPING
2611
UNSECURE TARGET SITE
2601
2605
WEB BROWSER
2606
IP STACK
2607
IP HOPPING

FIG. 26

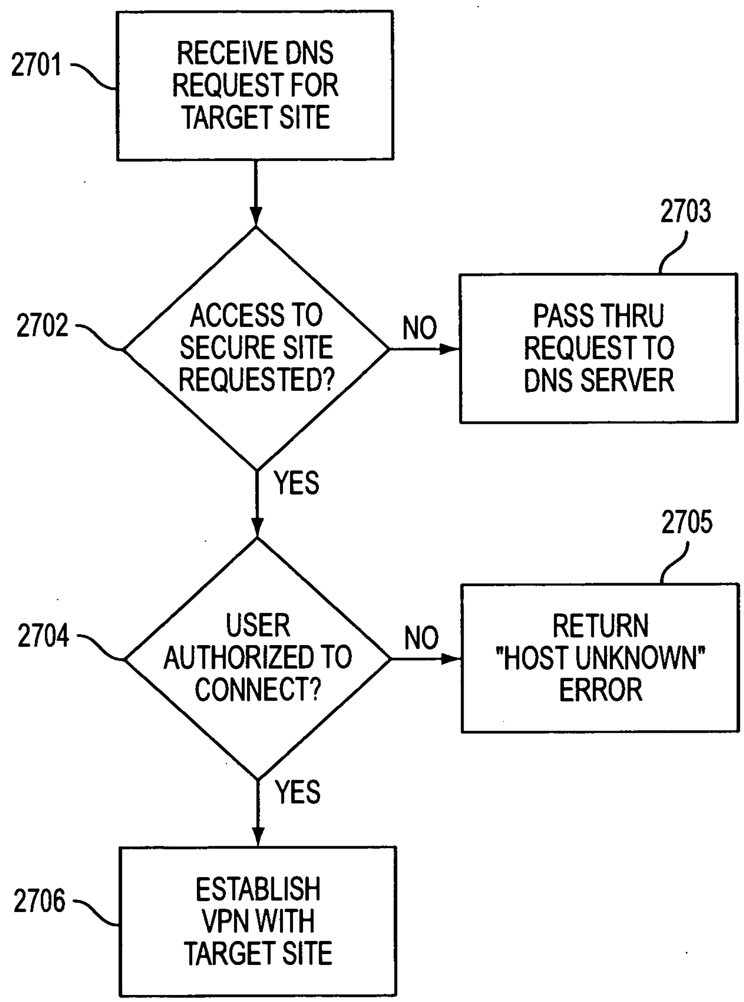

2701
RECEIVE DNS REQUEST FOR TARGET SITE
2702
ACCESS TO SECURE SITE REQUESTED?
NO
2703
PASS THRU REQUEST TO DNS SERVER
YES
2704
USER AUTHORIZED TO CONNECT?
NO
2705
RETURN "HOST UNKNOWN" ERROR
YES
2706
ESTABLISH VPN WITH TARGET SITE

FIG. 27

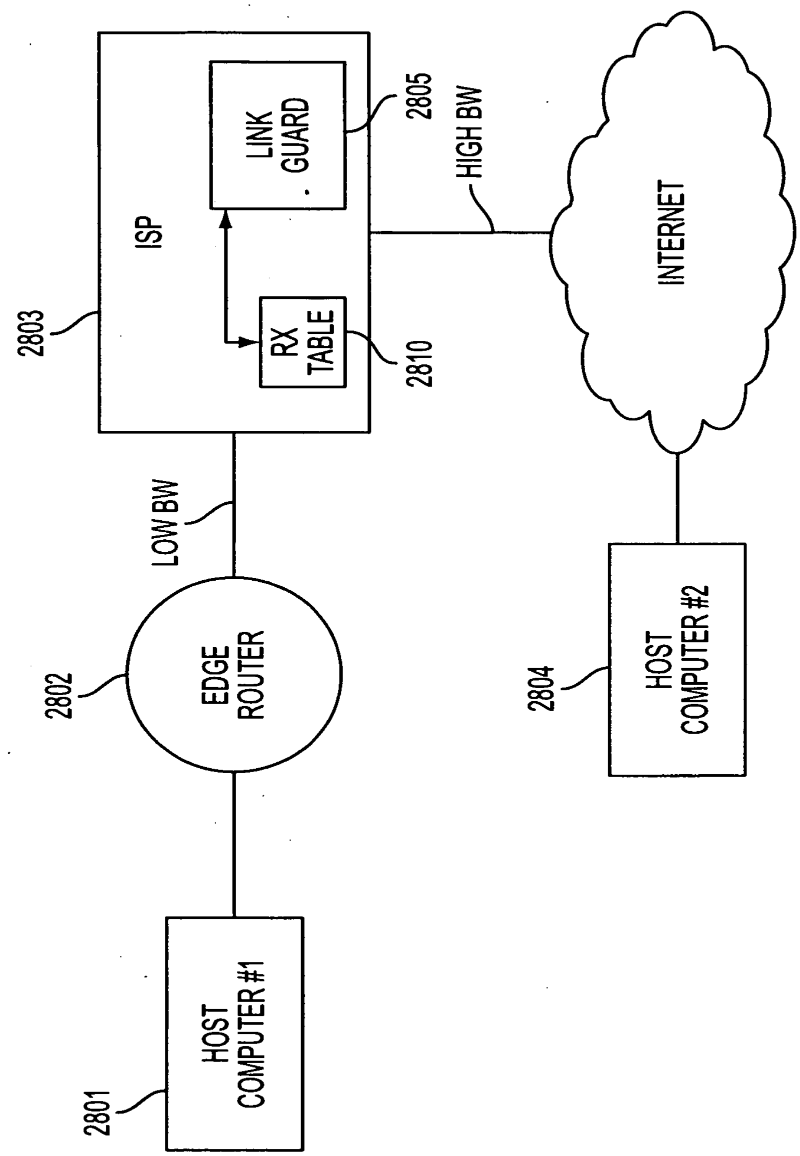

2801
HOST COMPUTER #1
2802
EDGE ROUTER
LOW BW
2803
ISP
RX TABLE
2810
LINK GUARD
2805
HIGH BW
INTERNET
2804
HOST COMPUTER #2

FIG. 28

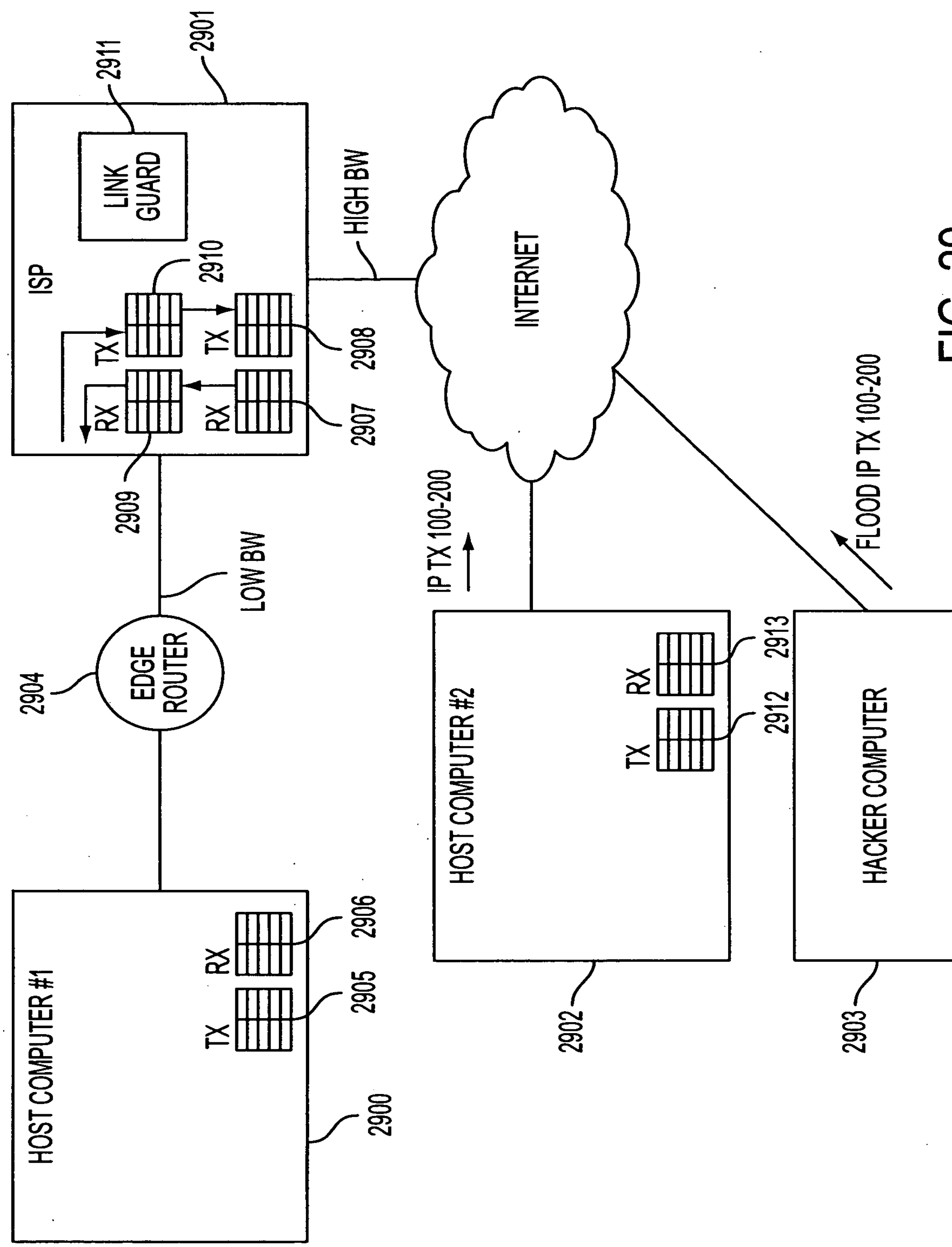

2901
2911
LINK GUARD
ISP
2910
TX
TX
RX
RX
2908
2907
2909
HIGH BW
INTERNET
LOW BW
EDGE ROUTER
2904
IP TX 100-200
FLOOD IP TX 100-200
HOST COMPUTER #2
TX
RX
2912
2913
HACKER COMPUTER
HOST COMPUTER #1
TX
RX
2905
2906
2900
2902
2903

FIG. 29

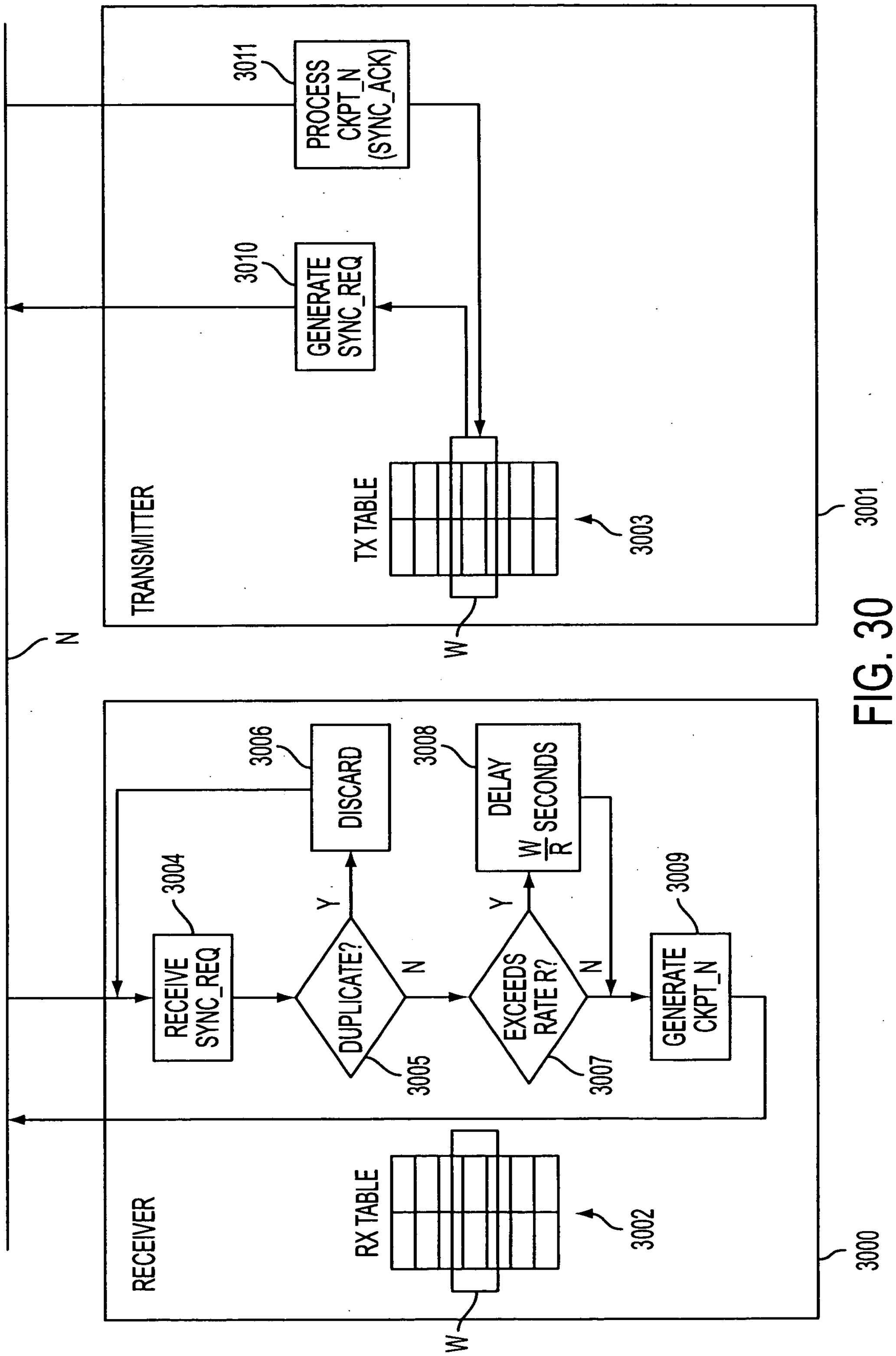

TRANSMITTER
PROCESS CKPT_N (SYNC_ACK)
3011
GENERATE SYNC_REQ
3010
TX TABLE
3003
3001
W
N
RECEIVER
RECEIVE SYNC_REQ
3004
DUPLICATE?
3005
Y
N
DISCARD
3006
EXCEEDS RATE R?
3007
Y
N
DELAY W/R SECONDS
3008
GENERATE CKPT_N
3009
RX TABLE
3002
3000
W

FIG. 30

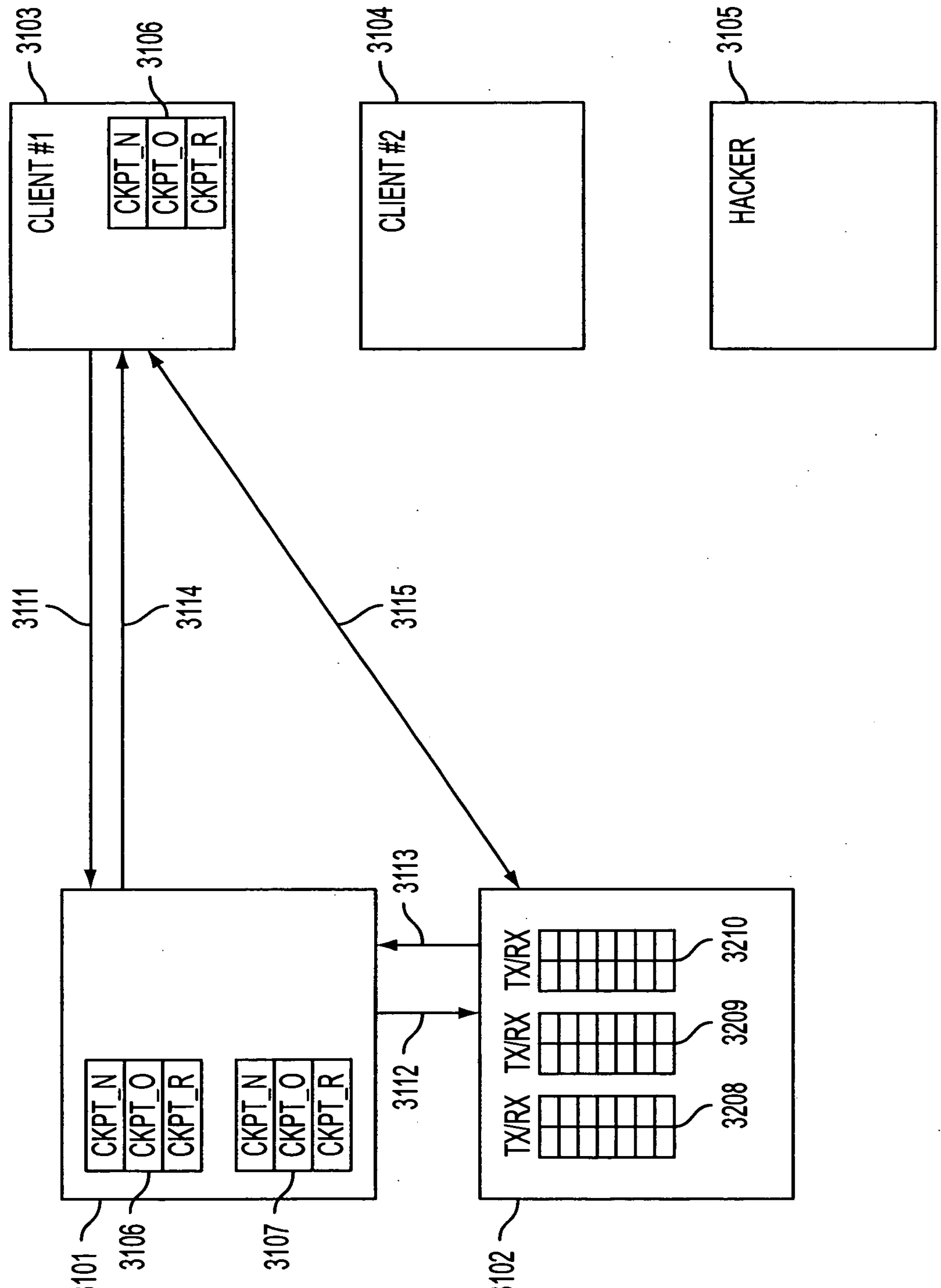

3103
3106
CLIENT #1
CKPT_N
CKPT_O
CKPT_R
3104
CLIENT #2
3105
HACKER
3111
3114
3115
3113
3112
3101
3106
3107
CKPT_N
CKPT_O
CKPT_R
CKPT_N
CKPT_O
CKPT_R
3102
TX/RX
TX/RX
TX/RX
3208
3209
3210

FIG. 31

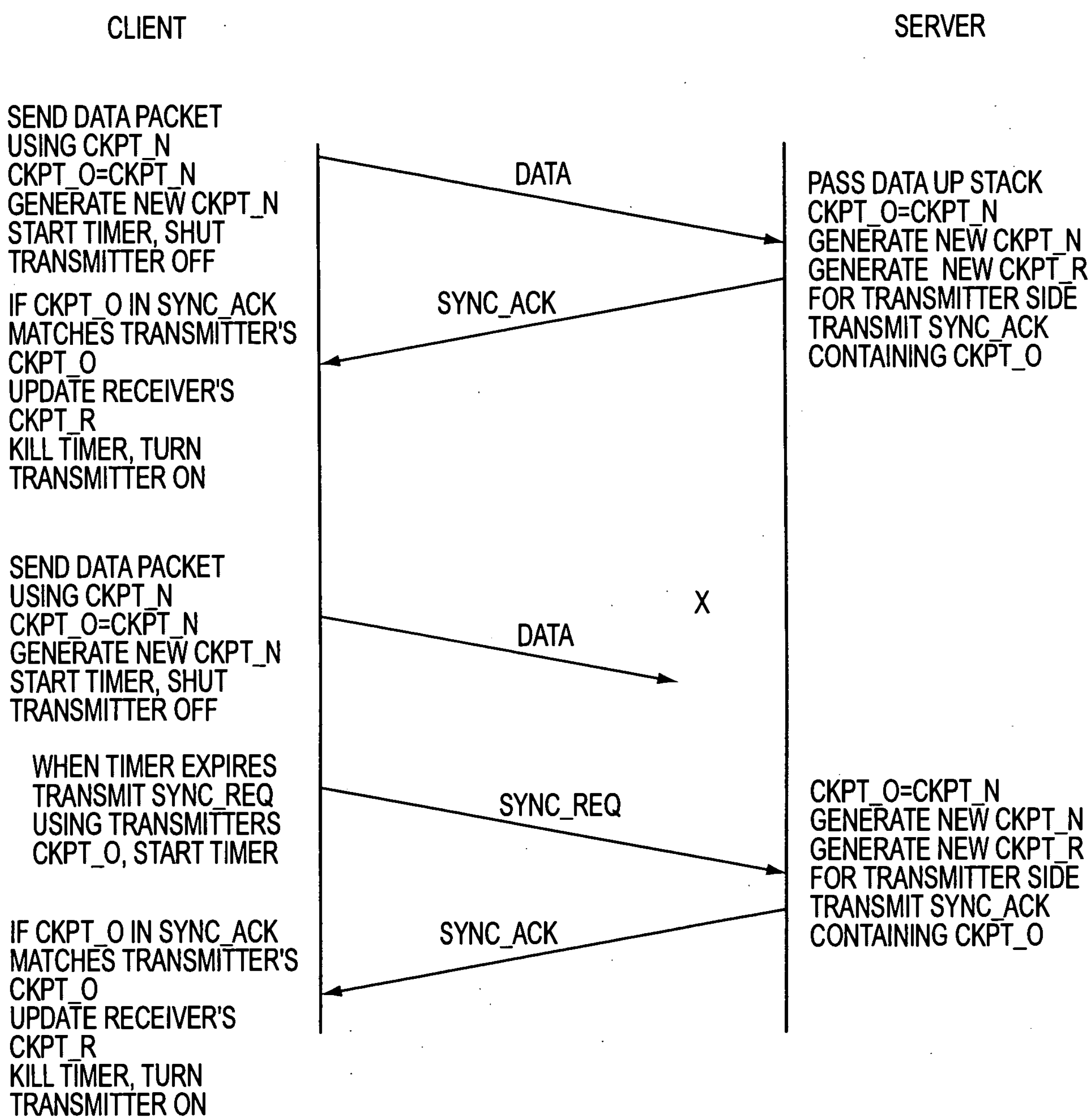

CLIENT
SERVER
SEND DATA PACKET USING CKPT_N CKPT_O=CKPT_N GENERATE NEW CKPT_N START TIMER, SHUT TRANSMITTER OFF
DATA
PASS DATA UP STACK CKPT_O=CKPT_N GENERATE NEW CKPT_N GENERATE NEW CKPT_R FOR TRANSMITTER SIDE TRANSMIT SYNC_ACK CONTAINING CKPT_O
SYNC_ACK
IF CKPT_O IN SYNC_ACK MATCHES TRANSMITTER'S CKPT_O UPDATE RECEIVER'S CKPT_R KILL TIMER, TURN TRANSMITTER ON
SEND DATA PACKET USING CKPT_N CKPT_O=CKPT_N GENERATE NEW CKPT_N START TIMER, SHUT TRANSMITTER OFF
X
DATA
WHEN TIMER EXPIRES TRANSMIT SYNC_REQ USING TRANSMITTERS CKPT_O, START TIMER
SYNC_REQ
CKPT_O=CKPT_N GENERATE NEW CKPT_N GENERATE NEW CKPT_R FOR TRANSMITTER SIDE TRANSMIT SYNC_ACK CONTAINING CKPT_O
SYNC_ACK
IF CKPT_O IN SYNC_ACK MATCHES TRANSMITTER'S CKPT_O UPDATE RECEIVER'S CKPT_R KILL TIMER, TURN TRANSMITTER ON

FIG. 32

IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

CROSS-REFERENCE TO RELATED APPLICATIONS

[01] This application is a divisional application of 09/504,783 (filed February 15, 2000), which claims priority from and is a continuation-in-part of previously filed U.S. application serial number 09/429,643 (filed October 29, 1999). The subject matter of that application, which is bodily incorporated herein, derives from provisional U.S. application numbers 60/106,261 (filed October 30, 1998) and 60/137,704 (filed June 7, 1999).

BACKGROUND OF THE INVENTION

[02] A tremendous variety of methods have been proposed and implemented to provide security and anonymity for communications over the Internet. The variety stems, in part, from the different needs of different Internet users. A basic heuristic framework to aid in discussing these different security techniques is illustrated in FIG. 1. Two terminals, an originating terminal 100 and a destination terminal 110 are in communication over the Internet. It is desired for the communications to be secure, that is, immune to eavesdropping. For example, terminal 100 may transmit secret information to terminal 110 over the Internet 107. Also, it may be desired to prevent an eavesdropper from discovering that terminal 100 is in communication with terminal 110. For example, if terminal 100 is a user and terminal 110 hosts a web site, terminal 100's user may not want anyone in the intervening networks to know what web sites he is "visiting." Anonymity would thus be an issue, for example, for companies that want to keep their market research interests private and thus would prefer to prevent outsiders from knowing which web-sites or other Internet resources they are "visiting." These two security issues may be called data security and anonymity, respectively.

[03] Data security is usually tackled using some form of data encryption. An encryption key 48 is known at both the originating and terminating terminals 100 and 110. The keys may be private and public at the originating and destination terminals 100 and 110, respectively or they may be symmetrical keys (the same key is used by both parties to encrypt and decrypt). Many encryption methods are known and usable in this context.

[04] To hide traffic from a local administrator or ISP, a user can employ a local proxy server in communicating over an encrypted channel with an outside proxy such that the local administrator or ISP only sees the encrypted traffic. Proxy servers prevent destination servers from determining the identities of the originating clients. This system employs an intermediate server interposed between client and destination server. The destination server sees only the Internet Protocol (IP) address of the proxy server and not the originating client. The target server only sees the address of the outside proxy. This scheme relies on a trusted outside proxy server. Also, proxy schemes are vulnerable to traffic analysis methods of determining identities of transmitters and receivers. Another important limitation of proxy servers is that the server knows the identities of both calling and called parties. In many instances, an originating terminal, such as terminal A, would prefer to keep its identity concealed from the proxy, for example, if the proxy server is provided by an Internet service provider (ISP).

[05] To defeat traffic analysis, a scheme called Chaum's mixes employs a proxy server that transmits and receives fixed length messages, including dummy messages. Multiple originating terminals are connected through a mix (a server) to multiple target servers. It is difficult to tell which of the originating terminals are communicating to which of the connected target servers, and the dummy messages confuse eavesdroppers' efforts to detect communicating pairs by analyzing traffic. A drawback is that there is a risk that the mix server could be compromised. One way to deal with this risk is to spread the trust among multiple mixes. If one mix is compromised, the identities of the originating and target terminals may remain concealed. This strategy requires a number of alternative mixes so that the intermediate servers interposed between the originating and target terminals are not determinable except by compromising more than one mix. The strategy wraps the message with multiple layers of encrypted addresses. The first mix in a sequence can decrypt only the outer layer of the message to reveal the next destination mix in sequence. The second mix can decrypt the message to reveal the next mix and so on. The target server receives the message and, optionally, a multi-layer encrypted payload containing return information to send data back in the same fashion. The only way to defeat such a mix scheme is to collude among mixes. If the packets are all fixed-length and intermixed with dummy packets, there is no way to do any kind of traffic analysis.

[06] Still another anonymity technique, called 'crowds,' protects the identity of the originating terminal from the intermediate proxies by providing that originating terminals belong to groups of proxies called crowds. The crowd proxies are interposed between originating and target terminals. Each proxy through which the message is sent is randomly chosen by an upstream proxy. Each intermediate proxy can send the message either to another randomly chosen proxy in the "crowd" or to the destination. Thus, even crowd members cannot determine if a preceding proxy is the originator of the message or if it was simply passed from another proxy.

[07] ZKS (Zero-Knowledge Systems) Anonymous IP Protocol allows users to select up to any of five different pseudonyms, while desktop software encrypts outgoing traffic and wraps it in User Datagram Protocol (UDP) packets. The first server in a 2+-hop system gets the UDP packets, strips off one layer of encryption to add another, then sends the traffic to the next server, which strips off yet another layer of encryption and adds a new one. The user is permitted to control the number of hops. At the final server, traffic is decrypted with an untraceable IP address. The technique is called onion-routing. This method can be defeated using traffic analysis. For a simple example, bursts of packets from a user during low-duty periods can reveal the identities of sender and receiver.

[08] Firewalls attempt to protect LANs from unauthorized access and hostile exploitation or damage to computers connected to the LAN. Firewalls provide a server through which all access to the LAN must pass. Firewalls are centralized systems that require administrative overhead to maintain. They can be compromised by virtual-machine applications ("applets"). They instill a false sense of security that leads to security breaches for example by users sending sensitive information to servers outside the firewall or encouraging use of modems to sidestep the firewall security. Firewalls are not useful for distributed systems such as business travelers, extranets, small teams, etc.

SUMMARY OF THE INVENTION

[09] A secure mechanism for communicating over the internet, including a protocol referred to as the Tunneled Agile Routing Protocol (TARP), uses a unique two-layer encryption format and special TARP routers. TARP routers are similar in function to regular IP routers. Each TARP router has one or more IP addresses and uses normal IP protocol to send IP packet

messages ("packets" or "datagrams"). The IP packets exchanged between TARP terminals via TARP routers are actually encrypted packets whose true destination address is concealed except to TARP routers and servers. The normal or "clear" or "outside" IP header attached to TARP IP packets contains only the address of a next hop router or destination server. That is, instead of indicating a final destination in the destination field of the IP header, the TARP packet's IP header always points to a next-hop in a series of TARP router hops, or to the final destination. This means there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet since the destination could always be next-hop TARP router as well as the final destination.

[10] Each TARP packet's true destination is concealed behind a layer of encryption generated using a link key. The link key is the encryption key used for encrypted communication between the hops intervening between an originating TARP terminal and a destination TARP terminal. Each TARP router can remove the outer layer of encryption to reveal the destination router for each TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal by the sender/receiver IP numbers in the cleartext IP header.

[11] Once the outer layer of encryption is removed, the TARP router determines the final destination. Each TARP packet 140 undergoes a minimum number of hops to help foil traffic analysis. The hops may be chosen at random or by a fixed value. As a result, each TARP packet may make random trips among a number of geographically disparate routers before reaching its destination. Each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined. This feature is called *agile routing*. The fact that different packets take different routes provides distinct advantages by making it difficult for an interloper to obtain all the packets forming an entire multi-packet message. The associated advantages have to do with the inner layer of encryption discussed below. Agile routing is combined with another feature that furthers this purpose; a feature that ensures that any message is broken into multiple packets.

[12] The IP address of a TARP router can be changed, a feature called *IP agility*. Each TARP router, independently or under direction from another TARP terminal or router, can change its IP address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs.

[13] The message payload is hidden behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the intervening TARP routers. The session key is used to decrypt the payloads of the TARP packets permitting the data stream to be reconstructed.

[14] Communication may be made private using link and session keys, which in turn may be shared and used according to any desired method. For example, public/private keys or symmetric keys may be used.

[15] To transmit a data stream, a TARP originating terminal constructs a series of TARP packets from a series of IP packets generated by a network (IP) layer process. (Note that the terms "network layer," "data link layer," "application layer," etc. used in this specification correspond to the Open Systems Interconnection (OSI) network terminology.) The payloads of these packets are assembled into a block and chain-block encrypted using the session key. This assumes, of course, that all the IP packets are destined for the same TARP terminal. The block is then interleaved and the interleaved encrypted block is broken into a series of payloads, one for each TARP packet to be generated. Special TARP headers IP_T are then added to each payload using the IP headers from the data stream packets. The TARP headers can be identical to normal IP headers or customized in some way. They should contain a formula or data for deinterleaving the data at the destination TARP terminal, a time-to-live (TTL) parameter to indicate the number of hops still to be executed, a data type identifier which indicates whether the payload contains, for example, TCP or UDP data, the sender's TARP address, the destination TARP address, and an indicator as to whether the packet contains real or decoy data or a formula for filtering out decoy data if decoy data is spread in some way through the TARP payload data.

[16] Note that although chain-block encryption is discussed here with reference to the session key, any encryption method may be used. Preferably, as in chain block encryption, a method should be used that makes unauthorized decryption difficult without an entire result of the encryption process. Thus, by separating the encrypted block among multiple packets and making it difficult for an interloper to obtain access to all of such packets, the contents of the communications are provided an extra layer of security.

[17] Decoy or dummy data can be added to a stream to help foil traffic analysis by reducing the peak-to-average network load. It may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

[18] Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to portion, or entirety, of a message, and that portion or entirety then interleaved into a number of separate packets. Considering the agile IP routing of the packets, and the attendant difficulty of reconstructing an entire sequence of packets to form a single block-encrypted message element, decoy packets can significantly increase the difficulty of reconstructing an entire data stream.

[19] The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system. Because the encryption system described above is insertable between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the Network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the

difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

[20] IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

[21] As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicating that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. In addition, it may create a subprocess that maintains the original IP address and continues interacting with the attacker in some manner.

[22] Decoy packets may be generated by each TARP terminal on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more likely to insert decoy packets when a message stream is being received. Alternatively, if a large number of decoy packets is received along with regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis.

[23] In various other embodiments of the invention, a scalable version of the system may be constructed in which a plurality of IP addresses are preassigned to each pair of communicating nodes in the network. Each pair of nodes agrees upon an algorithm for "hopping" between IP addresses (both sending and receiving), such that an eavesdropper sees apparently continuously random IP address pairs (source and destination) for packets transmitted between the pair. Overlapping or "reusable" IP addresses may be allocated to different users on the same subnet, since each node merely verifies that a particular packet includes a valid source/destination pair from the agreed-upon algorithm. Source/destination pairs are preferably not reused between any two nodes during any given end-to-end session, though limited IP block sizes or lengthy sessions might require it.

[24] Further improvements described in this continuation-in-part application include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities

BRIEF DESCRIPTION OF THE DRAWINGS

[25] FIG. 1 is an illustration of secure communications over the Internet according to a prior art embodiment.

[26] FIG. 2 is an illustration of secure communications over the Internet according to a an embodiment of the invention.

[27] FIG. 3a is an illustration of a process of forming a tunneled IP packet according to an embodiment of the invention.

[28] FIG. 3b is an illustration of a process of forming a tunneled IP packet according to another embodiment of the invention.

[29] FIG. 4 is an illustration of an OSI layer location of processes that may be used to implement the invention.

[30] FIG. 5 is a flow chart illustrating a process for routing a tunneled packet according to an embodiment of the invention.

[31] FIG. 6 is a flow chart illustrating a process for forming a tunneled packet according to an embodiment of the invention.

[32] FIG. 7 is a flow chart illustrating a process for receiving a tunneled packet according to an embodiment of the invention.

[33] FIG. 8 shows how a secure session is established and synchronized between a client and a TARP router.

[34] FIG. 9 shows an IP address hopping scheme between a client computer and TARP router using transmit and receive tables in each computer.

[35] FIG. 10 shows physical link redundancy among three Internet Service Providers (ISPs) and a client computer.

[36] FIG. 11 shows how multiple IP packets can be embedded into a single "frame" such as an Ethernet frame, and further shows the use of a discriminator field to camouflage true packet recipients.

[37] FIG. 12A shows a system that employs hopped hardware addresses, hopped IP addresses, and hopped discriminator fields.

[38] FIG. 12B shows several different approaches for hopping hardware addresses, IP addresses, and discriminator fields in combination.

[39] FIG. 13 shows a technique for automatically re-establishing synchronization between sender and receiver through the use of a partially public sync value.

[40] FIG. 14 shows a "checkpoint" scheme for regaining synchronization between a sender and recipient.

[41] FIG. 15 shows further details of the checkpoint scheme of FIG. 14.

[42] FIG. 16 shows how two addresses can be decomposed into a plurality of segments for comparison with presence vectors.

[43] FIG. 17 shows a storage array for a receiver's active addresses.

[44] FIG. 18 shows the receiver's storage array after receiving a sync request.

[45] FIG. 19 shows the receiver's storage array after new addresses have been generated.

[46] FIG. 20 shows a system employing distributed transmission paths.

[47] FIG. 21 shows a plurality of link transmission tables that can be used to route packets in the system of FIG. 20.

[48] FIG. 22A shows a flowchart for adjusting weight value distributions associated with a plurality of transmission links.

[49] FIG. 22B shows a flowchart for setting a weight value to zero if a transmitter turns off.

[50] FIG. 23 shows a system employing distributed transmission paths with adjusted weight value distributions for each path.

[51] FIG. 24 shows an example using the system of FIG. 23.

[52] FIG. 25 shows a conventional domain-name look-up service.

[53] FIG. 26 shows a system employing a DNS proxy server with transparent VPN creation.

[54] FIG. 27 shows steps that can be carried out to implement transparent VPN creation based on a DNS look-up function.

[55] FIG. 28 shows a system including a link guard function that prevents packet overloading on a low-bandwidth link LOW BW.

[56] FIG. 29 shows one embodiment of a system employing the principles of FIG. 28.

[57] FIG. 30 shows a system that regulates packet transmission rates by throttling the rate at which synchronizations are performed.

[58] FIG. 31 shows a signaling server 3101 and a transport server 3102 used to establish a VPN with a client computer.

[59] FIG. 32 shows message flows relating to synchronization protocols of FIG. 31.

DETAILED DESCRIPTION OF THE INVENTION

[60] Referring to FIG. 2, a secure mechanism for communicating over the internet employs a number of special routers or servers, called TARP routers 122-127 that are similar to regular IP routers 128-132 in that each has one or more IP addresses and uses normal IP protocol to send normal-looking IP packet messages, called TARP packets 140. TARP packets 140 are identical to normal IP packet messages that are routed by regular IP routers 128-132 because each TARP packet 140 contains a destination address as in a normal IP packet. However, instead of indicating a final destination in the destination field of the IP header, the TARP packet's 140 IP header always points to a next-hop in a series of TARP router hops, or the final destination, TARP terminal 110. Because the header of the TARP packet contains only the next-hop destination, there is no overt indication from an intercepted TARP packet of the true destination of the TARP packet 140 since the destination could always be the next-hop TARP router as well as the final destination, TARP terminal 110.

[61] Each TARP packet's true destination is concealed behind an outer layer of encryption generated using a link key 146. The link key 146 is the encryption key used for encrypted communication between the end points (TARP terminals or TARP routers) of a single link in the chain of hops connecting the originating TARP terminal 100 and the destination TARP terminal 110. Each TARP router 122-127, using the link key 146 it uses to communicate with the previous hop in a chain, can use the link key to reveal the true destination of a TARP packet. To identify the link key needed to decrypt the outer layer of encryption of a TARP packet, a receiving TARP or routing terminal may identify the transmitting terminal (which may indicate the link key used) by the sender field of the clear IP header. Alternatively, this identity may be hidden behind another layer of encryption in available bits in the clear IP header. Each TARP router, upon receiving a TARP message, determines if the message is a TARP message by using authentication data in the TARP packet. This could be recorded in available bytes in the TARP packet's IP header. Alternatively, TARP packets could be authenticated by attempting to decrypt using the link key 146 and determining if the results are as expected. The former may have computational advantages because it does not involve a decryption process.

[62] Once the outer layer of decryption is completed by a TARP router 122-127, the TARP router determines the final destination. The system is preferably designed to cause each TARP packet 140 to undergo a minimum number of hops to help foil traffic analysis. The time to live counter in the IP header of the TARP message may be used to indicate a number of TARP router hops yet to be completed. Each TARP router then would decrement the counter and determine from that whether it should forward the TARP packet 140 to another TARP router 122-127 or to the destination TARP terminal 110. If the time to live counter is zero or below zero after decrementing, for an example of usage, the TARP router receiving the TARP packet 140 may forward the TARP packet 140 to the destination TARP terminal 110. If the time to live counter is above zero after decrementing, for an example of usage, the TARP router receiving the TARP packet 140 may forward the TARP packet 140 to a TARP router 122-127 that the current TARP terminal chooses at random. As a result, each TARP packet 140 is routed through some minimum number of hops of TARP routers 122-127 which are chosen at random.

[63] Thus, each TARP packet, irrespective of the traditional factors determining traffic in the Internet, makes random trips among a number of geographically disparate routers before reaching its destination and each trip is highly likely to be different for each packet composing a given message because each trip is independently randomly determined as described above. This feature is called *agile routing*. For reasons that will become clear shortly, the fact that different packets take different routes provides distinct advantages by making it difficult for an interloper to obtain all the packets forming an entire multi-packet message. Agile routing is combined with another feature that furthers this purpose, a feature that ensures that any message is broken into multiple packets.

[64] A TARP router receives a TARP packet when an IP address used by the TARP router coincides with the IP address in the TARP packet's IP header IP_C. The IP address of a TARP router, however, may not remain constant. To avoid and manage attacks, each TARP router, independently or under direction from another TARP terminal or router, may change its IP address. A separate, unchangeable identifier or address is also defined. This address, called the TARP address, is known only to TARP routers and terminals and may be correlated at any time by a TARP router or a TARP terminal using a Lookup Table (LUT). When a TARP router or terminal changes its IP address, it updates the other TARP routers and terminals which in turn update their respective LUTs. In reality, whenever a TARP router looks up the address of a destination in the encrypted header, it must convert a TARP address to a real IP address using its LUT.

[65] While every TARP router receiving a TARP packet has the ability to determine the packet's final destination, the message payload is embedded behind an inner layer of encryption in the TARP packet that can only be unlocked using a session key. The session key is not available to any of the TARP routers 122-127 intervening between the originating 100 and destination 110 TARP terminals. The session key is used to decrypt the payloads of the TARP packets 140 permitting an entire message to be reconstructed.

[66] In one embodiment, communication may be made private using link and session keys, which in turn may be shared and used according any desired method. For example, a public key or symmetric keys may be communicated between link or session endpoints using a public key method. Any of a variety of other mechanisms for securing data to ensure that only authorized computers can have access to the private information in the TARP packets 140 may be used as desired.

[67] Referring to FIG. 3a, to construct a series of TARP packets, a data stream 300 of IP packets 207a, 207b, 207c, etc., such series of packets being formed by a network (IP) layer process, is broken into a series of small sized segments. In the present example, equal-sized segments 1-9 are defined and used to construct a set of interleaved data packets A, B, and C. Here it is assumed that the number of interleaved packets A, B, and C formed is three and that the number of IP packets 207a-207c used to form the three interleaved packets A, B, and C is exactly three. Of course, the number of IP packets spread over a group of interleaved packets may be any convenient number as may be the number of interleaved packets over which the incoming data stream is spread. The latter, the number of interleaved packets over which the data stream is spread, is called the *interleave window*.

[68] To create a packet, the transmitting software interleaves the normal IP packets 207a *et. seq*. to form a new set of interleaved payload data 320. This payload data 320 is then encrypted using a session key to form a set of session-key-encrypted payload data 330, each of which, A, B, and C, will form the payload of a TARP packet. Using the IP header data, from the original packets 207a-207c, new TARP headers IP_T are formed. The TARP headers IP_T can be identical to normal IP headers or customized in some way. In a preferred embodiment, the TARP headers IP_T are IP headers with added data providing the following information required for routing and reconstruction of messages, some of which data is ordinarily, or capable of being, contained in normal IP headers:

1. A window sequence number – an identifier that indicates where the packet belongs in the original message sequence.
2. An interleave sequence number – an identifier that indicates the interleaving sequence used to form the packet so that the packet can be deinterleaved along with other packets in the interleave window.
3. A time-to-live (TTL) datum – indicates the number of TARP-router-hops to be executed before the packet reaches its destination. Note that the TTL parameter

may provide a datum to be used in a probabilistic formula for determining whether to route the packet to the destination or to another hop.

4. Data type identifier – indicates whether the payload contains, for example, TCP or UDP data.

5. Sender's address – indicates the sender's address in the TARP network.

6. Destination address – indicates the destination terminal's address in the TARP network.

7. Decoy/Real – an indicator of whether the packet contains real message data or dummy decoy data or a combination.

[69] Obviously, the packets going into a single interleave window must include only packets with a common destination. Thus, it is assumed in the depicted example that the IP headers of IP packets 207a-207c all contain the same destination address or at least will be received by the same terminal so that they can be deinterleaved. Note that dummy or decoy data or packets can be added to form a larger interleave window than would otherwise be required by the size of a given message. Decoy or dummy data can be added to a stream to help foil traffic analysis by leveling the load on the network. Thus, it may be desirable to provide the TARP process with an ability to respond to the time of day or other criteria to generate more decoy data during low traffic periods so that communication bursts at one point in the Internet cannot be tied to communication bursts at another point to reveal the communicating endpoints.

[70] Dummy data also helps to break the data into a larger number of inconspicuously-sized packets permitting the interleave window size to be increased while maintaining a reasonable size for each packet. (The packet size can be a single standard size or selected from a fixed range of sizes.) One primary reason for desiring for each message to be broken into multiple packets is apparent if a chain block encryption scheme is used to form the first encryption layer prior to interleaving. A single block encryption may be applied to a portion, or the entirety, of a message, and that portion or entirety then interleaved into a number of separate packets.

[71] Referring to FIG. 3b, in an alternative mode of TARP packet construction, a series of IP packets are accumulated to make up a predefined interleave window. The payloads of the packets are used to construct a single block 520 for chain block encryption using the session key.

The payloads used to form the block are presumed to be destined for the same terminal. The block size may coincide with the interleave window as depicted in the example embodiment of FIG. 3b. After encryption, the encrypted block is broken into separate payloads and segments which are interleaved as in the embodiment of Fig 3a. The resulting interleaved packets A, B, and C, are then packaged as TARP packets with TARP headers as in the Example of FIG. 3a. The remaining process is as shown in, and discussed with reference to, FIG. 3a.

[72] Once the TARP packets 340 are formed, each entire TARP packet 340, including the TARP header IP_T, is encrypted using the link key for communication with the first-hop-TARP router. The first hop TARP router is randomly chosen. A final unencrypted IP header IP_C is added to each encrypted TARP packet 340 to form a normal IP packet 360 that can be transmitted to a TARP router. Note that the process of constructing the TARP packet 360 does not have to be done in stages as described. The above description is just a useful heuristic for describing the final product, namely, the TARP packet.

[73] Note that, TARP header IP_T could be a completely custom header configuration with no similarity to a normal IP header except that it contain the information identified above. This is so since this header is interpreted by only TARP routers.

[74] The above scheme may be implemented entirely by processes operating between the data link layer and the network layer of each server or terminal participating in the TARP system. Referring to FIG. 4, a TARP transceiver 405 can be an originating terminal 100, a destination terminal 110, or a TARP router 122-127. In each TARP Transceiver 405, a transmitting process is generated to receive normal packets from the Network (IP) layer and generate TARP packets for communication over the network. A receiving process is generated to receive normal IP packets containing TARP packets and generate from these normal IP packets which are "passed up" to the Network (IP) layer. Note that where the TARP Transceiver 405 is a router, the received TARP packets 140 are not processed into a stream of IP packets 415 because they need only be authenticated as proper TARP packets and then passed to another TARP router or a TARP destination terminal 110. The intervening process, a "TARP Layer" 420, could be combined with either the data link layer 430 or the Network layer 410. In either case, it would intervene between the data link layer 430 so that the process would receive regular IP packets containing embedded TARP packets and "hand up" a series of reassembled IP packets to the Network layer 410. As an example of combining the TARP layer 420 with the data link layer 430, a program may augment the normal processes running a communications card, for example, an Ethernet card. Alternatively, the TARP layer processes may form part of a dynamically loadable module that is loaded and executed to support communications between the network and data link layers.

[75] Because the encryption system described above can be inserted between the data link and network layers, the processes involved in supporting the encrypted communication may be completely transparent to processes at the IP (network) layer and above. The TARP processes may also be completely transparent to the data link layer processes as well. Thus, no operations at or above the network layer, or at or below the data link layer, are affected by the insertion of the TARP stack. This provides additional security to all processes at or above the network layer, since the difficulty of unauthorized penetration of the network layer (by, for example, a hacker) is increased substantially. Even newly developed servers running at the session layer leave all processes below the session layer vulnerable to attack. Note that in this architecture, security is distributed. That is, notebook computers used by executives on the road, for example, can communicate over the Internet without any compromise in security.

[76] Note that IP address changes made by TARP terminals and routers can be done at regular intervals, at random intervals, or upon detection of "attacks." The variation of IP addresses hinders traffic analysis that might reveal which computers are communicating, and also provides a degree of immunity from attack. The level of immunity from attack is roughly proportional to the rate at which the IP address of the host is changing.

[77] As mentioned, IP addresses may be changed in response to attacks. An attack may be revealed, for example, by a regular series of messages indicates that a router is being probed in some way. Upon detection of an attack, the TARP layer process may respond to this event by changing its IP address. To accomplish this, the TARP process will construct a TARP-formatted message, in the style of Internet Control Message Protocol (ICMP) datagrams as an example; this message will contain the machine's TARP address, its previous IP address, and its new IP address. The TARP layer will transmit this packet to at least one known TARP router; then upon

receipt and validation of the message, the TARP router will update its LUT with the new IP address for the stated TARP address. The TARP router will then format a similar message, and broadcast it to the other TARP routers so that they may update their LUTs. Since the total number of TARP routers on any given subnet is expected to be relatively small, this process of updating the LUTs should be relatively fast. It may not, however, work as well when there is a relatively large number of TARP routers and/or a relatively large number of clients; this has motivated a refinement of this architecture to provide scalability; this refinement has led to a second embodiment, which is discussed below.

[78] Upon detection of an attack, the TARP process may also create a subprocess that maintains the original IP address and continues interacting with the attacker. The latter may provide an opportunity to trace the attacker or study the attacker's methods (called "fishbowling" drawing upon the analogy of a small fish in a fish bowl that "thinks" it is in the ocean but is actually under captive observation). A history of the communication between the attacker and the abandoned (fishbowled) IP address can be recorded or transmitted for human analysis or further synthesized for purposes of responding in some way.

[79] As mentioned above, decoy or dummy data or packets can be added to outgoing data streams by TARP terminals or routers. In addition to making it convenient to spread data over a larger number of separate packets, such decoy packets can also help to level the load on inactive portions of the Internet to help foil traffic analysis efforts.

[80] Decoy packets may be generated by each TARP terminal 100, 110 or each router 122-127 on some basis determined by an algorithm. For example, the algorithm may be a random one which calls for the generation of a packet on a random basis when the terminal is idle. Alternatively, the algorithm may be responsive to time of day or detection of low traffic to generate more decoy packets during low traffic times. Note that packets are preferably generated in groups, rather than one by one, the groups being sized to simulate real messages. In addition, so that decoy packets may be inserted in normal TARP message streams, the background loop may have a latch that makes it more likely to insert decoy packets when a message stream is being received. That is, when a series of messages are received, the decoy packet generation rate may be increased. Alternatively, if a large number of decoy packets is received along with

regular TARP packets, the algorithm may increase the rate of dropping of decoy packets rather than forwarding them. The result of dropping and generating decoy packets in this way is to make the apparent incoming message size different from the apparent outgoing message size to help foil traffic analysis. The rate of reception of packets, decoy or otherwise, may be indicated to the decoy packet dropping and generating processes through perishable decoy and regular packet counters. (A perishable counter is one that resets or decrements its value in response to time so that it contains a high value when it is incremented in rapid succession and a small value when incremented either slowly or a small number of times in rapid succession.) Note that destination TARP terminal 110 may generate decoy packets equal in number and size to those TARP packets received to make it appear it is merely routing packets and is therefore not the destination terminal.

[81] Referring to FIG. 5, the following particular steps may be employed in the above-described method for routing TARP packets.

- S0. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.
- S2. The TARP packet may be probed in some way to authenticate the packet before attempting to decrypt it using the link key. That is, the router may determine that the packet is an authentic TARP packet by performing a selected operation on some data included with the clear IP header attached to the encrypted TARP packet contained in the payload. This makes it possible to avoid performing decryption on packets that are not authentic TARP packets.
- S3. The TARP packet is decrypted to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.
- S4. If the packet is a decoy packet, the perishable decoy counter is incremented.
- S5. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the router may choose to throw it away. If the received packet is a decoy packet and it is determined that it should be thrown away (S6), control returns to step S0.

- S7. The TTL parameter of the TARP header is decremented and it is determined if the TTL parameter is greater than zero.
- S8. If the TTL parameter is greater than zero, a TARP address is randomly chosen from a list of TARP addresses maintained by the router and the link key and IP address corresponding to that TARP address memorized for use in creating a new IP packet containing the TARP packet.
- S9. If the TTL parameter is zero or less, the link key and IP address corresponding to the TARP address of the destination are memorized for use in creating the new IP packet containing the TARP packet.
- S10. The TARP packet is encrypted using the memorized link key.
- S11. An IP header is added to the packet that contains the stored IP address, the encrypted TARP packet wrapped with an IP header, and the completed packet transmitted to the next hop or destination.

[82] Referring to FIG. 6, the following particular steps may be employed in the above-described method for generating TARP packets.

- S20. A background loop operation applies an algorithm that determines the generation of decoy IP packets. The loop is interrupted when a data stream containing IP packets is received for transmission.
- S21. The received IP packets are grouped into a set consisting of messages with a constant IP destination address. The set is further broken down to coincide with a maximum size of an interleave window The set is encrypted, and interleaved into a set of payloads destined to become TARP packets.
- S22. The TARP address corresponding to the IP address is determined from a lookup table and stored to generate the TARP header. An initial TTL count is generated and stored in the header. The TTL count may be random with minimum and maximum values or it may be fixed or determined by some other parameter.
- S23. The window sequence numbers and interleave sequence numbers are recorded in the TARP headers of each packet.
- S24. One TARP router address is randomly chosen for each TARP packet and the IP address corresponding to it stored for use in the clear IP header. The link key corresponding to this

router is identified and used to encrypt TARP packets containing interleaved and encrypted data and TARP headers.

- S25. A clear IP header with the first hop router's real IP address is generated and added to each of the encrypted TARP packets and the resulting packets.

[83] Referring to FIG. 7, the following particular steps may be employed in the above-described method for receiving TARP packets.

- S40. A background loop operation is performed which applies an algorithm which determines the generation of decoy IP packets. The loop is interrupted when an encrypted TARP packet is received.
- S42. The TARP packet may be probed to authenticate the packet before attempting to decrypt it using the link key.
- S43. The TARP packet is decrypted with the appropriate link key to expose the destination TARP address and an indication of whether the packet is a decoy packet or part of a real message.
- S44. If the packet is a decoy packet, the perishable decoy counter is incremented.
- S45. Based on the decoy generation/dropping algorithm and the perishable decoy counter value, if the packet is a decoy packet, the receiver may choose to throw it away.
- S46. The TARP packets are cached until all packets forming an interleave window are received.
- S47. Once all packets of an interleave window are received, the packets are deinterleaved.
- S48. The packets block of combined packets defining the interleave window is then decrypted using the session key.
- S49. The decrypted block is then divided using the window sequence data and the IP_T headers are converted into normal IP_C headers. The window sequence numbers are integrated in the IP_C headers.
- S50. The packets are then handed up to the IP layer processes.

<u>1. SCALABILITY ENHANCEMENTS</u>

[84] The IP agility feature described above relies on the ability to transmit IP address changes to all TARP routers. The embodiments including this feature will be referred to as "boutique" embodiments due to potential limitations in scaling these features up for a large network, such as the Internet. (The "boutique" embodiments would, however, be robust for use in smaller networks, such as small virtual private networks, for example). One problem with the boutique embodiments is that if IP address changes are to occur frequently, the message traffic required to update all routers sufficiently quickly creates a serious burden on the Internet when the TARP router and/or client population gets large. The bandwidth burden added to the networks, for example in ICMP packets, that would be used to update all the TARP routers could overwhelm the Internet for a large scale implementation that approached the scale of the Internet. In other words, the boutique system's scalability is limited.

[85] A system can be constructed which trades some of the features of the above embodiments to provide the benefits of IP agility without the additional messaging burden. This is accomplished by IP address-hopping according to shared algorithms that govern IP addresses used between links participating in communications sessions between nodes such as TARP nodes. (Note that the IP hopping technique is also applicable to the boutique embodiment.) The IP agility feature discussed with respect to the boutique system can be modified so that it becomes decentralized under this scalable regime and governed by the above-described shared algorithm. Other features of the boutique system may be combined with this new type of IP-agility.

[86] The new embodiment has the advantage of providing IP agility governed by a local algorithm and set of IP addresses exchanged by each communicating pair of nodes. This local governance is session-independent in that it may govern communications between a pair of nodes, irrespective of the session or end points being transferred between the directly communicating pair of nodes.

[87] In the scalable embodiments, blocks of IP addresses are allocated to each node in the network. (This scalability will increase in the future, when Internet Protocol addresses are increased to 128-bit fields, vastly increasing the number of distinctly addressable nodes). Each node can thus use any of the IP addresses assigned to that node to communicate with other nodes in the network. Indeed, each pair of communicating nodes can use a plurality of source IP addresses and destination IP addresses for communicating with each other.

[88] Each communicating pair of nodes in a chain participating in any session stores two blocks of IP addresses, called netblocks, and an algorithm and randomization seed for selecting, from each netblock, the next pair of source/destination IP addresses that will be used to transmit the next message. In other words, the algorithm governs the sequential selection of IP-address pairs, one sender and one receiver IP address, from each netblock. The combination of algorithm, seed, and netblock (IP address block) will be called a "hopblock." A router issues separate transmit and receive hopblocks to its clients. The send address and the receive address of the IP header of each outgoing packet sent by the client are filled with the send and receive IP addresses generated by the algorithm. The algorithm is "clocked" (indexed) by a counter so that each time a pair is used, the algorithm turns out a new transmit pair for the next packet to be sent.

[89] The router's receive hopblock is identical to the client's transmit hopblock. The router uses the receive hopblock to predict what the send and receive IP address pair for the next expected packet from that client will be. Since packets can be received out of order, it is not possible for the router to predict with certainty what IP address pair will be on the next sequential packet. To account for this problem, the router generates a range of predictions encompassing the number of possible transmitted packet send/receive addresses, of which the next packet received could leap ahead. Thus, if there is a vanishingly small probability that a given packet will arrive at the router ahead of 5 packets transmitted by the client before the given packet, then the router can generate a series of 6 send/receive IP address pairs (or "hop window") to compare with the next received packet. When a packet is received, it is marked in the hop window as such, so that a second packet with the same IP address pair will be discarded. If an out-of-sequence packet does not arrive within a predetermined timeout period, it can be requested for retransmission or simply discarded from the receive table, depending upon the protocol in use for that communications session, or possibly by convention.

[90] When the router receives the client's packet, it compares the send and receive IP addresses of the packet with the next N predicted send and receive IP address pairs and rejects the packet if it is not a member of this set. Received packets that do not have the predicted source/destination IP addresses falling with the window are rejected, thus thwarting possible hackers. (With the number of possible combinations, even a fairly large window would be hard

to fall into at random.) If it is a member of this set, the router accepts the packet and processes it further. This link-based IP-hopping strategy, referred to as "IHOP," is a network element that stands on its own and is not necessarily accompanied by elements of the boutique system described above. If the routing agility feature described in connection with the boutique embodiment is combined with this link-based IP-hopping strategy, the router's next step would be to decrypt the TARP header to determine the destination TARP router for the packet and determine what should be the next hop for the packet. The TARP router would then forward the packet to a random TARP router or the destination TARP router with which the source TARP router has a link-based IP hopping communication established.

[91] Figure 8 shows how a client computer 801 and a TARP router 811 can establish a secure session. When client 801 seeks to establish an IHOP session with TARP router 811, the client 801 sends "secure synchronization" request ("SSYN") packet 821 to the TARP router 811. This SYN packet 821 contains the client's 801 authentication token, and may be sent to the router 811 in an encrypted format. The source and destination IP numbers on the packet 821 are the client's 801 current fixed IP address, and a "known" fixed IP address for the router 811. (For security purposes, it may be desirable to reject any packets from outside of the local network that are destined for the router's known fixed IP address.) Upon receipt and validation of the client's 801 SSYN packet 821, the router 811 responds by sending an encrypted "secure synchronization acknowledgment" ("SSYN ACK") 822 to the client 801. This SSYN ACK 822 will contain the transmit and receive hopblocks that the client 801 will use when communicating with the TARP router 811. The client 801 will acknowledge the TARP router's 811 response packet 822 by generating an encrypted SSYN ACK ACK packet 823 which will be sent from the client's 801 fixed IP address and to the TARP router's 811 known fixed IP address. The client 801 will simultaneously generate a SSYN ACK ACK packet; this SSYN ACK packet, referred to as the Secure Session Initiation (SSI) packet 824, will be sent with the first {sender, receiver} IP pair in the client's transmit table 921 (FIG. 9), as specified in the transmit hopblock provided by the TARP router 811 in the SSYN ACK packet 822. The TARP router 811 will respond to the SSI packet 824 with an SSI ACK packet 825, which will be sent with the first {sender, receiver} IP pair in the TARP router's transmit table 923. Once these packets have been successfully exchanged, the secure communications session is established, and all further secure communications between the client 801 and the TARP router 811 will be conducted via this secure session, as long as synchronization is maintained. If synchronization is lost, then the client 801 and TARP router 802 may re-establish the secure session by the procedure outlined in Figure 8 and described above.

[92] While the secure session is active, both the client 901 and TARP router 911 (FIG. 9) will maintain their respective transmit tables 921, 923 and receive tables 922, 924, as provided by the TARP router during session synchronization 822. It is important that the sequence of IP pairs in the client's transmit table 921 be identical to those in the TARP router's receive table 924; similarly, the sequence of IP pairs in the client's receive table 922 must be identical to those in the router's transmit table 923. This is required for the session synchronization to be maintained. The client 901 need maintain only one transmit table 921 and one receive table 922 during the course of the secure session. Each sequential packet sent by the client 901 will employ the next {send, receive} IP address pair in the transmit table, regardless of TCP or UDP session. The TARP router 911 will expect each packet arriving from the client 901 to bear the next IP address pair shown in its receive table.

[93] Since packets can arrive out of order, however, the router 911 can maintain a "look ahead" buffer in its receive table, and will mark previously-received IP pairs as invalid for future packets; any future packet containing an IP pair that is in the look-ahead buffer but is marked as previously received will be discarded. Communications from the TARP router 911 to the client 901 are maintained in an identical manner; in particular, the router 911 will select the next IP address pair from its transmit table 923 when constructing a packet to send to the client 901, and the client 901 will maintain a look-ahead buffer of expected IP pairs on packets that it is receiving. Each TARP router will maintain separate pairs of transmit and receive tables for each client that is currently engaged in a secure session with or through that TARP router.

[94] While clients receive their hopblocks from the first server linking them to the Internet, routers exchange hopblocks. When a router establishes a link-based IP-hopping communication regime with another router, each router of the pair exchanges its transmit hopblock. The transmit hopblock of each router becomes the receive hopblock of the other router. The communication between routers is governed as described by the example of a client sending a packet to the first router.

[95] While the above strategy works fine in the IP milieu, many local networks that are connected to the Internet are Ethernet systems. In Ethernet, the IP addresses of the destination devices must be translated into hardware addresses, and vice versa, using known processes ("address resolution protocol," and "reverse address resolution protocol"). However, if the link-based IP-hopping strategy is employed, the correlation process would become explosive and burdensome. An alternative to the link-based IP hopping strategy may be employed within an Ethernet network. The solution is to provide that the node linking the Internet to the Ethernet (call it the border node) use the link-based IP-hopping communication regime to communicate with nodes outside the Ethernet LAN. Within the Ethernet LAN, each TARP node would have a single IP address which would be addressed in the conventional way. Instead of comparing the {sender, receiver} IP address pairs to authenticate a packet, the intra-LAN TARP node would use one of the IP header extension fields to do so. Thus, the border node uses an algorithm shared by the intra-LAN TARP node to generate a symbol that is stored in the free field in the IP header, and the intra-LAN TARP node generates a range of symbols based on its prediction of the next expected packet to be received from that particular source IP address. The packet is rejected if it does not fall into the set of predicted symbols (for example, numerical values) or is accepted if it does. Communications from the intra-LAN TARP node to the border node are accomplished in the same manner, though the algorithm will necessarily be different for security reasons. Thus, each of the communicating nodes will generate transmit and receive tables in a similar manner to that of Figure 9; the intra-LAN TARP nodes transmit table will be identical to the border node's receive table, and the intra-LAN TARP node's receive table will be identical to the border node's transmit table.

[96] The algorithm used for IP address-hopping can be any desired algorithm. For example, the algorithm can be a given pseudo-random number generator that generates numbers of the range covering the allowed IP addresses with a given seed. Alternatively, the session participants can assume a certain type of algorithm and specify simply a parameter for applying the

algorithm. For example the assumed algorithm could be a particular pseudo-random number generator and the session participants could simply exchange seed values.

[97] Note that there is no permanent physical distinction between the originating and destination terminal nodes. Either device at either end point can initiate a synchronization of the pair. Note also that the authentication/synchronization-request (and acknowledgment) and hopblock-exchange may all be served by a single message so that separate message exchanges may not be required.

[98] As another extension to the stated architecture, multiple physical paths can be used by a client, in order to provide link redundancy and further thwart attempts at denial of service and traffic monitoring. As shown in Figure 10, for example, client 1001 can establish three simultaneous sessions with each of three TARP routers provided by different ISPs 1011, 1012, 1013. As an example, the client 1001 can use three different telephone lines 1021, 1022, 1023 to connect to the ISPs, or two telephone lines and a cable modem, etc. In this scheme, transmitted packets will be sent in a random fashion among the different physical paths. This architecture provides a high degree of communications redundancy, with improved immunity from denial-of-service attacks and traffic monitoring.

2. FURTHER EXTENSIONS

[99] The following describes various extensions to the techniques, systems, and methods described above. As described above, the security of communications occurring between computers in a computer network (such as the Internet, an Ethernet, or others) can be enhanced by using seemingly random source and destination Internet Protocol (IP) addresses for data packets transmitted over the network. This feature prevents eavesdroppers from determining which computers in the network are communicating with each other while permitting the two communicating computers to easily recognize whether a given received data packet is legitimate or not. In one embodiment of the above-described systems, an IP header extension field is used to authenticate incoming packets on an Ethernet.

[100] Various extensions to the previously described techniques described herein include: (1) use of hopped hardware or "MAC" addresses in broadcast type network; (2) a self-

synchronization technique that permits a computer to automatically regain synchronization with a sender; (3) synchronization algorithms that allow transmitting and receiving computers to quickly re-establish synchronization in the event of lost packets or other events; and (4) a fast-packet rejection mechanism for rejecting invalid packets. Any or all of these extensions can be combined with the features described above in any of various ways.

A. Hardware Address Hopping

[101] Internet protocol-based communications techniques on a LAN—or across any dedicated physical medium—typically embed the IP packets within lower-level packets, often referred to as "frames." As shown in FIG. 11, for example, a first Ethernet frame 1150 comprises a frame header 1101 and two embedded IP packets IP1 and IP2, while a second Ethernet frame 1160 comprises a different frame header 1104 and a single IP packet IP3. Each frame header generally includes a source hardware address 1101A and a destination hardware address 1101B; other well-known fields in frame headers are omitted from FIG. 11 for clarity. Two hardware nodes communicating over a physical communication channel insert appropriate source and destination hardware addresses to indicate which nodes on the channel or network should receive the frame.

[102] It may be possible for a nefarious listener to acquire information about the contents of a frame and/or its communicants by examining frames on a local network rather than (or in addition to) the IP packets themselves. This is especially true in broadcast media, such as Ethernet, where it is necessary to insert into the frame header the hardware address of the machine that generated the frame and the hardware address of the machine to which frame is being sent. All nodes on the network can potentially "see" all packets transmitted across the network. This can be a problem for secure communications, especially in cases where the communicants do not want for any third party to be able to identify who is engaging in the information exchange. One way to address this problem is to push the address-hopping scheme down to the hardware layer. In accordance with various embodiments of the invention, hardware addresses are "hopped" in a manner similar to that used to change IP addresses, such that a listener cannot determine which hardware node generated a particular message nor which node is the intended recipient.

[103] FIG. 12A shows a system in which Media Access Control ("MAC") hardware addresses are "hopped" in order to increase security over a network such as an Ethernet. While the description refers to the exemplary case of an Ethernet environment, the inventive principles are equally applicable to other types of communications media. In the Ethernet case, the MAC address of the sender and receiver are inserted into the Ethernet frame and can be observed by anyone on the LAN who is within the broadcast range for that frame. For secure communications, it becomes desirable to generate frames with MAC addresses that are not attributable to any specific sender or receiver.

[104] As shown in FIG. 12A, two computer nodes 1201 and 1202 communicate over a communication channel such as an Ethernet. Each node executes one or more application programs 1203 and 1218 that communicate by transmitting packets through communication software 1204 and 1217, respectively. Examples of application programs include video conferencing, e-mail, word processing programs, telephony, and the like. Communication software 1204 and 1217 can comprise, for example, an OSI layered architecture or "stack" that standardizes various services provided at different levels of functionality.

[105] The lowest levels of communication software 1204 and 1217 communicate with hardware components 1206 and 1214 respectively, each of which can include one or more registers 1207 and 1215 that allow the hardware to be reconfigured or controlled in accordance with various communication protocols. The hardware components (an Ethernet network interface card, for example) communicate with each other over the communication medium. Each hardware component is typically pre-assigned a fixed hardware address or MAC number that identifies the hardware component to other nodes on the network. One or more interface drivers control the operation of each card and can, for example, be configured to accept or reject packets from certain hardware addresses. As will be described in more detail below, various embodiments of the inventive principles provide for "hopping" different addresses using one or more algorithms and one or more moving windows that track a range of valid addresses to validate received packets. Packets transmitted according to one or more of the inventive principles will be generally referred to as "secure" packets or "secure communications" to

differentiate them from ordinary data packets that are transmitted in the clear using ordinary, machine-correlated addresses.

[106] One straightforward method of generating non-attributable MAC addresses is an extension of the IP hopping scheme. In this scenario, two machines on the same LAN that desire to communicate in a secure fashion exchange random-number generators and seeds, and create sequences of quasi-random MAC addresses for synchronized hopping. The implementation and synchronization issues are then similar to that of IP hopping.

[107] This approach, however, runs the risk of using MAC addresses that are currently active on the LAN—which, in turn, could interrupt communications for those machines. Since an Ethernet MAC address is at present 48 bits in length, the chance of randomly misusing an active MAC address is actually quite small. However, if that figure is multiplied by a large number of nodes (as would be found on an extensive LAN), by a large number of frames (as might be the case with packet voice or streaming video), and by a large number of concurrent Virtual Private Networks (VPNs), then the chance that a non-secure machine's MAC address could be used in an address-hopped frame can become non-trivial. In short, any scheme that runs even a small risk of interrupting communications for other machines on the LAN is bound to receive resistance from prospective system administrators. Nevertheless, it is technically feasible, and can be implemented without risk on a LAN on which there is a small number of machines, or if all of the machines on the LAN are engaging in MAC-hopped communications.

[108] Synchronized MAC address hopping may incur some overhead in the course of session establishment, especially if there are multiple sessions or multiple nodes involved in the communications. A simpler method of randomizing MAC addresses is to allow each node to receive and process *every* incident frame on the network. Typically, each network interface driver will check the destination MAC address in the header of every incident frame to see if it matches that machine's MAC address; if there is no match, then the frame is discarded. In one embodiment, however, these checks can be disabled, and every incident packet is passed to the TARP stack for processing. This will be referred to as "promiscuous" mode, since every incident frame is processed. Promiscuous mode allows the sender to use completely random, unsynchronized MAC addresses, since the destination machine is guaranteed to process the frame. The decision as to whether the packet was truly intended for that machine is handled by the TARP stack, which checks the source and destination IP addresses for a match in its IP synchronization tables. If no match is found, the packet is discarded; if there is a match, the packet is unwrapped, the inner header is evaluated, and if the inner header indicates that the packet is destined for that machine then the packet is forwarded to the IP stack—otherwise it is discarded.

[109] One disadvantage of purely-random MAC address hopping is its impact on processing overhead; that is, since every incident frame must be processed, the machine's CPU is engaged considerably more often than if the network interface driver is discriminating and rejecting packets unilaterally. A compromise approach is to select either a single fixed MAC address or a small number of MAC addresses (e.g., one for each virtual private network on an Ethernet) to use for MAC-hopped communications, regardless of the actual recipient for which the message is intended. In this mode, the network interface driver can check each incident frame against one (or a few) pre-established MAC addresses, thereby freeing the CPU from the task of physical-layer packet discrimination. This scheme does not betray any useful information to an interloper on the LAN; in particular, every secure packet can already be identified by a unique packet type in the outer header. However, since all machines engaged in secure communications would either be using the same MAC address, or be selecting from a small pool of predetermined MAC addresses, the association between a specific machine and a specific MAC address is effectively broken.

[110] In this scheme, the CPU will be engaged more often than it would be in non-secure communications (or in synchronized MAC address hopping), since the network interface driver cannot always unilaterally discriminate between secure packets that are destined for that machine, and secure packets from other VPNs. However, the non-secure traffic is easily eliminated at the network interface, thereby reducing the amount of processing required of the CPU. There are boundary conditions where these statements would not hold, of course—e.g., if *all* of the traffic on the LAN is secure traffic, then the CPU would be engaged to the same degree as it is in the purely-random address hopping case; alternatively, if each VPN on the LAN uses a different MAC address, then the network interface can perfectly discriminate secure frames destined for the local machine from those constituting other VPNs. These are engineering tradeoffs that might be best handled by providing administrative options for the users when installing the software and/or establishing VPNs.

[111] Even in this scenario, however, there still remains a slight risk of selecting MAC addresses that are being used by one or more nodes on the LAN. One solution to this problem is to formally assign one address or a range of addresses for use in MAC-hopped communications. This is typically done via an assigned numbers registration authority; e.g., in the case of Ethernet, MAC address ranges are assigned to vendors by the Institute of Electrical and Electronics Engineers (IEEE). A formally-assigned range of addresses would ensure that secure frames do not conflict with any properly-configured and properly-functioning machines on the LAN.

[112] Reference will now be made to FIGS. 12A and 12B in order to describe the many combinations and features that follow the inventive principles. As explained above, two computer nodes 1201 and 1202 are assumed to be communicating over a network or communication medium such as an Ethernet. A communication protocol in each node (1204 and 1217, respectively) contains a modified element 1205 and 1216 that performs certain functions that deviate from the standard communication protocols. In particular, computer node 1201 implements a first "hop" algorithm 1208X that selects seemingly random source and destination IP addresses (and, in one embodiment, seemingly random IP header discriminator fields) in order to transmit each packet to the other computer node. For example, node 1201 maintains a transmit table 1208 containing triplets of source (S), destination (D), and discriminator fields (DS) that are inserted into outgoing IP packet headers. The table is generated through the use of an appropriate algorithm (e.g., a random number generator that is seeded with an appropriate seed) that is known to the recipient node 1202. As each new IP packet is formed, the next sequential entry out of the sender's transmit table 1208 is used to populate the IP source, IP destination, and IP header extension field (e.g., discriminator field). It will be appreciated that the transmit table need not be created in advance but could instead be created on-the-fly by executing the algorithm when each packet is formed.

[113] At the receiving node 1202, the same IP hop algorithm 1222X is maintained and used to generate a receive table 1222 that lists valid triplets of source IP address, destination IP address, and discriminator field. This is shown by virtue of the first five entries of transmit table 1208 matching the second five entries of receive table 1222. (The tables may be slightly offset at any particular time due to lost packets, misordered packets, or transmission delays). Additionally, node 1202 maintains a receive window W3 that represents a list of valid IP source, IP destination, and discriminator fields that will be accepted when received as part of an incoming IP packet. As packets are received, window W3 slides down the list of valid entries, such that the possible valid entries change over time. Two packets that arrive out of order but are nevertheless matched to entries within window W3 will be accepted; those falling outside of window W3 will be rejected as invalid. The length of window W3 can be adjusted as necessary to reflect network delays or other factors.

[114] Node 1202 maintains a similar transmit table 1221 for creating IP packets and frames destined for node 1201 using a potentially different hopping algorithm 1221X, and node 1201 maintains a matching receive table 1209 using the same algorithm 1209X. As node 1202 transmits packets to node 1201 using seemingly random IP source, IP destination, and/or discriminator fields, node 1201 matches the incoming packet values to those falling within window W1 maintained in its receive table. In effect, transmit table 1208 of node 1201 is synchronized (i.e., entries are selected in the same order) to receive table 1222 of receiving node 1202. Similarly, transmit table 1221 of node 1202 is synchronized to receive table 1209 of node 1201. It will be appreciated that although a common algorithm is shown for the source, destination and discriminator fields in FIG. 12A (using, e.g., a different seed for each of the three fields), an entirely different algorithm could in fact be used to establish values for each of these fields. It will also be appreciated that one or two of the fields can be "hopped" rather than all three as illustrated.

[115] In accordance with another aspect of the invention, hardware or "MAC" addresses are hopped instead of or in addition to IP addresses and/or the discriminator field in order to improve security in a local area or broadcast-type network. To that end, node 1201 further maintains a transmit table 1210 using a transmit algorithm 1210X to generate source and destination

hardware addresses that are inserted into frame headers (e.g., fields 1101A and 1101B in FIG. 11) that are synchronized to a corresponding receive table 1224 at node 1202. Similarly, node 1202 maintains a different transmit table 1223 containing source and destination hardware addresses that is synchronized with a corresponding receive table 1211 at node 1201. In this manner, outgoing hardware frames appear to be originating from and going to completely random nodes on the network, even though each recipient can determine whether a given packet is intended for it or not. It will be appreciated that the hardware hopping feature can be implemented at a different level in the communications protocol than the IP hopping feature (e.g., in a card driver or in a hardware card itself to improve performance).

[116] FIG. 12B shows three different embodiments or modes that can be employed using the aforementioned principles. In a first mode referred to as "promiscuous" mode, a common hardware address (e.g., a fixed address for source and another for destination) or else a completely random hardware address is used by all nodes on the network, such that a particular packet cannot be attributed to any one node. Each node must initially accept all packets containing the common (or random) hardware address and inspect the IP addresses or discriminator field to determine whether the packet is intended for that node. In this regard, either the IP addresses or the discriminator field or both can be varied in accordance with an algorithm as described above. As explained previously, this may increase each node's overhead since additional processing is involved to determine whether a given packet has valid source and destination hardware addresses.

[117] In a second mode referred to as "promiscuous per VPN" mode, a small set of fixed hardware addresses are used, with a fixed source/destination hardware address used for all nodes communicating over a virtual private network. For example, if there are six nodes on an Ethernet, and the network is to be split up into two private virtual networks such that nodes on one VPN can communicate with only the other two nodes on its own VPN, then two sets of hardware addresses could be used: one set for the first VPN and a second set for the second VPN. This would reduce the amount of overhead involved in checking for valid frames since only packets arriving from the designated VPN would need to be checked. IP addresses and one or more discriminator fields could still be hopped as before for secure communication within the

VPN. Of course, this solution compromises the anonymity of the VPNs (i.e., an outsider can easily tell what traffic belongs in which VPN, though he cannot correlate it to a specific machine/person). It also requires the use of a discriminator field to mitigate the vulnerability to certain types of DoS attacks. (For example, without the discriminator field, an attacker on the LAN could stream frames containing the MAC addresses being used by the VPN; rejecting those frames could lead to excessive processing overhead. The discriminator field would provide a low-overhead means of rejecting the false packets.)

[118] In a third mode referred to as "hardware hopping" mode, hardware addresses are varied as illustrated in FIG. 12A, such that hardware source and destination addresses are changed constantly in order to provide non-attributable addressing. Variations on these embodiments are of course possible, and the invention is not intended to be limited in any respect by these illustrative examples.

B. Extending the Address Space

[119] Address hopping provides security and privacy. However, the level of protection is limited by the number of addresses in the blocks being hopped. A hopblock denotes a field or fields modulated on a packet-wise basis for the purpose of providing a VPN. For instance, if two nodes communicate with IP address hopping using hopblocks of 4 addresses (2 bits) each, there would be 16 possible address-pair combinations. A window of size 16 would result in most address pairs being accepted as valid most of the time. This limitation can be overcome by using a discriminator field in addition to or instead of the hopped address fields. The discriminator field would be hopped in exactly the same fashion as the address fields and it would be used to determine whether a packet should be processed by a receiver.

[120] Suppose that two clients, each using four-bit hopblocks, would like the same level of protection afforded to clients communicating via IP hopping between two A blocks (24 address bits eligible for hopping). A discriminator field of 20 bits, used in conjunction with the 4 address bits eligible for hopping in the IP address field, provides this level of protection. A 24-bit discriminator field would provide a similar level of protection if the address fields were not hopped or ignored. Using a discriminator field offers the following advantages: (1) an arbitrarily high level of protection can be provided, and (2) address hopping is unnecessary to provide protection. This may be important in environments where address hopping would cause routing problems.

C. Synchronization Techniques

[121] It is generally assumed that once a sending node and receiving node have exchanged algorithms and seeds (or similar information sufficient to generate quasi-random source and destination tables), subsequent communication between the two nodes will proceed smoothly. Realistically, however, two nodes may lose synchronization due to network delays or outages, or other problems. Consequently, it is desirable to provide means for re-establishing synchronization between nodes in a network that have lost synchronization.

[122] One possible technique is to require that each node provide an acknowledgment upon successful receipt of each packet and, if no acknowledgment is received within a certain period of time, to re-send the unacknowledged packet. This approach, however, drives up overhead costs and may be prohibitive in high-throughput environments such as streaming video or audio, for example.

[123] A different approach is to employ an automatic synchronizing technique that will be referred to herein as "self-synchronization." In this approach, synchronization information is embedded into each packet, thereby enabling the receiver to re-synchronize itself upon receipt of a single packet if it determines that is has lost synchronization with the sender. (If communications are already in progress, and the receiver determines that it is still in sync with the sender, then there is no need to re-synchronize.) A receiver could detect that it was out of synchronization by, for example, employing a "dead-man" timer that expires after a certain period of time, wherein the timer is reset with each valid packet. A time stamp could be hashed into the public sync field (see below) to preclude packet-retry attacks.

[124] In one embodiment, a "sync field" is added to the header of each packet sent out by the sender. This sync field could appear in the clear or as part of an encrypted portion of the packet. Assuming that a sender and receiver have selected a random-number generator (RNG) and seed value, this combination of RNG and seed can be used to generate a random-number sequence

(RNS). The RNS is then used to generate a sequence of source/destination IP pairs (and, if desired, discriminator fields and hardware source and destination addresses), as described above. It is not necessary, however, to generate the entire sequence (or the first N-1 values) in order to generate the Nth random number in the sequence; if the sequence index N is known, the random value corresponding to that index can be directly generated (see below). Different RNGs (and seeds) with different fundamental periods could be used to generate the source and destination IP sequences, but the basic concepts would still apply. For the sake of simplicity, the following discussion will assume that IP source and destination address pairs (only) are hopped using a single RNG sequencing mechanism.

[125] In accordance with a "self-synchronization" feature, a sync field in each packet header provides an index (i.e., a sequence number) into the RNS that is being used to generate IP pairs. Plugging this index into the RNG that is being used to generate the RNS yields a specific random number value, which in turn yields a specific IP pair. That is, an IP pair can be generated directly from knowledge of the RNG, seed, and index number; it is not necessary, in this scheme, to generate the entire sequence of random numbers that precede the sequence value associated with the index number provided.

[126] Since the communicants have presumably previously exchanged RNGs and seeds, the only new information that must be provided in order to generate an IP pair is the sequence number. If this number is provided by the sender in the packet header, then the receiver need only plug this number into the RNG in order to generate an IP pair – and thus verify that the IP pair appearing in the header of the packet is valid. In this scheme, if the sender and receiver lose synchronization, the receiver can immediately re-synchronize upon receipt of a single packet by simply comparing the IP pair in the packet header to the IP pair generated from the index number. Thus, synchronized communications can be resumed upon receipt of a single packet, making this scheme ideal for multicast communications. Taken to the extreme, it could obviate the need for synchronization tables entirely; that is, the sender and receiver could simply rely on the index number in the sync field to validate the IP pair on each packet, and thereby eliminate the tables entirely.

[127] The aforementioned scheme may have some inherent security issues associated with it — namely, the placement of the sync field. If the field is placed in the outer header, then an interloper could observe the values of the field and their relationship to the IP stream. This could potentially compromise the algorithm that is being used to generate the IP-address sequence, which would compromise the security of the communications. If, however, the value is placed in the inner header, then the sender must decrypt the inner header before it can extract the sync value and validate the IP pair; this opens up the receiver to certain types of denial-of-service (DoS) attacks, such as packet replay. That is, if the receiver must decrypt a packet before it can validate the IP pair, then it could potentially be forced to expend a significant amount of processing on decryption if an attacker simply retransmits previously valid packets. Other attack methodologies are possible in this scenario.

[128] A possible compromise between algorithm security and processing speed is to split up the sync value between an inner (encrypted) and outer (unencrypted) header. That is, if the sync value is sufficiently long, it could potentially be split into a rapidly-changing part that can be viewed in the clear, and a fixed (or very slowly changing) part that must be protected. The part that can be viewed in the clear will be called the "public sync" portion and the part that must be protected will be called the "private sync" portion.

[129] Both the public sync and private sync portions are needed to generate the complete sync value. The private portion, however, can be selected such that it is fixed or will change only occasionally. Thus, the private sync value can be stored by the recipient, thereby obviating the need to decrypt the header in order to retrieve it. If the sender and receiver have previously agreed upon the frequency with which the private part of the sync will change, then the receiver can selectively decrypt a single header in order to extract the new private sync if the communications gap that has led to lost synchronization has exceeded the lifetime of the previous private sync. This should not represent a burdensome amount of decryption, and thus should not open up the receiver to denial-of-service attack simply based on the need to occasionally decrypt a single header.

[130] One implementation of this is to use a hashing function with a one-to-one mapping to generate the private and public sync portions from the sync value. This implementation is shown

in FIG. 13, where (for example) a first ISP 1302 is the sender and a second ISP 1303 is the receiver. (Other alternatives are possible from FIG. 13.) A transmitted packet comprises a public or "outer" header 1305 that is not encrypted, and a private or "inner" header 1306 that is encrypted using for example a link key. Outer header 1305 includes a public sync portion while inner header 1306 contains the private sync portion. A receiving node decrypts the inner header using a decryption function 1307 in order to extract the private sync portion. This step is necessary only if the lifetime of the currently buffered private sync has expired. (If the currently-buffered private sync is still valid, then it is simply extracted from memory and "added" (which could be an inverse hash) to the public sync, as shown in step 1308.) The public and decrypted private sync portions are combined in function 1308 in order to generate the combined sync 1309. The combined sync (1309) is then fed into the RNG (1310) and compared to the IP address pair (1311) to validate or reject the packet.

[131] An important consideration in this architecture is the concept of "future" and "past" where the public sync values are concerned. Though the sync values, themselves, should be random to prevent spoofing attacks, it may be important that the receiver be able to quickly identify a sync value that has already been sent — even if the packet containing that sync value was never actually received by the receiver. One solution is to hash a time stamp or sequence number into the public sync portion, which could be quickly extracted, checked, and discarded, thereby validating the public sync portion itself.

[132] In one embodiment, packets can be checked by comparing the source/destination IP pair generated by the sync field with the pair appearing in the packet header. If (1) they match, (2) the time stamp is valid, and (3) the dead-man timer has expired, then re-synchronization occurs; otherwise, the packet is rejected. If enough processing power is available, the dead-man timer and synchronization tables can be avoided altogether, and the receiver would simply resynchronize (e.g., validate) on every packet.

[133] The foregoing scheme may require large-integer (e.g., 160-bit) math, which may affect its implementation. Without such large-integer registers, processing throughput would be affected, thus potentially affecting security from a denial-of-service standpoint. Nevertheless, as large-integer math processing features become more prevalent, the costs of implementing such a feature will be reduced.

D. Other Synchronization Schemes

[134] As explained above, if W or more consecutive packets are lost between a transmitter and receiver in a VPN (where W is the window size), the receiver's window will not have been updated and the transmitter will be transmitting packets not in the receiver's window. The sender and receiver will not recover synchronization until perhaps the random pairs in the window are repeated by chance. Therefore, there is a need to keep a transmitter and receiver in synchronization whenever possible and to re-establish synchronization whenever it is lost.

[135] A "checkpoint" scheme can be used to regain synchronization between a sender and a receiver that have fallen out of synchronization. In this scheme, a checkpoint message comprising a random IP address pair is used for communicating synchronization information. In one embodiment, two messages are used to communicate synchronization information between a sender and a recipient:

1. SYNC_REQ is a message used by the sender to indicate that it wants to synchronize; and
2. SYNC_ACK is a message used by the receiver to inform the transmitter that it has been synchronized.

[136] According to one variation of this approach, both the transmitter and receiver maintain three checkpoints (see FIG. 14):

1. In the transmitter, ckpt_o ("checkpoint old") is the IP pair that was used to re-send the last SYNC_REQ packet to the receiver. In the receiver, ckpt_o ("checkpoint old") is the IP pair that receives repeated SYNC_REQ packets from the transmitter.
2. In the transmitter, ckpt_n ("checkpoint new") is the IP pair that will be used to send the next SYNC_REQ packet to the receiver. In the receiver, ckpt_n ("checkpoint new") is the IP pair that receives a new SYNC_REQ packet from the transmitter and which causes the receiver's window to be re-aligned, ckpt_o set to ckpt_n, a new ckpt_n to be generated and a new ckpt_r to be generated.

3. In the transmitter, ckpt_r is the IP pair that will be used to send the next SYNC_ACK packet to the receiver. In the receiver, ckpt_r is the IP pair that receives a new SYNC_ACK packet from the transmitter and which causes a new ckpt_n to be generated. Since SYNC_ACK is transmitted from the receiver ISP to the sender ISP, the transmitter ckpt_r refers to the ckpt_r of the receiver and the receiver ckpt_r refers to the ckpt_r of the transmitter (see FIG. 14).

[137] When a transmitter initiates synchronization, the IP pair it will use to transmit the next data packet is set to a predetermined value and when a receiver first receives a SYNC_REQ, the receiver window is updated to be centered on the transmitter's next IP pair. This is the primary mechanism for checkpoint synchronization.

[138] Synchronization can be initiated by a packet counter (e.g., after every N packets transmitted, initiate a synchronization) or by a timer (every S seconds, initiate a synchronization) or a combination of both. See FIG. 15. From the transmitter's perspective, this technique operates as follows: (1) Each transmitter periodically transmits a "sync request" message to the receiver to make sure that it is in sync. (2) If the receiver is still in sync, it sends back a "sync ack" message. (If this works, no further action is necessary). (3) If no "sync ack" has been received within a period of time, the transmitter retransmits the sync request again. If the transmitter reaches the next checkpoint without receiving a "sync ack" response, then synchronization is broken, and the transmitter should stop transmitting. The transmitter will continue to send sync_reqs until it receives a sync_ack , at which point transmission is reestablished.

[139] From the receiver's perspective, the scheme operates as follows: (1) when it receives a "sync request" request from the transmitter, it advances its window to the next checkpoint position (even skipping pairs if necessary), and sends a "sync ack" message to the transmitter. If sync was never lost, then the "jump ahead" really just advances to the next available pair of addresses in the table (i.e., normal advancement).

[140] If an interloper intercepts the "sync request" messages and tries to interfere with communication by sending new ones, it will be ignored if the synchronization has been established or it it will actually help to re-establish synchronization.

[141] A window is realigned whenever a re-synchronization occurs. This realignment entails updating the receiver's window to straddle the address pairs used by the packet transmitted immediately after the transmission of the SYNC_REQ packet. Normally, the transmitter and receiver are in synchronization with one another. However, when network events occur, the receiver's window may have to be advanced by many steps during resynchronization. In this case, it is desirable to move the window ahead without having to step through the intervening random numbers sequentially. (This feature is also desirable for the auto-sync approach discussed above).

E. Random Number Generator with a Jump-Ahead capability

[142] An attractive method for generating randomly hopped addresses is to use identical random number generators in the transmitter and receiver and advance them as packets are transmitted and received. There are many random number generation algorithms that could be used. Each one has strengths and weaknesses for address hopping applications.

[143] Linear congruential random number generators (LCRs) are fast, simple and well characterized random number generators that can be made to jump ahead *n* steps efficiently. An LCR generates random numbers $X_1, X_2, X_3 \ldots X_k$ starting with seed X_0 using a recurrence

$$X_i=(a X_{i-1} + b) \bmod c \quad (1)$$

where a, b and c define a particular LCR. Another expression for X_i,

$$X_i=((a^i(X_0+b)-b)/(a-1)) \bmod c \quad (2)$$

enables the jump-ahead capability. The factor a^i can grow very large even for modest i if left unfettered. Therefore some special properties of the modulo operation can be used to control the size and processing time required to compute (2). (2) can be rewritten as:

$$X_i=(a^i (X_0(a-1)+b)-b)/(a-1) \bmod c \quad (3)$$

[144] It can be shown that:

$(a^i(X_0(a-1)+b)-b)/(a-1) \bmod c = ((a^i \bmod((a-1)c)(X_0(a-1)+b)-b)/(a-1)) \bmod c,$ (4)

$(X_0(a-1)+b)$ can be stored as $(X_0(a-1)+b) \bmod c$, b as b mod c and compute $a^i \bmod((a-1)c)$ (this requires O(log(i)) steps).

[145] A practical implementation of this algorithm would jump a fixed distance, n, between synchronizations; this is tantamount to synchronizing every *n* packets. The window would commence *n* IP pairs from the start of the previous window. Using X_j^w, the random number at the j^{th} checkpoint, as X_0 and *n* as *i*, a node can store $a^n \bmod((a-1)c)$ once per LCR and set

$X_{j+1}^w = X_{n(j+1)} = ((a^n \bmod((a-1)c)\,(X_j^w\,(a-1)+b)-b)/(a-1)) \bmod c,$ (5)

to generate the random number for the $j+1^{th}$ synchronization. Using this construction, a node could jump ahead an arbitrary (but fixed) distance between synchronizations in a constant amount of time (independent of n).

[146] Pseudo-random number generators, in general, and LCRs, in particular, will eventually repeat their cycles. This repetition may present vulnerability in the IP hopping scheme. An adversary would simply have to wait for a repeat to predict future sequences. One way of coping with this vulnerability is to create a random number generator with a known long cycle. A random sequence can be replaced by a new random number generator before it repeats. LCRs can be constructed with known long cycles. This is not currently true of many random number generators.

[147] Random number generators can be cryptographically insecure. An adversary can derive the RNG parameters by examining the output or part of the output. This is true of LCGs. This vulnerability can be mitigated by incorporating an encryptor, designed to scramble the output as part of the random number generator. The random number generator prevents an adversary from mounting an attack—e.g., a known plaintext attack—against the encryptor.

F. Random Number Generator Example

[148] Consider a RNG where a=31,b=4 and c=15. For this case equation (1) becomes:

$X_i = (31\,X_{i-1} + 4) \bmod 15.$ (6)

[149] If one sets X_0=1, equation (6) will produce the sequence 1, 5, 9, 13, 2, 6, 10, 14, 3, 7, 11, 0, 4, 8, 12. This sequence will repeat indefinitely. For a jump ahead of 3 numbers in this sequence a^n= 31^3=29791, c*(a-1)=15*30=450 and a^n mod((a-1)c) = 31^3mod(15*30)=29791mod(450)=91. Equation (5) becomes:

$$((91\ (X_i 30+4)-4)/30) \bmod 15 \tag{7}$$

[150] Table 1 shows the jump ahead calculations from (7) . The calculations start at 5 and jump ahead 3.

TABLE 1

I	X_i	(X_i30+4)	91 (X_i30+4)-4	((91 (X_i30+4)-4)/30	X_{i+3}
1	5	154	14010	467	2
4	2	64	5820	194	14
7	14	424	38580	1286	11
10	11	334	30390	1013	8
13	8	244	22200	740	5

G. Fast Packet Filter

[151] Address hopping VPNs must rapidly determine whether a packet has a valid header and thus requires further processing, or has an invalid header (a hostile packet) and should be immediately rejected. Such rapid determinations will be referred to as "fast packet filtering." This capability protects the VPN from attacks by an adversary who streams hostile packets at the receiver at a high rate of speed in the hope of saturating the receiver's processor (a so-called "denial of service" attack). Fast packet filtering is an important feature for implementing VPNs on shared media such as Ethernet.

[152] Assuming that all participants in a VPN share an unassigned "A" block of addresses, one possibility is to use an experimental "A" block that will never be assigned to any machine that is not address hopping on the shared medium. "A" blocks have a 24 bits of address that can be hopped as opposed to the 8 bits in "C" blocks. In this case a hopblock will be the "A" block. The use of the experimental "A" block is a likely option on an Ethernet because:

1. The addresses have no validity outside of the Ethernet and will not be routed out to a valid outside destination by a gateway.
2. There are 2^{24} (~16 million) addresses that can be hopped within each "A" block. This yields >280 trillion possible address pairs making it very unlikely that an adversary would guess a valid address. It also provides acceptably low probability of collision between separate VPNs (all VPNs on a shared medium independently generate random address pairs from the same "A" block).
3. The packets will not be received by someone on the Ethernet who is not on a VPN (unless the machine is in promiscuous mode) minimizing impact on non-VPN computers.

[153] The Ethernet example will be used to describe one implementation of fast packet filtering. The ideal algorithm would quickly examine a packet header, determine whether the packet is hostile, and reject any hostile packets or determine which active IP pair the packet header matches. The problem is a classical associative memory problem. A variety of techniques have been developed to solve this problem (hashing, B–trees etc). Each of these approaches has its strengths and weaknesses. For instance, hash tables can be made to operate quite fast in a statistical sense, but can occasionally degenerate into a much slower algorithm. This slowness can persist for a period of time. Since there is a need to discard hostile packets quickly at all times, hashing would be unacceptable.

H. Presence Vector Algorithm

[154] A presence vector is a bit vector of length 2^n that can be indexed by n-bit numbers (each ranging from 0 to 2^n-1). One can indicate the presence of k n-bit numbers (not necessarily unique), by setting the bits in the presence vector indexed by each number to 1. Otherwise, the bits in the presence vector are 0. An n-bit number, x, is one of the k numbers if and only if the x^{th} bit of the presence vector is 1. A fast packet filter can be implemented by indexing the presence vector and looking for a 1, which will be referred to as the "test."

[155] For example, suppose one wanted to represent the number 135 using a presence vector. The 135^{th} bit of the vector would be set. Consequently, one could very quickly determine whether an address of 135 was valid by checking only one bit: the 135^{th} bit. The presence vectors could be created in advance corresponding to the table entries for the IP addresses. In effect, the incoming addresses can be used as indices into a long vector, making comparisons very fast. As each RNG generates a new address, the presence vector is updated to reflect the information. As the window moves, the presence vector is updated to zero out addresses that are no longer valid.

[156] There is a trade-off between efficiency of the test and the amount of memory required for storing the presence vector(s). For instance, if one were to use the 48 bits of hopping addresses as an index, the presence vector would have to be 35 terabytes. Clearly, this is too large for practical purposes. Instead, the 48 bits can be divided into several smaller fields. For instance, one could subdivide the 48 bits into four 12-bit fields (see FIG. 16). This reduces the storage requirement to 2048 bytes at the expense of occasionally having to process a hostile packet. In effect, instead of one long presence vector, the decomposed address portions must match all four shorter presence vectors before further processing is allowed. (If the first part of the address portion doesn't match the first presence vector, there is no need to check the remaining three presence vectors).

[157] A presence vector will have a 1 in the y^{th} bit if and only if one or more addresses with a corresponding field of y are active. An address is active only if each presence vector indexed by the appropriate sub-field of the address is 1.

[158] Consider a window of 32 active addresses and 3 checkpoints. A hostile packet will be rejected by the indexing of one presence vector more than 99% of the time. A hostile packet will be rejected by the indexing of all 4 presence vectors more than 99.9999995% of the time. On average, hostile packets will be rejected in less than 1.02 presence vector index operations.

[159] The small percentage of hostile packets that pass the fast packet filter will be rejected when matching pairs are not found in the active window or are active checkpoints. Hostile packets that serendipitously match a header will be rejected when the VPN software attempts to decrypt the header. However, these cases will be extremely rare. There are many other ways this method can be configured to arbitrate the space/speed tradeoffs.

I. <u>Further Synchronization Enhancements</u>

[160] A slightly modified form of the synchronization techniques described above can be employed. The basic principles of the previously described checkpoint synchronization scheme remain unchanged. The actions resulting from the reception of the checkpoints are, however, slightly different. In this variation, the receiver will maintain between OoO ("Out of Order") and 2×WINDOW_SIZE+OoO active addresses (1 ≤OoO ≤WINDOW_SIZE and WINDOW_SIZE ≥1). OoO and WINDOW_SIZE are engineerable parameters, where OoO is the minimum number of addresses needed to accommodate lost packets due to events in the network or out of order arrivals and WINDOW_SIZE is the number of packets transmitted before a SYNC_REQ is issued. FIG. 17 depicts a storage array for a receiver's active addresses.

[161] The receiver starts with the first 2×WINDOW_SIZE addresses loaded and active (ready to receive data). As packets are received, the corresponding entries are marked as "used" and are no longer eligible to receive packets. The transmitter maintains a packet counter, initially set to 0, containing the number of data packets transmitted since the last *initial* transmission of a SYNC_REQ for which SYNC_ACK has been received. When the transmitter packet counter equals WINDOW_SIZE, the transmitter generates a SYNC_REQ and does its initial transmission. When the receiver receives a SYNC_REQ corresponding to its current CKPT_N, it generates the next WINDOW_SIZE addresses and starts loading them in order starting at the first location after the last active address wrapping around to the beginning of the array after the end of the array has been reached. The receiver's array might look like FIG. 18 when a SYNC_REQ has been received. In this case a couple of packets have been either lost or will be received out of order when the SYNC_REQ is received.

[162] FIG. 19 shows the receiver's array after the new addresses have been generated. If the transmitter does not receive a SYNC_ACK, it will re-issue the SYNC_REQ at regular intervals. When the transmitter receives a SYNC_ACK, the packet counter is decremented by WINDOW_SIZE. If the packet counter reaches 2×WINDOW_SIZE – OoO then the transmitter ceases sending data packets until the appropriate SYNC_ACK is finally received. The transmitter then resumes sending data packets. Future behavior is essentially a repetition of this initial cycle. The advantages of this approach are:

1. There is no need for an efficient jump ahead in the random number generator,

2. No packet is ever transmitted that does not have a corresponding entry in the receiver side
3. No timer based re-synchronization is necessary. This is a consequence of 2.
4. The receiver will always have the ability to accept data messages transmitted within OoO messages of the most recently transmitted message.

J. Distributed Transmission Path Variant

[163] Another embodiment incorporating various inventive principles is shown in FIG. 20. In this embodiment, a message transmission system includes a first computer 2001 in communication with a second computer 2002 through a network 2011 of intermediary computers. In one variant of this embodiment, the network includes two edge routers 2003 and 2004 each of which is linked to a plurality of Internet Service Providers (ISPs) 2005 through 2010. Each ISP is coupled to a plurality of other ISPs in an arrangement as shown in FIG. 20, which is a representative configuration only and is not intended to be limiting. Each connection between ISPs is labeled in FIG. 20 to indicate a specific physical transmission path (e.g., AD is a physical path that links ISP A (element 2005) to ISP D (element 2008)). Packets arriving at each edge router are selectively transmitted to one of the ISPs to which the router is attached on the basis of a randomly or quasi-randomly selected basis.

[164] As shown in FIG. 21, computer 2001 or edge router 2003 incorporates a plurality of link transmission tables 2100 that identify, for each potential transmission path through the network, valid sets of IP addresses that can be used to transmit the packet. For example, AD table 2101 contains a plurality of IP source/destination pairs that are randomly or quasi-randomly generated. When a packet is to be transmitted from first computer 2001 to second computer 2002, one of the link tables is randomly (or quasi-randomly) selected, and the next valid source/destination address pair from that table is used to transmit the packet through the network. If path AD is randomly selected, for example, the next source/destination IP address pair (which is pre-determined to transmit between ISP A (element 2005) and ISP B (element 2008)) is used to transmit the packet. If one of the transmission paths becomes degraded or inoperative, that link table can be set to a "down" condition as shown in table 2105, thus preventing addresses from being selected from that table. Other transmission paths would be unaffected by this broken link.

3. CONTINUATION-IN-PART IMPROVEMENTS

[165] The following describes various improvements and features that can be applied to the embodiments described above. The improvements include: (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities. Each is discussed separately below.

A. Load Balancer

[166] Various embodiments described above include a system in which a transmitting node and a receiving node are coupled through a plurality of transmission paths, and wherein successive packets are distributed quasi-randomly over the plurality of paths. See, for example, FIGS. 20 and 21 and accompanying description. The improvement extends this basic concept to encompass distributing packets across different paths in such a manner that the loads on the paths are generally balanced according to transmission link quality.

[167] In one embodiment, a system includes a transmitting node and a receiving node that are linked via a plurality of transmission paths having potentially varying transmission quality. Successive packets are transmitted over the paths based on a weight value distribution function for each path. The rate that packets will be transmitted over a given path can be different for each path. The relative "health" of each transmission path is monitored in order to identify paths that have become degraded. In one embodiment, the health of each path is monitored in the transmitter by comparing the number of packets transmitted to the number of packet acknowledgements received. Each transmission path may comprise a physically separate path (e.g., via dial-up phone line, computer network, router, bridge, or the like), or may comprise logically separate paths contained within a broadband communication medium (e.g., separate channels in an FDM, TDM, CDMA, or other type of modulated or unmodulated transmission link).

[168] When the transmission quality of a path falls below a predetermined threshold and there are other paths that can transmit packets, the transmitter changes the weight value used for that path, making it less likely that a given packet will be transmitted over that path. The weight will preferably be set no lower than a minimum value that keeps nominal traffic on the path. The weights of the other available paths are altered to compensate for the change in the affected path. When the quality of a path degrades to where the transmitter is turned off by the synchronization function (i.e., no packets are arriving at the destination), the weight is set to zero. If all transmitters are turned off, no packets are sent.

[169] Conventional TCP/IP protocols include a "throttling" feature that reduces the transmission rate of packets when it is determined that delays or errors are occurring in transmission. In this respect, timers are sometimes used to determine whether packets have been received. These conventional techniques for limiting transmission of packets, however, do not involve multiple transmission paths between two nodes wherein transmission across a particular path relative to the others is changed based on link quality.

[170] According to certain embodiments, in order to damp oscillations that might otherwise occur if weight distributions are changed drastically (e.g., according to a step function), a linear or an exponential decay formula can be applied to gradually decrease the weight value over time that a degrading path will be used. Similarly, if the health of a degraded path improves, the weight value for that path is gradually increased.

[171] Transmission link health can be evaluated by comparing the number of packets that are acknowledged within the transmission window (see embodiments discussed above) to the number of packets transmitted within that window and by the state of the transmitter (i.e., on or off). In other words, rather than accumulating general transmission statistics over time for a path, one specific implementation uses the "windowing" concepts described above to evaluate transmission path health.

[172] The same scheme can be used to shift virtual circuit paths from an "unhealthy" path to a "healthy" one, and to select a path for a new virtual circuit.

[173] FIG. 22A shows a flowchart for adjusting weight values associated with a plurality of transmission links. It is assumed that software executing in one or more computer nodes executes the steps shown in FIG. 22A. It is also assumed that the software can be stored on a computer-readable medium such as a magnetic or optical disk for execution by a computer.

[174] Beginning in step 2201, the transmission quality of a given transmission path is measured. As described above, this measurement can be based on a comparison between the number of packets transmitted over a particular link to the number of packet acknowledgements received over the link (e.g., per unit time, or in absolute terms). Alternatively, the quality can be evaluated by comparing the number of packets that are acknowledged within the transmission window to the number of packets that were transmitted within that window. In yet another variation, the number of missed synchronization messages can be used to indicate link quality. Many other variations are of course possible.

[175] In step 2202, a check is made to determine whether more than one transmitter (e.g., transmission path) is turned on. If not, the process is terminated and resumes at step 2201.

[176] In step 2203, the link quality is compared to a given threshold (e.g., 50%, or any arbitrary number). If the quality falls below the threshold, then in step 2207 a check is made to determine whether the weight is above a minimum level (e.g., 1%). If not, then in step 2209 the weight is set to the minimum level and processing resumes at step 2201. If the weight is above the minimum level, then in step 2208 the weight is gradually decreased for the path, then in step 2206 the weights for the remaining paths are adjusted accordingly to compensate (e.g., they are increased).

[177] If in step 2203 the quality of the path was greater than or equal to the threshold, then in step 2204 a check is made to determine whether the weight is less than a steady-state value for that path. If so, then in step 2205 the weight is increased toward the steady-state value, and in step 2206 the weights for the remaining paths are adjusted accordingly to compensate (e.g., they are decreased). If in step 2204 the weight is not less than the steady-state value, then processing resumes at step 2201 without adjusting the weights.

[178] The weights can be adjusted incrementally according to various functions, preferably by changing the value gradually. In one embodiment, a linearly decreasing function is used to adjust the weights; according to another embodiment, an exponential decay function is used. Gradually changing the weights helps to damp oscillators that might otherwise occur if the probabilities were abruptly.

[179] Although not explicitly shown in FIG. 22A the process can be performed only periodically (e.g., according to a time schedule), or it can be continuously run, such as in a background mode of operation. In one embodiment, the combined weights of all potential paths should add up to unity (e.g., when the weighting for one path is decreased, the corresponding weights that the other paths will be selected will increase).

[180] Adjustments to weight values for other paths can be prorated. For example, a decrease of 10% in weight value for one path could result in an evenly distributed increase in the weights for the remaining paths. Alternatively, weightings could be adjusted according to a weighted formula as desired (e.g., favoring healthy paths over less healthy paths). In yet another variation, the difference in weight value can be amortized over the remaining links in a manner that is proportional to their traffic weighting.

[181] FIG. 22B shows steps that can be executed to shut down transmission links where a transmitter turns off. In step 2210, a transmitter shut-down event occurs. In step 2211, a test is made to determine whether at least one transmitter is still turned on. If not, then in step 2215 all packets are dropped until a transmitter turns on. If in step 2211 at least one transmitter is turned on, then in step 2212 the weight for the path is set to zero, and the weights for the remaining paths are adjusted accordingly.

[182] FIG. 23 shows a computer node 2301 employing various principles of the above-described embodiments. It is assumed that two computer nodes of the type shown in FIG. 23 communicate over a plurality of separate physical transmission paths. As shown in FIG. 23, four transmission paths X1 through X4 are defined for communicating between the two nodes. Each node includes a packet transmitter 2302 that operates in accordance with a transmit table 2308 as described above. (The packet transmitter could also operate without using the IP-hopping features described above, but the following description assumes that some form of hopping is employed in conjunction with the path selection mechanism.). The computer node also includes a packet receiver 2303 that operates in accordance with a receive table 2309, including a moving window W that moves as valid packets are received. Invalid packets having source and destination addresses that do not fall within window W are rejected.

[183] As each packet is readied for transmission, source and destination IP addresses (or other discriminator values) are selected from transmit table 2308 according to any of the various algorithms described above, and packets containing these source/destination address pairs, which correspond to the node to which the four transmission paths are linked, are generated to a transmission path switch 2307. Switch 2307, which can comprise a software function, selects from one of the available transmission paths according to a weight distribution table 2306. For example, if the weight for path X1 is 0.2, then every fifth packet will be transmitted on path X1. A similar regime holds true for the other paths as shown. Initially, each link's weight value can be set such that it is proportional to its bandwidth, which will be referred to as its "steady-state" value.

[184] Packet receiver 2303 generates an output to a link quality measurement function 2304 that operates as described above to determine the quality of each transmission path. (The input to packet receiver 2303 for receiving incoming packets is omitted for clarity). Link quality measurement function 2304 compares the link quality to a threshold for each transmission link and, if necessary, generates an output to weight adjustment function 2305. If a weight adjustment is required, then the weights in table 2306 are adjusted accordingly, preferably according to a gradual (e.g., linearly or exponentially declining) function. In one embodiment, the weight values for all available paths are initially set to the same value, and only when paths degrade in quality are the weights changed to reflect differences.

[185] Link quality measurement function 2304 can be made to operate as part of a synchronizer function as described above. That is, if resynchronization occurs and the receiver detects that synchronization has been lost (e.g., resulting in the synchronization window W being advanced out of sequence), that fact can be used to drive link quality measurement function 2304. According to one embodiment, load balancing is performed using information garnered during the normal synchronization, augmented slightly to communicate link health from the receiver to the transmitter. The receiver maintains a count, MESS_R(W), of the messages received in synchronization window W. When it receives a synchronization request (SYNC_REQ) corresponding to the end of window W, the receiver includes counter MESS_R in the resulting synchronization acknowledgement (SYNC_ACK) sent back to the transmitter. This allows the transmitter to compare messages sent to messages received in order to asses the health of the link.

[186] If synchronization is completely lost, weight adjustment function 2305 decreases the weight value on the affected path to zero. When synchronization is regained, the weight value for the affected path is gradually increased to its original value. Alternatively, link quality can be measured by evaluating the length of time required for the receiver to acknowledge a synchronization request. In one embodiment, separate transmit and receive tables are used for each transmission path.

[187] When the transmitter receives a SYNC_ACK, the MESS_R is compared with the number of messages transmitted in a window (MESS_T). When the transmitter receives a SYNC_ACK, the traffic probabilities will be examined and adjusted if necessary. MESS_R is compared with the number of messages transmitted in a window (MESS_T). There are two possibilities:

1. If MESS_R is less than a threshold value, THRESH, then the link will be deemed to be unhealthy. If the transmitter was turned off, the transmitter is turned on and the weight P for that link will be set to a minimum value MIN. This will keep a trickle of traffic on the link for monitoring purposes until it recovers. If the transmitter was turned on, the weight P for that link will be set to:

$$P' = \alpha \times MIN + (1 - \alpha) \times P \quad (1)$$

Equation 1 will exponentially damp the traffic weight value to MIN during sustained periods of degraded service.

2. If MESS_R for a link is greater than or equal to THRESH, the link will be deemed healthy. If the weight P for that link is greater than or equal to the steady state value S for that link, then P is left unaltered. If the weight P for that link is less than THRESH then P will be set to:

$$P' = \beta \times S + (1 - \beta) \times P \qquad (2)$$

where β is a parameter such that $0 <= \beta <= 1$ that determines the damping rate of P.

[188] Equation 2 will increase the traffic weight to S during sustained periods of acceptable service in a damped exponential fashion.

[189] A detailed example will now be provided with reference to FIG. 24. As shown in FIG. 24, a first computer 2401 communicates with a second computer 2402 through two routers 2403 and 2404. Each router is coupled to the other router through three transmission links. As described above, these may be physically diverse links or logical links (including virtual private networks).

[190] Suppose that a first link L1 can sustain a transmission bandwidth of 100 Mb/s and has a window size of 32; link L2 can sustain 75 Mb/s and has a window size of 24; and link L3 can sustain 25 Mb/s and has a window size of 8. The combined links can thus sustain 200Mb/s. The steady state traffic weights are 0.5 for link L1; 0.375 for link L2, and 0.125 for link L3. MIN=1Mb/s, THRESH =0.8 MESS_T for each link, α=.75 and β=.5. These traffic weights will remain stable until a link stops for synchronization or reports a number of packets received less than its THRESH. Consider the following sequence of events:

1. Link L1 receives a SYNC_ACK containing a MESS_R of 24, indicating that only 75% of the MESS_T (32) messages transmitted in the last window were successfully received. Link 1 would be below THRESH (0.8). Consequently, link L1's traffic weight value would be reduced to 0.12825, while link L2's traffic weight value would be increased to 0.65812 and link L3's traffic weight value would be increased to 0.217938.
2. Link L2 and L3 remained healthy and link L1 stopped to synchronize. Then link L1's traffic weight value would be set to 0, link L2's traffic weight value would be set to 0.75, and link L33's traffic weight value would be set to 0.25.
3. Link L1 finally received a SYNC_ACK containing a MESS_R of 0 indicating that none of the MESS_T (32) messages transmitted in the last window were successfully received. Link L1 would be below THRESH. Link L1's traffic weight value would be

increased to .005, link L2's traffic weight value would be decreased to 0.74625, and link L3's traffic weight value would be decreased to 0.24875.

4. Link L1 received a SYNC_ACK containing a MESS_R of 32 indicating that 100% of the MESS_T (32) messages transmitted in the last window were successfully received. Link L1 would be above THRESH. Link L1's traffic weight value would be increased to 0.2525, while link L2's traffic weight value would be decreased to 0.560625 and link L3's traffic weight value would be decreased to .186875.
5. Link L1 received a SYNC_ACK containing a MESS_R of 32 indicating that 100% of the MESS_T (32) messages transmitted in the last window were successfully received. Link L1 would be above THRESH. Link L1's traffic weight value would be increased to 0.37625; link L2's traffic weight value would be decreased to 0.4678125, and link L3's traffic weight value would be decreased to 0.1559375.
6. Link L1 remains healthy and the traffic probabilities approach their steady state traffic probabilities.

B. <u>Use of a DNS Proxy to Transparently Create Virtual Private Networks</u>

[191] A second improvement concerns the automatic creation of a virtual private network (VPN) in response to a domain-name server look-up function.

[192] Conventional Domain Name Servers (DNSs) provide a look-up function that returns the IP address of a requested computer or host. For example, when a computer user types in the web name "Yahoo.com," the user's web browser transmits a request to a DNS, which converts the name into a four-part IP address that is returned to the user's browser and then used by the browser to contact the destination web site.

[193] This conventional scheme is shown in FIG. 25. A user's computer 2501 includes a client application 2504 (for example, a web browser) and an IP protocol stack 2505. When the user enters the name of a destination host, a request DNS REQ is made (through IP protocol stack 2505) to a DNS 2502 to look up the IP address associated with the name. The DNS returns the IP address DNS RESP to client application 2504, which is then able to use the IP address to communicate with the host 2503 through separate transactions such as PAGE REQ and PAGE RESP.

[194] In the conventional architecture shown in FIG. 25, nefarious listeners on the Internet could intercept the DNS REQ and DNS RESP packets and thus learn what IP addresses the user was contacting. For example, if a user wanted to set up a secure communication path with a web site having the name "Target.com," when the user's browser contacted a DNS to find the IP address for that web site, the true IP address of that web site would be revealed over the Internet as part of the DNS inquiry. This would hamper anonymous communications on the Internet.

[195] One conventional scheme that provides secure virtual private networks over the Internet provides the DNS server with the public keys of the machines that the DNS server has the addresses for. This allows hosts to retrieve automatically the public keys of a host that the host is to communicate with so that the host can set up a VPN without having the user enter the public key of the destination host. One implementation of this standard is presently being developed as part of the FreeS/WAN project(RFC 2535).

[196] The conventional scheme suffers from certain drawbacks. For example, any user can perform a DNS request. Moreover, DNS requests resolve to the same value for all users.

[197] According to certain aspects of the invention, a specialized DNS server traps DNS requests and, if the request is from a special type of user (e.g., one for which secure communication services are defined), the server does not return the true IP address of the target node, but instead automatically sets up a virtual private network between the target node and the user. The VPN is preferably implemented using the IP address "hopping" features of the basic invention described above, such that the true identity of the two nodes cannot be determined even if packets during the communication are intercepted. For DNS requests that are determined to not require secure services (e.g., an unregistered user), the DNS server transparently "passes through" the request to provide a normal look-up function and return the IP address of the target web server, provided that the requesting host has permissions to resolve unsecured sites. Different users who make an identical DNS request could be provided with different results.

[198] FIG. 26 shows a system employing various principles summarized above. A user's computer 2601 includes a conventional client (e.g., a web browser) 2605 and an IP protocol stack 2606 that preferably operates in accordance with an IP hopping function 2607 as outlined

above. A modified DNS server 2602 includes a conventional DNS server function 2609 and a DNS proxy 2610. A gatekeeper server 2603 is interposed between the modified DNS server and a secure target site 2704. An "unsecure" target site 2611 is also accessible via conventional IP protocols.

[199] According to one embodiment, DNS proxy 2610 intercepts all DNS lookup functions from client 2605 and determines whether access to a secure site has been requested. If access to a secure site has been requested (as determined, for example, by a domain name extension, or by reference to an internal table of such sites), DNS proxy 2610 determines whether the user has sufficient security privileges to access the site. If so, DNS proxy 2610 transmits a message to gatekeeper 2603 requesting that a virtual private network be created between user computer 2601 and secure target site 2604. In one embodiment, gatekeeper 2603 creates "hopblocks" to be used by computer 2601 and secure target site 2604 for secure communication. Then, gatekeeper 2603 communicates these to user computer 2601. Thereafter, DNS proxy 2610 returns to user computer 2601 the resolved address passed to it by the gatekeeper (this address could be different from the actual target computer) 2604, preferably using a secure administrative VPN. The address that is returned need not be the actual address of the destination computer.

[200] Had the user requested lookup of a non-secure web site such as site 2611, DNS proxy would merely pass through to conventional DNS server 2609 the look-up request, which would be handled in a conventional manner, returning the IP address of non-secure web site 2611. If the user had requested lookup of a secure web site but lacked credentials to create such a connection, DNS proxy 2610 would return a "host unknown" error to the user. In this manner, different users requesting access to the same DNS name could be provided with different look-up results.

[201] Gatekeeper 2603 can be implemented on a separate computer (as shown in FIG. 26) or as a function within modified DNS server 2602. In general, it is anticipated that gatekeeper 2703 facilitates the allocation and exchange of information needed to communicate securely, such as using "hopped" IP addresses. Secure hosts such as site 2604 are assumed to be equipped with a secure communication function such as an IP hopping function 2608.

[202] It will be appreciated that the functions of DNS proxy 2610 and DNS server 2609 can be combined into a single server for convenience. Moreover, although element 2602 is shown as combining the functions of two servers, the two servers can be made to operate independently.

[203] FIG. 27 shows steps that can be executed by DNS proxy server 2610 to handle requests for DNS look-up for secure hosts. In step 2701, a DNS look-up request is received for a target host. In step 2702, a check is made to determine whether access to a secure host was requested. If not, then in step 2703 the DNS request is passed to conventional DNS server 2609, which looks up the IP address of the target site and returns it to the user's application for further processing.

[204] In step 2702, if access to a secure host was requested, then in step 2704 a further check is made to determine whether the user is authorized to connect to the secure host. Such a check can be made with reference to an internally stored list of authorized IP addresses, or can be made by communicating with gatekeeper 2603 (e.g., over an "administrative" VPN that is secure). It will be appreciated that different levels of security can also be provided for different categories of hosts. For example, some sites may be designated as having a certain security level, and the security level of the user requesting access must match that security level. The user's security level can also be determined by transmitting a request message back to the user's computer requiring that it prove that it has sufficient privileges.

[205] If the user is not authorized to access the secure site, then a "host unknown" message is returned (step 2705). If the user has sufficient security privileges, then in step 2706 a secure VPN is established between the user's computer and the secure target site. As described above, this is preferably done by allocating a hopping regime that will be carried out between the user's computer and the secure target site, and is preferably performed transparently to the user (i.e., the user need not be involved in creating the secure link). As described in various embodiments of this application, any of various fields can be "hopped" (e.g., IP source/destination addresses; a field in the header; etc.) in order to communicate securely.

[206] Some or all of the security functions can be embedded in gatekeeper 2603, such that it handles all requests to connect to secure sites. In this embodiment, DNS proxy 2610

communicates with gatekeeper 2603 to determine (preferably over a secure administrative VPN) whether the user has access to a particular web site. Various scenarios for implementing these features are described by way of example below:

Scenario #1: Client has permission to access target computer, and gatekeeper has a rule to make a VPN for the client. In this scenario, the client's DNS request would be received by the DNS proxy server 2610, which would forward the request to gatekeeper 2603. The gatekeeper would establish a VPN between the client and the requested target. The gatekeeper would provide the address of the destination to the DNS proxy, which would then return the resolved name as a result. The resolved address can be transmitted back to the client in a secure administrative VPN.

Scenario #2: Client does not have permission to access target computer. In this scenario, the client's DNS request would be received by the DNS proxy server 2610, which would forward the request to gatekeeper 2603. The gatekeeper would reject the request, informing DNS proxy server 2610 that it was unable to find the target computer. The DNS proxy 2610 would then return a "host unknown" error message to the client.

Scenario #3: Client has permission to connect using a normal non-VPN link, and the gatekeeper does not have a rule to set up a VPN for the client to the target site. In this scenario, the client's DNS request is received by DNS proxy server 2610, which would check its rules and determine that no VPN is needed. Gatekeeper 2603 would then inform the DNS proxy server to forward the request to conventional DNS server 2609, which would resolve the request and return the result to the DNS proxy server and then back to the client.

Scenario #4: Client does not have permission to establish a normal/non-VPN link, and the gatekeeper does not have a rule to make a VPN for the client to the target site. In this scenario, the DNS proxy server would receive the client's DNS request and forward it to gatekeeper 2603. Gatekeeper 2603 would determine that no special VPN was needed, but that the client is not authorized to communicate with non-VPN members. The gatekeeper would reject the request, causing DNS proxy server 2610 to return an error message to the client.

C. Large Link to Small Link Bandwidth Management

[207] One feature of the basic architecture is the ability to prevent so-called "denial of service" attacks that can occur if a computer hacker floods a known Internet node with packets, thus preventing the node from communicating with other nodes. Because IP addresses or other fields are "hopped" and packets arriving with invalid addresses are quickly discarded, Internet nodes are protected against flooding targeted at a single IP address.

[208] In a system in which a computer is coupled through a link having a limited bandwidth (e.g., an edge router) to a node that can support a much higher-bandwidth link (e.g., an Internet Service Provider), a potential weakness could be exploited by a determined hacker. Referring to FIG. 28, suppose that a first host computer 2801 is communicating with a second host computer 2804 using the IP address hopping principles described above. The first host computer is coupled through an edge router 2802 to an Internet Service Provider (ISP) 2803 through a low bandwidth link (LOW BW), and is in turn coupled to second host computer 2804 through parts of the Internet through a high bandwidth link (HIGH BW). In this architecture, the ISP is able to support a high bandwidth to the internet, but a much lower bandwidth to the edge router 2802.

[209] Suppose that a computer hacker is able to transmit a large quantity of dummy packets addressed to first host computer 2801 across high bandwidth link HIGH BW. Normally, host computer 2801 would be able to quickly reject the packets since they would not fall within the acceptance window permitted by the IP address hopping scheme. However, because the packets must travel across low bandwidth link LOW BW, the packets overwhelm the lower bandwidth link before they are received by host computer 2801. Consequently, the link to host computer 2801 is effectively flooded before the packets can be discarded.

[210] According to one inventive improvement, a "link guard" function 2805 is inserted into the high-bandwidth node (e.g., ISP 2803) that quickly discards packets destined for a low-bandwidth target node if they are not valid packets. Each packet destined for a low-bandwidth node is cryptographically authenticated to determine whether it belongs to a VPN. If it is not a valid VPN packet, the packet is discarded at the high-bandwidth node. If the packet is authenticated as belonging to a VPN, the packet is passed with high preference. If the packet is a valid non-VPN packet, it is passed with a lower quality of service (e.g., lower priority).

[211] In one embodiment, the ISP distinguishes between VPN and non-VPN packets using the protocol of the packet. In the case of IPSEC [rfc 2401], the packets have IP protocols 420 and

421. In the case of the TARP VPN, the packets will have an IP protocol that is not yet defined. The ISP's link guard, 2805, maintains a table of valid VPNs which it uses to validate whether VPN packets are cryptographically valid.

[212] According to one embodiment, packets that do not fall within any hop windows used by nodes on the low-bandwidth link are rejected, or are sent with a lower quality of service. One approach for doing this is to provide a copy of the IP hopping tables used by the low-bandwidth nodes to the high-bandwidth node, such that both the high-bandwidth and low-bandwidth nodes track hopped packets (e.g., the high-bandwidth node moves its hopping window as valid packets are received). In such a scenario, the high-bandwidth node discards packets that do not fall within the hopping window before they are transmitted over the low-bandwidth link. Thus, for example, ISP 2903 maintains a copy 2910 of the receive table used by host computer 2901. Incoming packets that do not fall within this receive table are discarded. According to a different embodiment, link guard 2805 validates each VPN packet using a keyed hashed message authentication code (HMAC) [rfc 2104]. According to another embodiment, separate VPNs (using, for example, hopblocks) can be established for communicating between the low-bandwidth node and the high-bandwidth node (i.e., packets arriving at the high-bandwidth node are converted into different packets before being transmitted to the low-bandwidth node).

[213] As shown in FIG. 29, for example, suppose that a first host computer 2900 is communicating with a second host computer 2902 over the Internet, and the path includes a high bandwidth link HIGH BW to an ISP 2901 and a low bandwidth link LOW BW through an edge router 2904. In accordance with the basic architecture described above, first host computer 2900 and second host computer 2902 would exchange hopblocks (or a hopblock algorithm) and would be able to create matching transmit and receive tables 2905, 2906, 2912 and 2913. Then in accordance with the basic architecture, the two computers would transmit packets having seemingly random IP source and destination addresses, and each would move a corresponding hopping window in its receive table as valid packets were received.

[214] Suppose that a nefarious computer hacker 2903 was able to deduce that packets having a certain range of IP addresses (e.g., addresses 100 to 200 for the sake of simplicity) are being transmitted to ISP 2901, and that these packets are being forwarded over a low-bandwidth link.

Hacker computer 2903 could thus "flood" packets having addresses falling into the range 100 to 200, expecting that they would be forwarded along low bandwidth link LOW BW, thus causing the low bandwidth link to become overwhelmed. The fast packet reject mechanism in first host computer 3000 would be of little use in rejecting these packets, since the low bandwidth link was effectively jammed before the packets could be rejected. In accordance with one aspect of the improvement, however, VPN link guard 2911 would prevent the attack from impacting the performance of VPN traffic because the packets would either be rejected as invalid VPN packets or given a lower quality of service than VPN traffic over the lower bandwidth link. A denial-of-service flood attack could, however, still disrupt non-VPN traffic.

[215] According to one embodiment of the improvement, ISP 2901 maintains a separate VPN with first host computer 2900, and thus translates packets arriving at the ISP into packets having a different IP header before they are transmitted to host computer 2900. The cryptographic keys used to authenticate VPN packets at the link guard 2911 and the cryptographic keys used to encrypt and decrypt the VPN packets at host 2902 and host 2901 can be different, so that link guard 2911 does not have access to the private host data; it only has the capability to authenticate those packets.

[216] According to yet a third embodiment, the low-bandwidth node can transmit a special message to the high-bandwidth node instructing it to shut down all transmissions on a particular IP address, such that only hopped packets will pass through to the low-bandwidth node. This embodiment would prevent a hacker from flooding packets using a single IP address. According to yet a fourth embodiment, the high-bandwidth node can be configured to discard packets transmitted to the low-bandwidth node if the transmission rate exceeds a certain predetermined threshold for any given IP address; this would allow hopped packets to go through. In this respect, link guard 2911 can be used to detect that the rate of packets on a given IP address are exceeding a threshold rate; further packets addressed to that same IP address would be dropped or transmitted at a lower priority (e.g., delayed).

D. Traffic Limiter

[217] In a system in which multiple nodes are communicating using "hopping" technology, a treasonous insider could internally flood the system with packets. In order to prevent this possibility, one inventive improvement involves setting up "contracts" between nodes in the system, such that a receiver can impose a bandwidth limitation on each packet sender. One technique for doing this is to delay acceptance of a checkpoint synchronization request from a sender until a certain time period (e.g., one minute) has elapsed. Each receiver can effectively control the rate at which its hopping window moves by delaying "SYNC ACK" responses to "SYNC_REQ" messages.

[218] A simple modification to the checkpoint synchronizer will serve to protect a receiver from accidental or deliberate overload from an internally treasonous client. This modification is based on the observation that a receiver will not update its tables until a SYNC_REQ is received on hopped address CKPT_N. It is a simple matter of deferring the generation of a new CKPT_N until an appropriate interval after previous checkpoints.

[219] Suppose a receiver wished to restrict reception from a transmitter to 100 packets a second, and that checkpoint synchronization messages were triggered every 50 packets. A compliant transmitter would not issue new SYNC_REQ messages more often than every 0.5 seconds. The receiver could delay a non-compliant transmitter from synchronizing by delaying the issuance of CKPT_N for 0.5 second after the last SYNC_REQ was accepted.

[220] In general, if M receivers need to restrict N transmitters issuing new SYNC_REQ messages after every W messages to sending R messages a second in aggregate, each receiver could defer issuing a new CKPT_N until MxNxW/R seconds have elapsed since the last SYNC_REQ has been received and accepted. If the transmitter exceeds this rate between a pair of checkpoints, it will issue the new checkpoint before the receiver is ready to receive it, and the SYNC_REQ will be discarded by the receiver. After this, the transmitter will re-issue the SYNC_REQ every T1 seconds until it receives a SYNC_ACK. The receiver will eventually update CKPT_N and the SYNC_REQ will be acknowledged. If the transmission rate greatly exceeds the allowed rate, the transmitter will stop until it is compliant. If the transmitter exceeds the allowed rate by a little, it will eventually stop after several rounds of delayed synchronization until it is in compliance. Hacking the transmitter's code to not shut off only permits the transmitter to lose the acceptance window. In this case it can recover the window and proceed only after it is compliant again.

[221] Two practical issues should be considered when implementing the above scheme:

1. The receiver rate should be slightly higher than the permitted rate in order to allow for statistical fluctuations in traffic arrival times and non-uniform load balancing.

2. Since a transmitter will rightfully continue to transmit for a period after a SYNC_REQ is transmitted, the algorithm above can artificially reduce the transmitter's bandwidth. If events prevent a compliant transmitter from synchronizing for a period (e.g. the network dropping a SYNC_REQ or a SYNC_ACK) a SYNC_REQ will be accepted later than expected. After this, the transmitter will transmit fewer than expected messages before encountering the next checkpoint. The new checkpoint will not have been activated and the transmitter will have to retransmit the SYNC_REQ. This will appear to the receiver as if the transmitter is not compliant. Therefore, the next checkpoint will be accepted late from the transmitter's perspective. This has the effect of reducing the transmitter's allowed packet rate until the transmitter transmits at a packet rate below the agreed upon rate for a period of time.

[222] To guard against this, the receiver should keep track of the times that the last C SYNC_REQs were received and accepted and use the minimum of MxNxW/R seconds after the last SYNC_REQ has been received and accepted, 2xMxNxW/R seconds after next to the last SYNC_REQ has been received and accepted, CxMxNxW/R seconds after $(C-1)^{th}$ to the last SYNC_REQ has been received, as the time to activate CKPT_N. This prevents the receiver from inappropriately limiting the transmitter's packet rate if at least one out of the last C SYNC_REQs was processed on the first attempt.

[223] FIG. 30 shows a system employing the above-described principles. In FIG. 30, two computers 3000 and 3001 are assumed to be communicating over a network N in accordance with the "hopping" principles described above (e.g., hopped IP addresses, discriminator values, etc.). For the sake of simplicity, computer 3000 will be referred to as the receiving computer and computer 3001 will be referred to as the transmitting computer, although full duplex operation is of course contemplated. Moreover, although only a single transmitter is shown, multiple transmitters can transmit to receiver 3000.

[224] As described above, receiving computer 3000 maintains a receive table 3002 including a window W that defines valid IP address pairs that will be accepted when appearing in incoming

data packets. Transmitting computer 3001 maintains a transmit table 3003 from which the next IP address pairs will be selected when transmitting a packet to receiving computer 3000. (For the sake of illustration, window W is also illustrated with reference to transmit table 3003). As transmitting computer moves through its table, it will eventually generate a SYNC_REQ message as illustrated in function 3010. This is a request to receiver 3000 to synchronize the receive table 3002, from which transmitter 3001 expects a response in the form of a CKPT_N (included as part of a SYNC_ACK message). If transmitting computer 3001 transmits more messages than its allotment, it will prematurely generate the SYNC_REQ message. (If it has been altered to remove the SYNC_REQ message generation altogether, it will fall out of synchronization since receiver 3000 will quickly reject packets that fall outside of window W, and the extra packets generated by transmitter 3001 will be discarded).

[225] In accordance with the improvements described above, receiving computer 3000 performs certain steps when a SYNC_REQ message is received, as illustrated in FIG. 30. In step 3004, receiving computer 3000 receives the SYNC_REQ message. In step 3005, a check is made to determine whether the request is a duplicate. If so, it is discarded in step 3006. In step 3007, a check is made to determine whether the SYNC_REQ received from transmitter 3001 was received at a rate that exceeds the allowable rate R (i.e., the period between the time of the last SYNC_REQ message). The value R can be a constant, or it can be made to fluctuate as desired. If the rate exceeds R, then in step 3008 the next activation of the next CKPT_N hopping table entry is delayed by W/R seconds after the last SYNC_REQ has been accepted.

[226] Otherwise, if the rate has not been exceeded, then in step 3109 the next CKPT_N value is calculated and inserted into the receiver's hopping table prior to the next SYNC_REQ from the transmitter 3101. Transmitter 3101 then processes the SYNC_REQ in the normal manner.

E. Signaling Synchronizer

[227] In a system in which a large number of users communicate with a central node using secure hopping technology, a large amount of memory must be set aside for hopping tables and their supporting data structures. For example, if one million subscribers to a web site occasionally communicate with the web site, the site must maintain one million hopping tables, thus using up valuable computer resources, even though only a small percentage of the users may actually be using the system at any one time. A desirable solution would be a system that permits a certain maximum number of simultaneous links to be maintained, but which would "recognize" millions of registered users at any one time. In other words, out of a population of a million registered users, a few thousand at a time could simultaneously communicate with a central server, without requiring that the server maintain one million hopping tables of appreciable size.

[228] One solution is to partition the central node into two nodes: a signaling server that performs session initiation for user log-on and log-off (and requires only minimally sized tables), and a transport server that contains larger hopping tables for the users. The signaling server listens for the millions of known users and performs a fast-packet reject of other (bogus) packets. When a packet is received from a known user, the signaling server activates a virtual private link (VPL) between the user and the transport server, where hopping tables are allocated and maintained. When the user logs onto the signaling server, the user's computer is provided with hop tables for communicating with the transport server, thus activating the VPL. The VPLs can be torn down when they become inactive for a time period, or they can be torn down upon user log-out. Communication with the signaling server to allow user log-on and log-off can be accomplished using a specialized version of the checkpoint scheme described above.

[229] FIG. 31 shows a system employing certain of the above-described principles. In FIG. 31, a signaling server 3101 and a transport server 3102 communicate over a link. Signaling server 3101 contains a large number of small tables 3106 and 3107 that contain enough information to authenticate a communication request with one or more clients 3103 and 3104. As described in more detail below, these small tables may advantageously be constructed as a special case of the synchronizing checkpoint tables described previously. Transport server 3102, which is preferably a separate computer in communication with signaling server 3101, contains a smaller number of larger hopping tables 3108, 3109, and 3110 that can be allocated to create a VPN with one of the client computers.

[230] According to one embodiment, a client that has previously registered with the system (e.g., via a system administration function, a user registration procedure, or some other method) transmits a request for information from a computer (e.g., a web site). In one variation, the request is made using a "hopped" packet, such that signaling server 3101 will quickly reject invalid packets from unauthorized computers such as hacker computer 3105. An "administrative" VPN can be established between all of the clients and the signaling server in order to ensure that a hacker cannot flood signaling server 3101 with bogus packets. Details of this scheme are provided below.

[231] Signaling server 3101 receives the request 3111 and uses it to determine that client 3103 is a validly registered user. Next, signaling server 3101 issues a request to transport server 3102 to allocate a hopping table (or hopping algorithm or other regime) for the purpose of creating a VPN with client 3103. The allocated hopping parameters are returned to signaling server 3101 (path 3113), which then supplies the hopping parameters to client 3103 via path 3114, preferably in encrypted form.

[232] Thereafter, client 3103 communicates with transport server 3102 using the normal hopping techniques described above. It will be appreciated that although signaling server 3101 and transport server 3102 are illustrated as being two separate computers, they could of course be combined into a single computer and their functions performed on the single computer. Alternatively, it is possible to partition the functions shown in FIG. 31 differently from as shown without departing from the inventive principles.

[233] One advantage of the above-described architecture is that signaling server 3101 need only maintain a small amount of information on a large number of potential users, yet it retains the capability of quickly rejecting packets from unauthorized users such as hacker computer 3105. Larger data tables needed to perform the hopping and synchronization functions are instead maintained in a transport server 3102, and a smaller number of these tables are needed since they are only allocated for "active" links. After a VPN has become inactive for a certain time period (e.g., one hour), the VPN can be automatically torn down by transport server 3102 or signaling server 3101.

[234] A more detailed description will now be provided regarding how a special case of the checkpoint synchronization feature can be used to implement the signaling scheme described above.

[235] The signaling synchronizer may be required to support many (millions) of standing, low bandwidth connections. It therefore should minimize per-VPL memory usage while providing the security offered by hopping technology. In order to reduce memory usage in the signaling server, the data hopping tables can be completely eliminated and data can be carried as part of the SYNC_REQ message. The table used by the server side (receiver) and client side (transmitter) is shown schematically as element 3106 in FIG. 31.

[236] The meaning and behaviors of CKPT_N, CKPT_O and CKPT_R remain the same from the previous description, except that CKPT_N can receive a combined data and SYNC_REQ message or a SYNC_REQ message without the data.

[237] The protocol is a straightforward extension of the earlier synchronizer. Assume that a client transmitter is on and the tables are synchronized. The initial tables can be generated "out of band." For example, a client can log into a web server to establish an account over the Internet. The client will receive keys etc encrypted over the Internet. Meanwhile, the server will set up the signaling VPN on the signaling server.

[238] Assuming that a client application wishes to send a packet to the server on the client's standing signaling VPL:

1. The client sends the message marked as a data message on the inner header using the transmitter's CKPT_N address. It turns the transmitter off and starts a timer T1 noting CKPT_O. Messages can be one of three types: DATA, SYNC_REQ and SYNC_ACK. In the normal algorithm, some potential problems can be prevented by identifying each message type as part of the encrypted inner header field. In this algorithm, it is important to distinguish a data packet and a SYNC_REQ in the signaling synchronizer since the data and the SYNC_REQ come in on the same address.
2. When the server receives a data message on its CKPT_N, it verifies the message and passes it up the stack. The message can be verified by checking message type and and other information (i.e user credentials) contained in the inner header It replaces its CKPT_O with CKPT_N and generates the next CKPT_N. It updates its transmitter side

CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.

3. When the client side receiver receives a SYNC_ACK on its CKPT_R with a payload matching its transmitter side CKPT_O and the transmitter is off, the transmitter is turned on and the receiver side CKPT_R is updated. If the SYNC_ACK's payload does not match the transmitter side CKPT_O or the transmitter is on, the SYNC_ACK is simply discarded.
4. T1 expires: If the transmitter is off and the client's transmitter side CKPT_O matches the CKPT_O associated with the timer, it starts timer T1 noting CKPT_O again, and a SYNC_REQ is sent using the transmitter's CKPT_O address. Otherwise, no action is taken.
5. When the server receives a SYNC_REQ on its CKPT_N, it replaces its CKPT_O with CKPT_N and generates the next CKPT_N. It updates its transmitter side CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.
6. When the server receives a SYNC_REQ on its CKPT_O, it updates its transmitter side CKPT_R to correspond to the client's receiver side CKPT_R and transmits a SYNC_ACK containing CKPT_O in its payload.

[239] FIG. 32 shows message flows to highlight the protocol. Reading from top to bottom, the client sends data to the server using its transmitter side CKPT_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT_N into CKPT_O and updates CKPT_N. This message is successfully received and a passed up the stack. It also synchronizes the receiver i.e, the server loads CKPT_N into CKPT_O and generates a new CKPT_N, it generates a new CKPT_R in the server side transmitter and transmits a SYNC_ACK containing the server side receiver's CKPT_O the server. The SYNC_ACK is successfully received at the client. The client side receiver's CKPT_R is updated, the transmitter is turned on and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

[240] Next, the client sends data to the server using its transmitter side CKPT_N. The client side transmitter is turned off and a retry timer is turned off. The transmitter will not transmit messages as long as the transmitter is turned off. The client side transmitter then loads CKPT_N into CKPT_O and updates CKPT_N. This message is lost. The client side timer expires and as a result a SYNC_REQ is transmitted on the client side transmitter's CKPT_O (this will keep happening until the SYNC_ACK has been received at the client). The SYNC_REQ is successfully received at the server. It synchronizes the receiver i.e, the server loads CKPT_N into CKPT_O and generates a new CKPT_N, it generates an new CKPT_R in the server side transmitter and transmits a SYNC_ACK containing the server side receiver's CKPT_O the server. The SYNC_ACK is successfully received at the client. The client side receiver's CKPT_R is updated, the transmitter is turned off and the retry timer is killed. The client side transmitter is ready to transmit a new data message.

[241] There are numerous other scenarios that follow this flow. For example, the SYNC_ACK could be lost. The transmitter would continue to re-send the SYNC_REQ until the receiver synchronizes and responds.

[242] The above-described procedures allow a client to be authenticated at signaling server 3201 while maintaining the ability of signaling server 3201 to quickly reject invalid packets, such as might be generated by hacker computer 3205. In various embodiments, the signaling synchronizer is really a derivative of the synchronizer. It provides the same protection as the hopping protocol, and it does so for a large number of low bandwidth connections.

CLAIMS

We Claim:

1. A method for establishing an encrypted channel between a client and a target computer, comprising the steps of:

(i) intercepting a DNS request sent by the client; and

(ii) based on the DNS request, establishing the encrypted channel between the client and the target.

2. The method of claim 1, wherein step (ii) comprises steps of:

a) determining whether the client is authorized to access the target;

b) when the client is authorized to access the target, initiating the encrypted channel; and

c) when the client is not authorized to access the target, sending an error message to the client.

3. The method of claim 2, wherein step b) comprises sending encrypted channel parameters to the client.

4. The method of claim 1, wherein step (ii) occurs in a communication protocol independently of an application program.

5. The method of claim 1, wherein step (i) comprises a DNS proxy server intercepting the DNS request sent by the client.

6. The method of claim 1, wherein step (ii) comprises establishing the encrypted channel responsive to intercepting a DNS request for a domain name comprising a predetermined domain name extension.

7. A method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name extension.

8. The method of claim 7, wherein the creating step is performed in a communication protocol independently of an application program.

9. A method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel in response to detecting a request for access to a predetermined IP address.

10. The method of claim 9, wherein the creating step is performed in a communication protocol independently of an application program.

11. A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:

(i) determining whether the intercepted DNS request corresponds to a secure server;

(ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

12. The data processing device of claim 11, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

13. The data processing device of claim 12, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

14. The data processing device of claim 13, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

15. A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

16. A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) intercepting a DNS request sent by a client;

(ii) determining whether the intercepted DNS request corresponds to a secure server;

(iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

17. The computer readable medium of claim 16, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

18. The computer readable medium of claim 17, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

19. The computer readable medium of claim 18, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

20. A computer readable medium comprising computer readable instructions that, when executed, cause a domain name server (DNS) proxy module to intercept DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

ABSTRACT

A plurality of computer nodes communicate using seemingly random Internet Protocol source and destination addresses. Data packets matching criteria defined by a moving window of valid addresses are accepted for further processing, while those that do not meet the criteria are quickly rejected. Improvements to the basic design include (1) a load balancer that distributes packets across different transmission paths according to transmission path quality; (2) a DNS proxy server that transparently creates a virtual private network in response to a domain name inquiry; (3) a large-to-small link bandwidth management feature that prevents denial-of-service attacks at system chokepoints; (4) a traffic limiter that regulates incoming packets by limiting the rate at which a transmitter can be synchronized with a receiver; and (5) a signaling synchronizer that allows a large number of nodes to communicate with a central node by partitioning the communication function between two separate entities.

Attorney Docket No. 00479.85672

JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

■ is attached hereto.

☐ was filed on ______ as Application Serial Number ________ and was amended on ________ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed Under 35 U.S.C. 119	Certified Copy Attached
				Yes / No	Yes / No

Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s)	Filing Date (MM/DD/YYYY)	☐ Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

Attorney Docket No. 00479.85672

We hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial Number	Date of Filing (Day, Month, Year)	Status Patented, Pending, Abandoned
09/429,643	10/29/99	Pending

Power of Attorney

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Altherr, Reg. No. 31,810, Donald W. Banner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.
Eleventh Floor
1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature________________ Date________________

Full Name of Joint Inventor: MUNGER (Family Name) | Edmund (First Given Name) | Colby (Second Given Name)

Residence: 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship: U.S.

Post Office Address: 1101 Opaca Court, Crownsville, Maryland 21032

Attorney Docket No. 00479.85672

Signature________________ Date________________

Full Name of
Joint Inventor SCHMIDT | Douglas | Charles
Family Name | First Given Name | Second Given Name

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.
Post Office
Address 230 Oak Court, Severna Park, Maryland 21146

Signature Robert D. Short Date 2/14/00

Full Name of
Joint Inventor SHORT | Robert | Dunham, III
Family Name | First Given Name | Second Given Name

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.
Post Office
Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature Victor Larson Date 2/14/2000

Full Name of
Joint Inventor LARSON | Victor
Family Name | First Given Name | Second Given Name

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.
Post Office
Address 12026 Lisa Marie Court, Fairfax, Virginia 22033

Attorney Docket No. 00479.85672

Signature ___[signature]___ Date 2/14/2000

Full Name of
Joint Inventor WILLIAMSON Michael

Family Name	First Given Name	Second Given Name
WILLIAMSON	Michael	

Residence 26203 Ocala Circle, South Riding, Virginia 20152

Citizenship U.S.

Post Office
Address 26203 Ocala Circle, South Riding, Virginia 20152

LAW OFFICES
BANNER & WITCOFF, LTD.
1001 G STREET, N.W.
WASHINGTON, D.C. 20001-4597
(202) 508-9100

Attorney Docket No. 00479.85672

JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

■ is attached hereto.

☐ was filed on ______ as Application Serial Number ________ and was amended on ________ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application Number	Date of Filing (day, month, year)	Date of Issue (day, month, year)	Priority Claimed Under 35 U.S.C. 119	Certified Copy Attached
				Yes / No	Yes / No

Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

Application Number(s)	Filing Date (MM/DD/YYYY)	☐ Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

We hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial Number	Date of Filing (Day, Month, Year)	Status Patented, Pending, Abandoned
09/429,643	10/29/99	Pending

Power of Attorney

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Altherr, Reg. No. 31,810, Donald W. Banner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.
Eleventh Floor
1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature__________ Date__________

Full Name of Joint Inventor: MUNGER (Family Name) Edmund (First Given Name) Colby (Second Given Name)

Residence: 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship: U.S.

Post Office Address: 1101 Opaca Court, Crownsville, Maryland 21032

Signature [signature] Date 2/14/00

Full Name of Joint Inventor SCHMIDT Douglas Charles

Family Name First Given Name Second Given Name

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.

Post Office Address 230 Oak Court, Severna Park, Maryland 21146

Signature___ Date___

Full Name of Joint Inventor SHORT Robert Dunham, III

Family Name First Given Name Second Given Name

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.

Post Office Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature___ Date___

Full Name of Joint Inventor LARSON Victor

Family Name First Given Name Second Given Name

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.

Post Office Address 12026 Lisa Marie Court, Fairfax, Virginia 22033

Attorney Docket No: 00479.85672

Signature______________________________ Date______________________

Full Name of
Joint Inventor WILLIAMSON Michael

Family Name	First Given Name	Second Given Name
WILLIAMSON	Michael	

Residence 26203 Ocala Circle, South Riding, Virginia 20152

Citizenship U.S.

Post Office
Address 26203 Ocala Circle, South Riding, Virginia 20152

LAW OFFICES
BANNER & WITCOFF, LTD.
1001 G STREET, N.W.
WASHINGTON, D.C. 20001-4597
(202) 508-9100

Attorney Docket No. 00479.85672

JOINT DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names:

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled:

IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

the specification of which

■ is attached hereto.

☐ was filed on ______ as Application Serial Number ________ and was amended on ________ (if applicable).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56.

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
				Yes / No	Yes / No

Prior United States Application(s)

We hereby claim the benefit under 35 U.S.C. 119(e) of any United States provisional application(s) listed below:

[illegible]	[illegible]	☐ Additional provisional application numbers are listed on a supplemental priority data sheet PTO/SB/02B attached hereto.
60/106,261	10/30/98	
60/137,704	6/7/99	

Attorney Docket No. 00479.85672

We hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, We acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT International filing date of this application:

[illegible]	[illegible]	[illegible]
09/429,643	10/29/99	Pending

Power of Attorney

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and transact all business in the U.S. Patent and Trademark Office connected herewith as well as before any office or agency of a foreign country or any international organization in connection with any foreign counterpart application claiming priority to this application, including the power to appoint agents and local representatives in connection with such foreign applications, the following attorneys of Banner & Witcoff, their registration numbers being listed after their names:

Robert Altherr, Reg. No. 31,810, Donald W. Banner, Reg. No. 17,037; Edward F. McKie, Jr., Reg. No. 17,335; William W. Beckett, Reg. No. 18,262; Dale H. Hoscheit, Reg. No. 19,090; Joseph M. Potenza, Reg. No. 28,175; James A. Niegowski, Reg. No. 28,331; Joseph M. Skerpon, Reg. No. 29,864; Thomas L. Peterson, Reg. No. 30,969; Nina L. Medlock, Reg. No. 29,673; William J. Fisher, Reg. No. 32,133; Thomas H. Jackson, Reg. No. 29,808; Franklin D. Wolffe, Reg. No. 19,724; Susan A. Wolffe, Reg. No. 33,568; Daniel E. Fisher, Reg. No. 34,162; Kevin A. Wolff, Reg. No. 42,233 and Bradley C. Wright, Reg. No. 38,061.

All correspondence and telephone communications should be addressed to:

Banner & Witcoff, Ltd.
Eleventh Floor
1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel. No. (202) 508-9100

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. 1001 and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature Edmund Colby Munger Date 15 FEB 2000

Full Name of Joint Inventor MUNGER (Family Name) Edmund (First Given Name) Colby (Second Given Name)

Residence 1101 Opaca Court, Crownsville, Maryland 21032

Citizenship U.S.

Post Office Address 1101 Opaca Court, Crownsville, Maryland 21032

Attorney Docket No. 00479.85672

Signature________________ Date________________

Full Name of Joint Inventor SCHMIDT (Family Name) Douglas (First Given Name) Charles (Second Given Name)

Residence 230 Oak Court, Severna Park, Maryland 21146

Citizenship U.S.

Post Office Address 230 Oak Court, Severna Park, Maryland 21146

Signature________________ Date________________

Full Name of Joint Inventor SHORT (Family Name) Robert (First Given Name) Dunham, III (Second Given Name)

Residence 38710 Goose Creek Lane, Leesburg, Virginia 20175

Citizenship U.S.

Post Office Address 38710 Goose Creek Lane, Leesburg, Virginia 20175

Signature________________ Date________________

Full Name of Joint Inventor LARSON (Family Name) Victor (First Given Name) ________ (Second Given Name)

Residence 12026 Lisa Marie Court, Fairfax, Virginia 22033

Citizenship U.S.

Post Office Address 12026 Lisa Marie Court, Fairfax, Virginia 22033

Attorney Docket No. 00479.85672

Signature________________________________ Date________________

Full Name of Joint Inventor WILLIAMSON (Family Name) Michael (First Given Name) ________ (Second Given Name)

Residence 26203 Ocala Circle, South Riding, Virginia 20152

Citizenship U.S.

Post Office Address 26203 Ocala Circle, South Riding, Virginia 20152

LAW OFFICES
BANNER & WITCOFF, LTD.
1001 G STREET, N.W.
WASHINGTON, D.C. 20001-4597
(202) 508-9100

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 5257

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/259,494	09/30/2002 RULE	709	2143	000479.00082

APPLICANTS

Edmund Colby Munger, Crownsville, MD;
Robert Dunham Short III, Leesburg, VA;
Victor Larson, Fairfax, VA;
Michael Williamson, South Riding, VA;

** **CONTINUING DATA** **************************

This application is a DIV of 09/504,783 02/15/2000 PAT 6,502,135
which is a CIP of 09/429,643 10/29/1999
which claims benefit of 60/106,261 10/30/1998
and claims benefit of 60/137,704 06/07/1999

now. U.S. Patent No 6,502,135

** **FOREIGN APPLICATIONS** ********************

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 10/31/2002

Foreign Priority claimed / 35 USC 119 (a-d) conditions met / Verified and Acknowledged	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ☐ yes ☒ no 35 USC 119 (a-d) conditions met ☐ yes ☒ no ☐ Met after Allowance Verified and Acknowledged [signature] Examiner's Signature [initials] Initials	MD	35	20	6

ADDRESS

22907

TITLE

Agile network protocol for secure communications with assured system availability

FILING FEE RECEIVED	FEES	
992	FEES: Authority has been given in Paper No. ________ to charge/credit DEPOSIT ACCOUNT No. ________ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other ____________ ☐ Credit

PATENT APPLICATION SERIAL NO. ____________________

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

10/02/2002 GGEBREGI 00000072 190733 10259494

01 FC:101 740.00 CH
02 FC:102 252.00 CH

PTO-1556
(5/87)

*U.S. Government Printing Office: 2002 — 489-267/69033

BEST AVAILABLE COPY

MULTIPLE DEPEN[illegible] [illegible]LAIM FEE CALCULATION S[illegible]EET
(FOR USE WITH FORM PTO-875)

SERIAL NO. | FILING DATE

APPLICANT(S) 10/259454

CLAIMS

	AS FILED IND.	AS FILED DEP.	AFTER 1st AMENDMENT IND.	AFTER 1st AMENDMENT DEP.	AFTER 2nd AMENDMENT IND.	AFTER 2nd AMENDMENT DEP.
1	/					
2						
3						
4						
5						
6						
7	/					
8						
9	/					
10						
11	/					
12						
13						
14						
15	/					
16	/					
17						
18						
19						
20	/					
21						
22						
23						
24						
25						
26						
27						
28						
29						
30						
31						
32						
33						
34						
35						
36						
37						
38						
39						
40						
41						
42						
43						
44						
45						
46						
47						
48						
49						
50						
TOTAL IND.	7	↲		↲		↲
TOTAL DEP.	13					
TOTAL CLAIMS	20					

	* IND.	* DEP.	* IND.	* DEP.	* IND.	* DEP.
51						
52						
53						
54						
55	/					
56						
57						
58						
59						
60						
61						
62						
63						
64						
65						
66						
67						
68						
69						
70						
71						
72						
73						
74						
75						
76						
77						
78						
79						
80						
81						
82						
83						
84						
85						
86						
87						
88						
89						
90						
91						
92						
93						
94						
95						
96						
97						
98						
99						
100						
TOTAL IND.		↲		↲		↲
TOTAL DEP.						
TOTAL CLAIMS						

PTO-1360 (5-78)

*MAY BE USED FOR ADDITIONAL CLAIMS OR AMENDMENTS

U.S. DEPARTMENT of COMMERCE Patent and Trademark Office

Gaines

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#2

In re Application of Edmond Colby Munger et al. Serial No.: TBA DIV of 09/504,783 Filed: September 30, 2002 For: IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY	Group Art Unit: 2153 Examiner: TBA Atty. Docket No.: 000479.00082

INFORMATION DISCLOSURE STATEMENT

Honorable Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Pursuant to 37 C.F.R. 1.56, the attention of the Patent and Trademark Office is hereby directed to the reference(s) listed on the attached PTO-1449. A copy of each cited prior art reference was provided or cited in the prior application serial number 09/504,783 in accordance with 37 C.F.R. 1.98(d). It is respectfully requested that the information be expressly considered during the prosecution of this application, and that the reference(s) be made of record therein and appear among the "References Cited" on any patent to issue therefrom.

Applicant does not waive any right to take appropriate action to establish patentability over the listed documents should they be applied as a reference against the claims of the present application.

The accompanying Information Disclosure Statement is being filed within three months of the U.S. filing date OR before the mailing date of a first Office Action on the merits. No certification or fee is required.

Respectfully submitted,
BANNER & WITCOFF, LTD.

By: [signature]
Ross A. Dannenberg
Registration No. 49,024

1001 G Street, N.W.
Eleventh Floor
Washington, D.C. 20001-4597
(202) 508-9100

Dated: September 30, 2002

Application Data Sheet

Application Information

Application number::	
Filing Date::	
Application Type::	Regular
Subject Matter::	Utility
Suggested classification::	
Suggested Group Art Unit::	
CD-ROM or CD-R?::	None
Number of CD disks::	
Number of copies of CDs::	
Sequence submission?::	
Computer Readable Form (CRF)?::	
Number of copies of CRF::	
Title::	IMPROVEMENTS TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY
Attorney Docket Number::	000479.00082
Request for Early Publication?::	NO
Request for Non-Publication?::	NO
Suggested Drawing Figure::	
Total Drawing Sheets::	35
Small Entity?::	NO
Latin name::	
Variety denomination name::	
Petition included?::	NO
Petition Type::	
Licensed US Govt. Agency::	
Contract or Grant Numbers::	

Secrecy Order in Parent Appl.?:: NO

Applicant Information

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Edward
Middle Name::	Colby
Family Name::	Munger
Name Suffix::	
City of Residence::	Crownsville
State or Province of Residence::	MD
Country of Residence::	USA
Street of mailing address::	1101 Opaca Court
City of mailing address::	Crownsville
State or Province of mailing address::	MD
Country of mailing address::	USA
Postal or Zip Code of mailing address::	21032

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Douglas
Middle Name::	Charles
Family Name::	Schmidt
Name Suffix::	
City of Residence::	Severna Park
State or Province of Residence::	MD
Country of Residence::	USA
Street of mailing address::	230 Oak Court

City of mailing address::	Serverna Park
State or Province of mailing address::	MD
Country of mailing address::	USA
Postal or Zip Code of mailing address::	21146

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Robert
Middle Name::	Dunham
Family Name::	Short
Name Suffix::	III
City of Residence::	Leesburg
State or Province of Residence::	VA
Country of Residence::	USA
Street of mailing address::	38710 Goose Creek Lane
City of mailing address::	Leesburg
State or Province of mailing address::	VA
Country of mailing address::	USA
Postal or Zip Code of mailing address::	20175

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Victor
Middle Name::	
Family Name::	Larson
Name Suffix::	
City of Residence::	Fairfax
State or Province of Residence::	VA

Country of Residence::	USA
Street of mailing address::	12026 Lisa Marie Court
City of mailing address::	Fairfax
State or Province of mailing address::	VA
Country of mailing address::	USA
Postal or Zip Code of mailing address::	22033

Applicant Authority Type::	Inventor
Primary Citizenship Country::	USA
Status::	Full Capacity
Given Name::	Michael
Middle Name::	
Family Name::	Williamson
Name Suffix::	
City of Residence::	South Riding
State or Province of Residence::	VA
Country of Residence::	USA
Street of mailing address::	26203 Ocala Circle
City of mailing address::	South Riding
State or Province of mailing address::	VA
Country of mailing address::	USA
Postal or Zip Code of mailing address::	20152

Correspondence Information

Correspondence Customer Number:: 22907

Representative Information

Representative Customer Number:: 22907

Domestic Priority Information

Application::	Continuity Type::	Parent Application::	Parent Filing Date::
This Application	Division of	09/504,783	02/15/00

Foreign Priority Information

Country::	Application number::	Filing Date::	Priority Claimed::

Assignee Information

Assignee name::	Science Applications International Corporation
Street of mailing address::	10260 Campus Point Drive
City of mailing address::	San Diego
State or Province of mailing address::	CA
Country of mailing address::	USA
Postal or Zip Code of mailing address::	92121

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 838 930 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
29.04.1998 Bulletin 1998/18

(51) Int. Cl.[6]: **H04L 29/06**

(21) Application number: **97118556.6**

(22) Date of filing: **24.10.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV RO SI

(30) Priority: **25.10.1996 US 738155**

(71) Applicant:
DIGITAL EQUIPMENT CORPORATION
Maynard, Massachusetts 01754 (US)

(72) Inventors:
- **Alden, Kenneth F.**
  Boylston, Massachusetts 01505 (US)
- **Lichtenberg, Mitchell P.**
  Sunnyvale, CA 94087 (US)
- **Wobber, Edward P.**
  Menlo Park, California 94025 (US)

(74) Representative: **Betten & Resch**
Reichenbachstrasse 19
80469 München (DE)

(54) Pseudo network adapter for frame capture, encapsulation and encryption

(57) A new pseudo network adapter provides an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network, and includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. A transmit path in the system processes data packets from the local communications protocol stack for transmission through the pseudo network adapter. An encryption engine encrypts the data packets and an encapsulation engine encapsulates the encrypted data packets into tunnel data frames. The network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for delivery to a user.

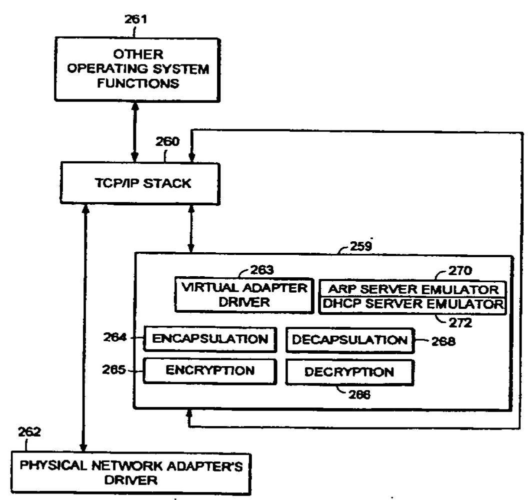


FIG. 15

EP 0 838 930 A2

Description

FIELD OF THE INVENTION

The invention relates generally to establishing secure virtual private networks. The invention relates specifically to a pseudo network adapter for capturing, encapsulating and encrypting messages or frames.

BACKGROUND

In data communications it is often required that secure communications be provided between users of network stations (also referred to as "network nodes") at different physical locations. Secure communications must potentially extend over public networks as well as through secure private networks. Secure private networks are protected by "firewalls", which separate the private network from a public network. Firewalls ordinarily provide some combination of packet filtering, circuit gateway, and application gateway technology, insulating the private network from unwanted communications with the public network.

One approach to providing secure communications is to form a virtual private network. In a virtual private network, secure communications are provided by encapsulating and encrypting messages. Encapsulated messaging in general is referred to as "tunneling". Tunnels using encryption may provide protected communications between users separated by a public network, or among a subset of users of a private network.

Encryption may for example be performed using an encryption algorithm using one or more encryption "keys". When an encryption key is used, the value of the key determines how the data is encrypted and decrypted. When a public-key encryption system is used, a key pair is associated with each communicating entity. The key pair consists of an encryption key and a decryption key. The two keys are formed such that it is unfeasible to generate one key from the other. Each entity makes its encryption key public, while keeping its decryption key secret. When sending a message to node A, for example, the transmitting entity uses the public key of node A to encrypt the message, and then the message can only be decrypted by node A using node A's private key.

In a symmetric key encryption system a single key is used as the basis for both encryption and decryption. An encryption key in a symmetric key encryption system is sometimes referred to as a "shared" key. For example, a pair of communicating nodes A and B could communicate securely as follows: a first shared key is used to encrypt data sent from node A to node B, while a second shared key is to be used to encrypt data sent from node B to node A. In such a system, the two shared keys must be known by both node A and node B. More examples of encryption algorithms and keyed encryption are disclosed in many textbooks, for example "Applied Cryptography - Protocols, Algorithms, and Source Code in C", by Bruce Schneier, published by John Wiley and Sons, New York, New York, copyright 1994.

Information regarding what encryption key or keys are to be used, and how they are to be used to encrypt data for a given secure communications session is referred to as "key exchange material". Key exchange material may for example determine what keys are used and a time duration for which each key is valid. Key exchange material for a pair of communicating stations must be known by both stations before encrypted data can be exchanged in a secure communications session. How key exchange material is made known to the communicating stations for a given secure communications session is referred to as "session key establishment".

A tunnel may be implemented using a virtual or "pseudo" network adapter that appears to the communications protocol stack as a physical device and which provides a virtual private network. A pseudo network adapter must have the capability to receive packets from the communications protocol stack, and to pass received packets back through the protocol stack either to a user or to be transmitted.

A tunnel endpoint is the point at which any encryption/decryption and encapsulation/decapsulation provided by a tunnel is performed. In existing systems, the tunnel end points are pre-determined network layer addresses. The source network layer address in a received message is used to determine the "credentials" of an entity requesting establishment of a tunnel connection. For example, a tunnel server uses the source network layer address to determine whether a requested tunnel connection is authorized. The source network layer address is also used to determine which cryptographic key or keys to use to decrypt received messages.

Existing tunneling technology is typically performed by encapsulating encrypted network layer packets (also referred to as "frames") at the network layer. Such systems provide "network layer within network layer" encapsulation of encrypted messages. Tunnels in existing systems are typically between firewall nodes which have statically allocated IP addresses. In such existing systems, the statically allocated IP address of the firewall is the address of a tunnel end point within the firewall. Existing systems fail to provide a tunnel which can perform authorization based for an entity which must dynamically allocate its network layer address. This is especially problematic for a user wishing to establish a tunnel in a mobile computing environment, and who requests a dynamically allocated IP address from an Internet Service Provider (ISP).

Because existing virtual private networks are based on network layer within network layer encapsulation, they are generally only capable of providing connectionless datagram type services. Because datagram type services do not guarantee delivery of packets, existing tunnels can only easily employ encryption methods over the data contained within each transmitted packet. Encryption based on the contents of multiple packets is desirable, such as cipher block chaining or stream ciphering over multiple packets. For example, encrypted data would advantageously be formed based not only on the contents of the present packet data being encrypted, but also based on some attribute of the connection or session history between the communicating stations. Examples of encryption algorithms and keyed encryption are disclosed in many textbooks, for example "Applied Cryptography - Protocols, Algorithms, and Source Code in C", by Bruce Schneier, published by John Wiley and Sons, New York, New York, copyright 1994.

Thus there is required a new pseudo network adapter providing a virtual private network having a dynamically determined end point to support a user in a mobile computing environment. The new pseudo network adapter should appear to the communications protocol stack of the node as an interface to an actual physical device. The new pseudo network adapter should support guaranteed, in-order delivery of frames over a tunnel to conveniently support cipher block chaining mode or stream cipher encryption over multiple packets.

SUMMARY OF THE INVENTION

A new pseudo network adapter is disclosed providing a virtual private network. The new system includes an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network. The interface appears to the local communications stack as a network adapter device driver for a network adapter.

The invention, in its broad form, includes a pseudo network adapter as recited in claim 1, providing a virtual network and a method therefor as recited in claim 9.

The system as described hereinafter further includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. The new pseudo network adapter includes a transmit path for processing data packets from the local communications protocol stack for transmission through the pseudo network adapter. The transmit path includes an encryption engine for encrypting the data packets and an encapsulation engine for encapsulating the encrypted data packets into tunnel data frames. The pseudo network adapter passes the tunnel data frames back to the local communications protocol stack for transmission to a physical network adapter on a remote server node.

Preferably, as described hereinafter, the pseudo network adapter includes a digest value in a digest field in each of the tunnel data frames. A keyed hash function is a hash function which takes data and a shared cryptographic key as inputs, and outputs a digital signature referred to as a digest. The value of the digest field is equal to an output of a keyed hash function applied to data consisting of the data packet encapsulated within the tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to the remote server node. In another aspect of the system, the pseudo network adapter processes an Ethernet header in each one of the captured data packets, including removing the Ethernet header.

The new pseudo network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for delivery to a user.

Thus there is disclosed a new pseudo network adapter providing a virtual private network having dynamically determined end points to support users in a mobile computing environment. The new pseudo network adapter provides a system for capturing a fully formed frame prior to transmission. The new pseudo network adapter appears to the communications protocol stack of the station as an interface to an actual physical device. The new pseudo network adapter further includes encryption capabilities to conveniently provide secure communications between tunnel end points using stream mode encryption or cipher block chaining over multiple packets.

BRIEF DESCRIPTION OF THE DRAWINGS

A more detailed understanding of the invention may be had from the following description of a preferred embodiment, given by way of example and to be understood in conjunction with the accompanying drawing in which:

- Fig. 1 is a block diagram showing the Open Systems Interconnection (OSI) reference model;
- Fig. 2 is a block diagram showing the TCP/IP internet protocol suite;
- Fig. 3 is a block diagram showing an examplary embodiment of a tunnel connection across a public network between two tunnel servers;
- Fig. 4 is a flow chart showing an examplary embodiment of steps performed to establish a tunnel con-

nection;

- Fig. 5 is a flow chart showing an examplary embodiment of steps performed to perform session key management for a tunnel connection;
- Fig. 6 is a block diagram showing an examplary embodiment of a relay frame;
- Fig. 7 is a block diagram showing an examplary embodiment of a connection request frame;
- Fig. 8 is a block diagram showing an examplary embodiment of a connection response frame;
- Fig. 9 is a block diagram showing an examplary embodiment of a data frame;
- Fig. 10 is a block diagram showing an examplary embodiment of a close connection frame;
- Fig. 11 is a state diagram showing an examplary embodiment of a state machine forming a tunnel connection in a network node initiating a tunnel connection;
- Fig. 12 is a state diagram showing an examplary embodiment of a state machine forming a tunnel connection in a server computer;
- Fig. 13 is a state diagram showing an examplary embodiment of a state machine forming a tunnel connection in a relay node;
- Fig. 14 is a block diagram showing an examplary embodiment of a tunnel connection between a client computer (tunnel client) and a server computer (tunnel server);
- Fig. 15 is a block diagram showing an examplary embodiment of a pseudo network adapter;
- Fig. 16 is a block diagram showing an examplary embodiment of a pseudo network adapter;
- Fig. 17 is a flow chart showing steps performed by an examplary embodiment of a pseudo network adapter during packet transmission;
- Fig. 18 is a flow chart showing steps performed by an examplary embodiment of a pseudo network adapter during packet receipt;
- Fig. 19 is a data flow diagram showing data flow in an examplary embodiment of a pseudo network adapter during packet transmission;
- Fig. 20 is a data flow diagram showing data flow in an examplary embodiment of a pseudo network adapter during packet receipt;
- Fig. 21 is a diagram showing the movement of encrypted and unencrypted data in an examplary embodiment of a system including a pseudo network adapter;
- Fig. 22 is a diagram showing the movement of encrypted and unencrypted data in an examplary embodiment of a system including a pseudo network adapter; and
- Fig. 23 is a flow chart showing steps initialization of an examplary embodiment of a system including a pseudo network adapter.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Now with reference to Fig. 1 there is described for purposes of explanation, communications based on the Open Systems Interconnection (OSI) reference model. In Fig. 1 there is shown communications 12 between a first protocol stack 10 and a second protocol stack 14. The first protocol stack 10 and second protocol stack 14 are implementations of the seven protocol layers (Application layer, Presentation layer, Session layer, Transport layer, Network layer, Data link layer, and Physical layer) of the OSI reference model. A protocol stack implementation is typically in some combination of software and hardware. Descriptions of the specific services provided by each protocol layer in the OSI reference model are found in many text books, for example "Computer Networks", Second Edition, by Andrew S. Tannenbaum, published by Prentice-Hall, Englewood Cliffs, New Jersey, copyright 1988.

As shown in Fig. 1, data 11 to be transmitted from a sending process 13 to a receiving process 15 is passed down through the protocol stack 10 of the sending process to the physical layer 9 for transmission on the data path 7 to the receiving process 15. As the data 11 is passed down through the protocol stack 10, each protocol layer prepends a header (and possibly also appends a trailer) portion to convey information used by that protocol layer. For example, the data link layer 16 of the sending process wraps the information received from the network layer 17 in a data link header 18 and a data link layer trailer 20 before the message is passed to the physical layer 9 for transmission on the actual transmission path 7.

Fig. 2 shows the TCP/IP protocol stack. Some protocol layers in the TCP/IP protocol stack correspond with layers in the OSI protocol stack shown in Fig. 1. The detailed services and header formats of each layer in the TCP/IP protocol stack are described in many texts, for example "Internetworking with TCP/IP, Vol. 1: Principles, Protocols, and Architecture", Second Edition, by Douglas E. Comer, published by Prentice-Hall, Englewood Cliffs, New Jersey, copyright 1991. The Transport Control Protocol (TCP) 22 corresponds to the Transport layer in the OSI reference model. The TCP protocol 22 provides a connection-oriented, end to end transport service with guaranteed, in-sequence packet delivery. In this way the TCP protocol 22 provides a reliable, transport layer connection.

The IP protocol 26 corresponds to the Network layer of the OSI reference model. The IP protocol 26 provides no guarantee of packet delivery to the upper layers. The hardware link level and access protocols 32 correspond to the Data link and Physical layers of the OSI reference model.

The Address Resolution Protocol (ARP) 28 is used to map IP layer addresses (referred to as "IP addresses") to addresses used by the hardware link level and access protocols 32 (referred to as "physical addresses" or "MAC addresses"). The ARP protocol layer in each network station typically contains a table of mappings between IP addresses and physical addresses (referred to as the "ARP cache"). When a mapping between an IP address and the corresponding physical address is not known, the ARP protocol 28 issues a broadcast packet (an "ARP request" packet) on the local network. The ARP request indicates an IP address for which a physical address is being requested. The ARP protocols 28 in each station connected to the local network examine the ARP request, and if a station recognizes the IP address indicated by the ARP request, it issues a response (an "ARP response" or "ARP reply" packet) to the requesting station indicating the responder's physical address. The requesting ARP protocol reports the received physical address to the local IP layer which then uses it to send datagrams directly to the responding station. As an alternative to having each station respond only for its own IP address, an ARP server may be used to respond for a set of IP addresses it stores internally, thus potentially eliminating the requirement of a broadcast request. In that case, the ARP request can be sent directly to the ARP server for physical addresses corresponding to any IP address mappings stored within the ARP server.

At system start up, each station on a network must determine an IP address for each of its network interfaces before it can communicate using TCP/IP. For example, a station may need to contact a server to dynamically obtain an IP address for one or more of its network interfaces. The station may use what is referred to as the Dynamic Host Configuration Protocol (DHCP) to issue a request for an IP address to a DHCP server. For example, a DHCP module broadcasts a DHCP request packet at system start up requesting allocation of an IP address for an indicated network interface. Upon receiving the DHCP request packet, the DHCP server allocates an IP address to the requesting station for use with the indicated network interface. The requesting station then stores the IP address in the response from the server as the IP address to associate with that network interface when communicating using TCP/IP.

Fig. 3 shows an example configuration of network nodes for which the presently disclosed system is applicable. In the example of Fig. 3, the tunnel server A is an initiator of the tunnel connection. As shown in Fig. 3, the term "tunnel relay" node is used to refer to a station which forwards data packets between transport layer connections (for example TCP connections).

For example, in the present system a tunnel relay may be dynamically configured to forward packets between transport layer connection 1 and transport layer connection 2. The tunnel relay replaces the header information of packets received over transport layer connection 1 with header information indicating transport layer connection 2. The tunnel relay can then forward the packet to a firewall, which may be conveniently programmed to pass packets received over transport layer connection 2 into a private network on the other side of the firewall. In the present system, the tunnel relay dynamically forms transport layer connections when a tunnel connection is established. Accordingly the tunnel relay is capable of performing dynamic load balancing or providing redundant service for fault tolerance over one or more tunnel servers at the time the tunnel connection is established.

Fig. 3 shows a Tunnel Server A 46 in a private network N1 48, physically connected with a first Firewall 50. The first Firewall 50 separates the private network N1 48 from a public network 52, for example the Internet. The first Firewall 50 is for example physically connected with a Tunnel Relay B 54, which in turn is virtually connected through the public network 52 with a Tunnel Relay C. The connection between Tunnel Relay B and Tunnel Relay C may for example span multiple intervening forwarding nodes such as routers or gateways through the public network 52.

The Tunnel Relay C is physically connected with a second Firewall 58, which separates the public network 52 from a private network N2 60. The second Firewall 58 is physically connected with a Tunnel Server D 62 on the private network N2 60. During operation of the elements shown in Fig. 3, the Tunnel Server D 62 provides routing of IP packets between the tunnel connection with Tunnel Server A 46 and other stations on the private network N2 60. In this way the Tunnel Server D 62 acts as a router between the tunnel connection and the private network N2 60.

During operation of the elements shown in Fig. 3, the present system establishes a tunnel connection between the private network N1 48 and the private network N2 60. The embodiment of Fig. 3 thus eliminates the need for a dedicated physical cable or line to provide secure communications between the private network 48 and the private network 60. The tunnel connection between Tunnel Server A 46 and Tunnel Server D 62 is composed of reliable, pair-wise transport layer connections between Tunnel Server A 46 (node "A"), Tunnel Relay B 54 (node "B"), Tunnel Relay C 56 (node "C"), and Tunnel Server D 62 (node "D"). For example, such pair-wise connections may be individual transport layer connections between each node A and node B, node B and node C, and node C and node D. In an alternative embodiment, as will be described below, a tunnel connection may alternatively be formed between a standalone PC in a public network and a tunnel server within a private network.

Fig. 4 and Fig. 5 show an example embodiment of steps performed during establishment of the tunnel connection between Tunnel Server A 46 (node "A") and Tunnel Server D 62 (node "D") as shown in Fig. 3. Prior to the steps shown in Fig. 4, node A selects a tunnel path to reach node D. The tunnel path includes the tunnel end points and any intervening tunnel relays. The tunnel path is for example predetermined by a system administrator for node A. Each tunnel relay along the tunnel path is capable of finding a next node in the tunnel path, for example based on a provided next node name (or "next node arc"), using a predetermined naming convention and service, for example the Domain Name System (DNS) of the TCP/IP protocol suite.

During the steps shown in Fig. 4, each of the nodes A, B and C perform the following steps:

- resolve the node name of the next node in the tunnel path, for example as found in a tunnel relay frame;
- establish a reliable transport layer (TCP) connection to the next node in the tunnel path;
- forward the tunnel relay frame down the newly formed reliable transport layer connection to the next node in the tunnel path.

As shown for example in Fig. 4, at step 70 node A establishes a reliable transport layer connection with node B. At step 72 node A identifies the next downstream node to node B by sending node B a tunnel relay frame over the reliable transport layer connection between node A and node B. The tunnel relay frame contains a string buffer describing all the nodes along the tunnel path (see below description of an example tunnel relay frame format). At step 74, responsive to the tunnel relay frame from node A, node B searches the string buffer in the relay frame to determine if the string buffer includes node B's node name. If node B finds its node name in the string buffer, it looks at the next node name in the string buffer to find the node name of the next node in the tunnel path.

Node B establishes a reliable transport layer connection with the next node in the tunnel path, for example node C. Node B further forms an association between the reliable transport layer connection between Node A and Node B, over which the relay frame was received, and the newly formed reliable transport layer connection between Node B and Node C, and as a result forwards subsequent packets received over the reliable transport layer connection with Node A onto the reliable transport layer connection with Node C, and vice versa. At step 76 node B forwards the tunnel relay frame on the newly formed reliable transport layer connection to node C.

At step 78, responsive to the relay frame forwarded from node B, node C determines that the next node in the tunnel path is the last node in the tunnel path, and accordingly is a tunnel server. Node C may actively determine whether alternative tunnel servers are available to form the tunnel connection. Node C may select one of the alternative available tunnel servers to form the tunnel connection in order to provide load balancing or fault tolerance. As a result node C may form a transport layer connections with one of several available tunnel servers, for example a tunnel server that is relatively underutilized at the time the tunnel connection is established. In the example embodiment, node C establishes a reliable transport layer connection with the next node along the tunnel path, in this case node D.

Node C further forms an association between the reliable transport layer connection between Node B and Node C, over which the relay frame was received, and the newly formed reliable transport layer connection between Node C and Node D, and as a result forwards subsequent packets received over the reliable transport layer connection with Node B to the reliable transport layer connection with Node D, and vice versa. At step 80 node C forwards the relay frame to node D on the newly formed reliable transport layer connection.

Fig. 5 shows an example of tunnel end point authentication and sharing of key exchange material provided by the present system. The present system supports passing authentication data and key exchange material through the reliable transport layer connections previously established on the tunnel path. The following are provided by use of a key exchange/authentication REQUEST frame and a key exchange/authentication RESPONSE frame:

a) mutual authentication of both endpoints of the tunnel connection;

b) establishment of shared session encryption keys and key lifetimes for encrypting/authenticating subsequent data sent through the tunnel connection;

d) agreement on a shared set of cryptographic transforms to be applied to subsequent data; and

e) exchange of any other connection-specific data between the tunnel endpoints, for example strength and type of cipher to be used, any compression of the data to be used, etc. This data can also be used by clients of this protocol to qualify the nature of the authenticated connection.

At step 90 a key exchange/authentication request frame is forwarded over the reliable transport layer connections formed along the tunnel path from node A to node D. At step 92, a key exchange/authentication response frame is forwarded from node D back to node A through the reliable transport layer connections. The attributes exchanged using the steps shown in Fig. 5 may be used for the lifetime of the tunnel connection. In an alternative embodiment the steps shown in Fig. 5 are repeated as needed for the tunnel end points to exchange sufficient key exchange material to agree upon a set of session parameters for use during the tunnel connection such as cryptographic keys, key durations, and choice of encryption/decryption algorithms.

Further in the disclosed system, the names used for authentication and access control with regard to node A and node D need not be the network layer address or physical address of the nodes. For example, in an alternative embodiment where the initiating node sending the tunnel relay frame is a stand-alone PC located within a public network, the user's name may be used for authentication and/or access control purposes. This provides a significant improvement over existing systems which base authorization on predetermined IP addresses.

Fig. 6 shows the format of an example embodiment of a tunnel relay frame. The tunnel frame formats shown in Figs. 6, 7, 8 and 9 are encapsulated within the data portion of a transport layer (TCP) frame when transmitted. Alternatively, another equivalent, connection-oriented transport layer protocol having guaranteed, in-sequence frame delivery may be used. The example TCP frame format, including TCP header fields, is conventional and not shown.

The field 100 contains a length of the frame. The field 102 contains a type of the frame, for example a type of RELAY. The field 104 contains a tunnel protocol version number. The field 106 contains an index into a string buffer field 112 at which a name of the originating node is located, for example a DNS host name of the node initially issuing the relay frame (node A in Fig. 3). The fields following the origin index field 106 contain indexes into the string buffer 112 at which names of nodes along the tunnel path are located. For example each index may be the offset of a DNS host name within the string buffer 112. In this way the field 108 contains the index of the name of the first node in the tunnel path, for example node B (Fig. 3). The field 110 contains the index of the name of the second node in the tunnel path, etc. The field 112 contains a string of node names of nodes in the tunnel path.

During operation of the present system, the initiating node, for example node A as shown in Fig. 3, transmits a tunnel relay frame such as the tunnel relay frame shown in Fig. 6. Node A sends the tunnel relay frame to the first station along the tunnel path, for example node B (Fig. 3), over a previously established reliable transport layer connection. Node B searches the string buffer in the tunnel relay frame to find its node name, for example its DNS host name. Node B finds its node name in the string buffer indexed by path index 0, and then uses the contents of path index 1 110 to determine the location within the string buffer 112 of the node name of the next node along the tunnel path. Node B uses this node name to establish a reliable transport layer connection with the next node along the tunnel path. Node B then forwards the relay frame to the next node. This process continues until the end node of the tunnel route, for example tunnel server D 62 (Fig. 3) is reached.

Fig. 7 shows the format of an example embodiment of a key exchange/key authentication request frame. The field 120 contains a length of the frame. The field 122 contains a type of the frame, for example a type of REQUEST indicating a key exchange/key authentication request frame. The field 124 contains a tunnel protocol version number. The field 126 contains an offset of the name of the entity initiating the tunnel connection, for example the name of a user on the node originally issuing the request frame. This name and key exchange material in the request frame are used by the receiving tunnel end point to authenticate the key exchange/authentication REQUEST. The name of the entity initiating the tunnel connection is also use to authorize any subsequent tunnel connection, based on predetermined security policies of the system. The field 128 contains an offset into the frame of the node name of the destination node, for example the end node of the tunnel shown as node D 62 in Fig. 3.

The field 130 contains an offset into the frame at which key exchange data as is stored, for example within the string buffer field 138. The key exchange data for example includes key exchange material used to determine a shared set of encryption parameters for the life of the tunnel connection such as cryptographic keys and any validity times associated with those keys. The key exchange data, as well as the field 132, further include information regarding any shared set of cryptographic transforms to be used and any other connection-specific parameters, such as strength and type of cipher to be used, type of compression of the data to be used, etc. The field 134 contains flags, for example indicating further information about the frame. The field 136 contains client data used in the tunnel end points to configure the local routing tables so that packets for nodes reachable through the virtual private network are sent through the pseudo network adapters. In an example embodiment, the string buffer 138 is encrypted using a public encryption key of the receiving tunnel end point.

During operation of the present system, one of the end nodes of the tunnel sends a key exchange/authentication REQUEST frame as shown in Fig. 7 to the other end node of the tunnel in order to perform key exchange and authentication as described in step 90 of Fig. 5.

Fig. 8 shows the format of an example embodiment of a key exchange/key authentication response frame, referred to as a connection RESPONSE frame. The field 150 contains a length of the frame. The field 152 contains a type of the frame, for example a type of connection RESPONSE indicating a key exchange/key authentication request frame. The field 154 contains a tunnel protocol version number.

The field 156 contains an offset into the frame at which key exchange data as is stored, for example within the string buffer field 163. The key exchange data for example includes key exchange material to be used for encryption/decryption over the life of the tunnel connection and any validity times associated with that key exchange material. The key exchange data, as well as the field 158, further includes information regarding any shared set of cryptographic transforms to be applied to subsequent data and any other connection-specific parameters, such as strength and type of cipher to be used, any compression of the data to be used, etc. The field 160 contains flags, for example indicating other information about the frame. The client data field 162 contains data used by the pseudo network adapters in the tunnel end points to configure the local routing tables so that packets for nodes in the virtual private network are sent through the pseudo network adapters. The string buffer includes key exchange material. The string buffer is for example encrypted using a public encryption key of the receiving tunnel end point, in the this case the initiator of the tunnel connection.

During operation of the present system, one of the end nodes of the tunnel sends a key exchange/authentication RESPONSE frame as shown in Fig. 7 to the other end node of the tunnel in order to perform key exchange and authentication as described in step 92 of Fig. 5.

Fig. 9 shows the format of an example embodiment of an tunnel data frame used to communicate through a tunnel connection. Fig. 9 shows how an IP datagram may be encapsulated within a tunnel frame by the present system for secure communications through a virtual private network. The field 170 contains a length of the frame. The field 172 contains a type of the frame, for example a type of DATA indicating a tunnel data frame. The field 174 contains a tunnel protocol version number.

The fields 176, 178 and 182 contain information regarding the encapsulated datagram. The field 180 contains flags indicating information regarding the frame. The field 184 contains a value indicating the length of the optional padding 189 at the end of the frame. The frame format allows for optional padding in the event that the amount of data in the frame needs to be padded to an even block boundary for the purpose of being encrypted using a block cipher. The field 186 contains a value indicating the length of the digest field 187.

The data frame format includes a digital signature generated by the transmitting tunnel end point referred to as a "digest". The value of the digest ensures data integrity, for example by detecting invalid frames and replays of previously transmitted valid frames. The digest is the output of a conventional keyed cryptographic hash function applied to both the encapsulated datagram 190 and a monotonically increasing sequence number. The resulting hash output is passed as the value of the digest field 187. The sequence number is not included in the data frame. In the example embodiment, the sequence number is a counter maintained by the transmitter (for example node A in Fig. 3) of all data frames sent to the receiving node (for example node D in Fig. 3) since establishment of the tunnel connection.

In order to determine if the data frame is invalid or a duplicate, the receiving node decrypts the encapsulated datagram 190, and applies the keyed cryptographic hash function (agreed to by the tunnel end nodes during the steps shown in Fig. 5) to both the decrypted encapsulated datagram and the value of a counter indicating the number of data frames received from the transmitter since establishment of the tunnel connection. For example the keyed hash function is applied to the datagram concatenated to the counter value. If the resulting hash output matches the value of the digest field 187, then the encapsulated datagram 190 was received correctly and is not a duplicate. If the hash output does not match the value of the digest field 187, then the integrity check fails, and the tunnel connection is closed. The field 188 contains an encrypted network layer datagram, for example an encrypted IP datagram.

The encapsulated datagram may be encrypted using various encryption techniques. An example embodiment of the present system advantageously encrypts the datagram 190 using either a stream cipher or cipher block chaining encryption over all data transmitted during the life of the tunnel connection. This is enabled by the reliable nature of the transport layer connections within the tunnel connection. The specific type of encryption and any connection specific symmetric encryption keys used is determined using the steps shown in Fig. 5. The fields in the tunnel data frame other than the encapsulated datagram 188 are referred to as the tunnel data frame header fields.

Fig. 10 is a block diagram showing an example embodiment of a "close connection" frame. The field 190 contains the length of the frame. The field 191 contains a frame type, for example having a value equal to CLOSE. Field 192 contains a value equal to the current protocol version number of the tunnel protocol. The field 193 contains a status code indicating the reason the tunnel connection is being closed.

During operation of the present system, when end point of a tunnel connection determines that the tunnel connection should be closed, a close connection frame as shown in Fig. 10 is transmitted to the other end point of the tunnel connection. When a close connection close frame is received, the receiver closes the tunnel connection and no further data will be transmitted or received through the tunnel connection.

Fig. 11 is a state diagram showing an example embodiment of forming a tunnel connection in a node initiating a tunnel connection. In Fig. 11, Fig. 12, and Fig. 13, states are indicated by ovals and actions or events are indicated by rectangles. For example the tunnel server node A as shown in Fig. 3 may act as a tunnel connection initiator when establishing a tunnel connection with the tunnel server node D. Similarly the client system 247 in Fig. 14 may act as a tunnel connection initiator when establishing a tunnel connection with the tunnel server. The tunnel initiator begins in an idle state 194. Responsive to an input from a user indicating that a tunnel connection should be established, the tunnel initiator transitions from the idle state 194 to a TCP Open state 195. In the TCP Open state 195, the tunnel initiator establishes a reliable transport layer connection with a first node along the tunnel path. For example, the tunnel initiator opens a socket interface associated with a TCP connection to the first node along the tunnel path. In Fig. 3 node A opens a socket interface associated with a TCP connection with node B.

Following establishment of the reliable transport layer connection in the TCP Open state 195, the tunnel initiator enters a Send Relay state 197. In the Send Relay state 197, the tunnel initiator transmits a relay frame at 198 over the reliable transport layer connection. Following transmission of the relay frame, the tunnel initiator enters the connect state 199. If during transmission of the relay frame there is a transmission error, the tunnel initiator enters the Network Error state 215 followed by the Dying state 208. In the Dying state 208, the tunnel initiator disconnects the reliable transport layer connection formed in the TCP Open state 195, for example by disconnecting a TCP connection with Node B. Following the disconnection at 209, the tunnel initiator enters the Dead state 210. The tunnel initiator subsequently transitions back to the Idle state 194 at a point in time predetermined by system security configuration parameters.

In the Connect state 199, the tunnel initiator sends a key exchange/authentication REQUEST frame at 200 to the tunnel server. Following transmission of the key exchange/authentication REQUEST frame 200, the tunnel initiator enters the Response Wait state 201. The tunnel initiator remains in the Response Wait state 201 until it receives a key exchange/authentication RESPONSE frame 202 from the tunnel server. After the key exchange/authentication RESPONSE frame is received at 202, the tunnel initiator enters the Authorized state 203, in which it may send or receive tunnel data frames. Upon receipt of a CLOSE connection frame at 216 in the Authorized state 203, the tunnel initiator transitions to the Dying state 208.

Upon expiration of a session encryption key at 211, the tunnel initiator enters the Reconnect state 212, and sends a CLOSE connection frame at 213 and disconnects the TCP connection with the first node along the tunnel path at 214. Subsequently the tunnel initiator enters the TCP Open state 195.

If during the authorized state 203, a local user issues an End Session command at 204, or there is a detection of an authentication or cryptography error in a received data frame at 205, the tunnel initiator enters the Close state 206. During the Close state 206 the tunnel initiator sends a CLOSE connection frame at 207 to the tunnel server. The tunnel initiator then enters the Dying state at 208.

Figure 12 is a state diagram showing the states within an example embodiment of a tunnel server, for example node D in Fig. 3 or tunnel server 253 in Fig. 14. The tunnel server begins in an Accept Wait state 217. In the Accept Wait state 217, the tunnel server receives a request for a reliable transport layer connection, for example a TCP connection reguest 218 from the last node in the tunnel path prior to the tunnel server, for example Node C in Fig. 3. In response to a TCP connection request 218 the tunnel server accepts the request and establishes a socket interface associated with the resulting TCP connection with Node C.

Upon establishment of the TCP connection with the last node in the tunnel path prior to the tunnel server, the tunnel server enters the Receive Relay state 219. In the Receive Relay state 219, the tunnel server waits to receive a relay frame at 220, at which time the tunnel server enters the Connect Wait state 221. If there is some sort of network error 234 during receipt of the relay frame at 219, the tunnel server enters the Dying state 230. During the Dying state 230 the tunnel server disconnects at 231 the transport layer connection with the last node in the tunnel path prior to the tunnel server. After disconnecting the connection, the tunnel server enters the Dead state 232.

In the Connect Wait state 221, the tunnel server waits for receipt of a key exchange/authentication REQUEST frame at 222. Following receipt of the key exchange/authentication REQUEST frame at 222, the tunnel server determines whether the requested tunnel connection is authorized at step 223. The determination of whether the tunnel connection is authorized is based on a name of the tunnel initiator, and the key exchange material within the key exchange/authentication REQUEST frame.

If the requested tunnel connection is authorized the tunnel server sends a key exchange/authentication RESPONSE frame at 224 back to the tunnel initiator. If the requested tunnel connection is not authorized, the tunnel server enters the Close state 228, in which it sends a close connection frame at 229 to the tunnel client. Following transmission of the CLOSE connection frame at 229, the tunnel server enters the Dying state 230.

If the requested tunnel connection is determined to be authorized at step 223, the tunnel server enters the Authorized state 225. In the Authorized state, the tunnel server transmits and receives tunnel data frames between itself and the tunnel initiator. If during the Authorized state 225, the tunnel server receives a CLOSE connection frame at 233, the tunnel server transitions to the Dying state 230. If during the authorized state 225, the tunnel server receives an end session command from a user at 226, then the tunnel server transitions to the Close state 228, and transmits a close connection frame at 229 to the tunnel initiator. If the tunnel server in the Authorized state 225 detects an integrity failure in a received packet, the tunnel server transitions to the Close state 228. In the close state 228 the tunnel server sends a CLOSE connection frame at 229 and subsequently enters the Dying state 230.

Fig. 13 is a state diagram showing an example embodiment of a state machine within a tunnel relay node. The tunnel relay node begins in an Accept Wait state 235. When a request is received to form a reliable transport layer connection at 236, a reliable transport layer connection is accepted with the requesting node. For example, a TCP connection is accepted between the relay node and the preceding node in the tunnel path.

The relay node then transitions to the Receive Relay state 237. During the Receive Relay state 237, the relay node receives a relay frame at 238. Following receipt of the relay frame at 238, the relay node determines what forwarding address should be used to forward frames received from the TCP connection established responsive to the TCP connect event 236. If the next node in the tunnel path is a tunnel server, the forwarding address may be selected at 239 so as to choose an underutilized tunnel server from a group of available tunnel servers or to choose an operational server where others are not operational.

Following determination of the forwarding address or addresses in step 239, the relay node enters the Forward Connect state 240. In the Forward Connect state 240, the relay node establishes a reliable transport layer connection with the node or nodes indicated by the forwarding address or addresses determined in step 239.

Following establishment of the new connection at event 241, the tunnel relay enters the Forward state 242. During the Forward state 242, the relay node forwards all frames between the connection established at 236 and those connections established at 241. Upon detection of a network error or receipt of a frame indicating a closure of the tunnel connection at 243, the tunnel relay enters the Dying state 244. Following the Dying state 244, the relay node disconnects any connections established at event 241. The relay node then enters the Dead state 246.

Fig. 14 shows an example embodiment of a virtual private network 249 formed by a pseudo network adapter 248 and a tunnel connection between a tunnel client 247 and a tunnel server 253 across a public network 251. The tunnel server 253 and tunnel client 247 are for example network stations including a CPU or microprocessor, memory, and various I/O devices. The tunnel server 253 is shown physically connected to a private LAN 256 including a Network Node 1 257 and a Network Node 2 258, through a physical network adapter 254. The tunnel server 253 is further shown physically connected with a firewall 252 which separates the private LAN 256 from the public network 251. The firewall 252 is physically connected with the public network 251. The tunnel server 253 is further shown including a pseudo network adapter 255. The client system 247 is shown including a physical network adapter 250 physically connected to the public network 251.

During operation of the elements shown in Fig. 14, nodes within the virtual private network 249 appear to the tunnel client 247 as if they were physically connected to the client system through the pseudo network adapter 248. Data transmissions between the tunnel client and any nodes that appear to be within the virtual private network are passed through the pseudo network adapter 248. Data transmissions between the tunnel client 247 and the tunnel server 253 are physically accomplished using a tunnel connection between the tunnel client 247 and the tunnel server 253.

Fig. 15 shows elements in an example embodiment of a pseudo network adapter such as the pseudo network adapter 248 in Fig. 14. In an example embodiment the elements shown in Fig. 15 are implemented as software executing on the tunnel client 247 as shown in Fig. 14. In Fig. 15 there is shown a pseudo network adapter 259 including a virtual adapter driver interface 263, an encapsulation engine 264, an encryption engine 265, a decapsulation engine 268, and a decryption engine 266. Further shown in the pseudo network adapter 259 are an ARP server emulator 270 and a Dynamic Host Configuration Protocol (DHCP) server emulator.

The pseudo network adapter 259 is shown interfaced to a TCP/IP protocol stack 260, through the virtual adapter driver interface 260. The TCP/IP protocol stack 260 is shown interfaced to other services in an operating system 261, as well as a physical network adapter's driver 262. The physical network adapter's driver 262 is for example a device driver which controls the operation of a physical network adapter such as physical network adapter 250 as shown in Fig. 14.

During operation of the elements shown in Fig. 15, the pseudo network adapter 259 registers with the network layer in the TCP/IP stack 260 that it is able to reach the IP addresses of nodes within the virtual private network 249 as shown in Fig. 14. For example, the pseudo network adapter on the client system registers that it can reach the pseudo network adapter on the server. Subsequently, a message from the tunnel client addressed to a node reachable through the virtual private network will be passed by the TCP/IP stack to the pseudo network adapter 259. The pseudo network adapter 259 then encrypts the message, and encapsulates the message into a tunnel data frame. The pseudo network adapter 259 then passes the tunnel data frame back to the TCP/IP protocol stack 260 to be sent through to the physical network adapter in the tunnel server. The tunnel server passes the received data frame to the pseudo network adapter in the server, which de-encapsulates and decrypts the message.

Fig. 16 shows a more detailed example embodiment of a pseudo network adapter 280. The pseudo network adapter 280 includes a virtual network adapter driver interface 288. The transmit path 290 includes an encryption engine 292, and an encapsulation engine 294. The encapsulation engine 294 is interfaced with a TCP/IP transmit interface 312 within a TCP/IP protocol stack, for example a socket interface associated with the first relay node in the tunnel path, or with the remote tunnel end point if the tunnel path includes no relays.

In the example embodiment of Fig. 16, the pseudo network adapter 280 appears to the TCP/IP protocol stack 282 as an Ethernet adapter. Accordingly, ethernet packets 286 for a destination addresses understood by the TCP/IP protocol stack to be reachable through the virtual private network are passed from the TCP/IP protocol stack 282 to the virtual network adapter interface 288 and through the transmit path 290. Similarly, ethernet packets 284 received through the pseudo network adapter 280 are passed from the receive path 296 to the virtual network adapter interface 288 and on to the TCP/IP protocol stack 282.

Further shown in the pseudo network adapter 280 of Fig. 16 is a receive path 296 having a decryption engine 298 interfaced to the virtual network adapter interface 288 and a decapsulation engine 300. The decapsulation engine 300 in turn is interfaced to a TCP/IP receive function 314 in the TCP/IP protocol stack 282, for example a socket interface associated with the first relay in the tunnel path, or with the remote tunnel end point if the tunnel path includes no relays. The pseudo network adapter 280 further includes an ARP server emulator 304 and a DHCP server emulator 306. ARP and DHCP request packets 302 are passed to the ARP server emulator 304 and DHCP server emulator 306 respectively. When a received packet is passed from the receive path 296 to the TCP/IP stack 282, a receive event must be indicated to the TCP/IP stack 282, for example through an interface such the Network Device Interface Specification (NDIS), defined by Microsoft™ Corporation.

Also in Fig. 16 is shown is an operating system 310 coupled with the TCP/IP protocol stack 282. The TCP/IP protocol stack 282 is generally considered to be a component part of the operating system. The operating system 310 in Fig. 16 is accordingly the remaining operating system functions and procedures outside the TCP/IP protocol stack 282. A physical network adapter 308 is further shown operated by the TCP/IP protocol stack 282.

During operation of the elements shown in Fig. 16, a user passes data for transmission to the TCP/IP protocol stack 282, and indicates the IP address of the node to which the message is to be transmitted, for example through a socket interface to the TCP layer. The TCP/IP protocol stack 282 then determines whether the destination node is reachable through the virtual private network. If the message is for a node that is reachable through the virtual private network, the TCP/IP protocol stack 282 an ethernet packet 286 corresponding to the message to the pseudo network adapter 280. The pseudo network adapter 280 then passes the ethernet packet 286 through the transmit path, in which the ethernet packet is encrypted and encapsulated into a tunnel data frame. The tunnel data frame is passed back into the TCP/IP protocol stack 282 through the TCP/IP transmit function 312 to be transmitted to the tunnel server through the tunnel connection. In an example embodiment, a digest value is calculated for the tunnel data frame before encryption within the transmit path within the pseudo network adapter.

Further during operation of the elements shown in Fig. 16, when the TCP/IP protocol stack 282 receives a packet from the remote endpoint of the TCP/IP tunnel connection, for example the tunnel server, the packet is passed to the pseudo network adapter 280 responsive to a TCP receive event. The pseudo network adapter 280 then decapsulates the packet by removing the tunnel header. The pseudo network adapter further decrypts the decapsulated data and passes it back to the TCP/IP protocol stack 282. The data passed from the pseudo network adapter 280 appears to the TCP/IP protocol stack 282 as an ethernet packet received from an actual physical device, and is the data it contains is passed on to the appropriate user by the TCP/IP protocol stack 282 based on information in the ethernet packet header provided by the pseudo network adapter.

Fig. 17 is a flow chart showing steps performed by an example embodiment of a pseudo network adapter during packet transmission, such as in the transmit path 290 of Fig. 14. The TCP/IP protocol stack determines that the destination node of a packet to be transmitted is reachable through the virtual LAN based on the destination IP address of the packet and a network layer routing table. At step 320 the packet is passed to the pseudo network adapter from the TCP/IP protocol stack. As a result, a send routine in the pseudo adapter is triggered for example in the virtual network adapter interface 288 of Fig. 16.

At step 322 the pseudo network adapter send routine processes the Ethernet header of the packet provided by the TCP/IP stack, and removes it. At step 324, the send routine determines whether the packet is an ARP request packet. If the packet is an ARP request packet for an IP address of a node on the virtual LAN, such as the pseudo network adapter of the tunnel server, then step 324 is followed by step 326. Otherwise, step 324 is followed by step 330.

At step 326, the ARP server emulator in the pseudo network adapter generates an ARP reply packet. For example, if the ARP request were for a physical address corresponding to the IP address of the pseudo network adapter on the tunnel server, the ARP reply would indicate a predetermined, reserved physical address to be associated with that IP address. At step 328 the pseudo network adapter passes the ARP response to the virtual network adapter interface. The virtual network adapter interface then indicates a received packet to the TCP/IP protocol stack, for example using an NDIS interface. The TCP/IP protocol stack then processes the ARP response as if it had been received over an actual physical network.

At step 330 the send routine determines whether the packet is a DHCP request packet requesting an IP address for the pseudo network adapter. If so, then step 330 is followed by step 332. Otherwise, step 330 is followed by step 334.

At step 334, the DHCP server emulator in the pseudo network adapter generates a DHCP response. The format of DHCP is generally described in the DHCP RFC. At step 328 the pseudo network adapter passes the DHCP response to the virtual network adapter interface, for example indicating an IP address received from the tunnel server in the client data field of the key exchange/authentication RESPONSE frame. The virtual network adapter interface then indicates a received packet to the TCP/IP protocol stack. The TCP/IP protocol stack then processes the DHCP response as if it had been received over an actual physical network.

At step 334 the pseudo network adapter encrypts the message using an encryption engine such that only the receiver is capable of decrypting and reading the message. At step 336 the pseudo network adapter encapsulates the encrypted message into a tunnel data frame. At step 338 the pseudo network adapter transmits the tunnel data frame through the tunnel connection using the TCP/IP protocol stack.

Fig. 18 is a flow chart showing steps performed by an example embodiment of a pseudo network adapter during packet receipt, such as in the receive path 296 of Fig. 14.

At step 350, the pseudo network adapter is notified that a packet has been received over the tunnel connection. At step 352 the pseudo network adapter decapsulates the received message by removing the header fields of the tunnel data frame. At step 354 the pseudo network adapter decrypts the decapsulated datagram from the tunnel data frame. At step 356, in an example embodiment, the pseudo network adapter forms an Ethernet packet from the decapsulated message. At step 358 the pseudo network adapter indicates that an Ethernet packet has been received to the TCP/IP protocol stack through the virtual network adapter interface. This causes the TCP/IP protocol stack to behave as if it had received an Ethernet packet from an actual Ethernet adapter.

Fig. 19 shows the data flow within the transmit path in an example embodiment of a pseudo network adapter. At step 1 370, an application submits data to be transmitted to the TCP protocol layer 372 within the TCP/IP protocol stack. The application uses a conventional socket interface to the TCP protocol layer 372 to pass the data, and indicates the destination IP address the data is to be transmitted to. The TCP protocol layer 372 then passes the data to the IP protocol layer 374 within the TCP/IP protocol stack. At step 2 376, the TCP/IP protocol stack refers to the routing table 378 to determine which network interface should be used to reach the destination IP address.

Because in the example the destination IP address is of a node reachable through the virtual private network, the IP layer 374 determines from the routing table 378 that the destination IP address is reachable through pseudo network adapter. Accordingly at step 3 380 the TCP/IP protocol stack passes a packet containing the data to the pseudo network adapter 382.

At step 4 384, the pseudo network adapter 382 encrypts the data packets and encapsulates them into tunnel data frames.

The pseudo network adapter 382 then passes the tunnel data frames packets back to the TCP protocol layer 372 within the TCP/IP protocol stack through a conventional socket interface to the tunnel connection with the first node in the tunnel path.

The TCP protocol layer 372 then forms a TCP layer packet for each tunnel data frame, having the tunnel data frame as its data. The TCP frames are passed to the IP layer 374. At step 5 386 the routing table 378 is again searched, and this time the destination IP address is the IP address associated with the physical network adapter on the tunnel server, and accordingly is determined to be reachable over the physical network adapter 390. Accordingly at step 6 388 the device driver 390 for the physical network adapter is called to pass the packets to the physical network adapter. At step 7 392 the physical network adapter transmits the data onto the physical network 394.

Fig. 20 is a data flow diagram showing data flow in an example embodiment of packet receipt involving a pseudo network adapter. At step 1 410 data arrives over the physical network 412 and is received by the physical network adapter and passed to the physical network driver 414. The physical network driver 414 passes the data at step 2 418 through the IP layer 420 and TCP layer 422 to the pseudo network adapter 426 at step 3 424, for example through a conventional socket interface. At step 4 428 the pseudo network adapter 426 decrypts and decapsulates the received data and passes it back to the IP layer of the TCP/IP protocol stack, for example through the TDI (Transport Layer Dependent Interface API) of the TCP/IP stack. The data is then passed through the TCP/IP protocol stack and to the user associated with the destination IP address in the decapsulated datagrams at step 5 430.

Fig. 21 shows data flow in an example embodiment of packet transmission involving a pseudo network adapter. Fig. 21 shows an example embodiment for use on a Microsoft™ Windows 95™ PC platform. In Fig. 21 a user application 450 passes unencrypted data to an interface into the TCP layer of the TCP/IP protocol, for example the WinSock API 452. The user indicates a destination IP address associated with a node reachable through a virtual private network accessible through the pseudo network adapter.

The TCP layer 454 passes the data to the IP layer 456, which in turn passes the data to the Network Device Interface Specification Media Access Control (NDIS MAC) interface 458. The pseudo network adapter 459 has previously registered with the routing layer (IP) that it is able to reach a gateway address associated with the destination IP address for the user data. Accordingly the IP layer uses the NDIS MAC layer interface to invoke the virtual device driver interface 460 to the pseudo network adapter 459. The pseudo network adapter 459 includes a virtual device driver interface 460, an ARP server emulator 462, and a DHCP server emulator 464.

In the example embodiment of Fig. 19, the pseudo network adapter 459 passes the data to a tunnel application program 466. The tunnel application program 466 encrypts the IP packet received from the IP layer and encapsulates it into a tunnel data frame. The tunnel application then passes the tunnel data frame including the encrypted data to the WinSock interface 452, indicating a destination IP address of the remote tunnel end point. The tunnel data frame is then passed through the TCP layer 454, IP layer 456, NDIS MAC layer interface 458, and physical layer 468, and transmitted on the network 470. Since the resulting packets do not contain a destination IP address which the pseudo network adapter has registered to convey, these packets will not be diverted to the pseudo network adapter.

Fig. 22 is a data flow diagram showing data flow in an example embodiment of packet transmission involving a pseudo network adapter. The embodiment shown in Fig. 22 is for use on a UNIX platform. In Fig. 20 a user application 472 passes unencrypted data to a socket interface to the TCP/IP protocol stack in the UNIX socket layer 474, indicating a destination IP address of a node reachable through the virtual private network.

The UNIX socket layer 474 passes the data through the TCP layer 476 and the IP layer 478. The pseudo network adapter 480 has previously registered with the routing layer (IP) that it is able to reach a gateway associated with the destination IP address for the user data. Accordingly the IP layer 478 invokes the virtual device driver interface 482 to the pseudo network adapter 480. The IP layer 478 passes the data to the pseudo network adapter 480. The pseudo network adapter 480 includes a virtual device driver interface 482, and a DHCP server emulator 484.

In the example embodiment of Fig. 22, the pseudo network adapter 480 passes IP datagrams to be transmitted to a UNIX Daemon 486 associated with the tunnel connection. The UNIX Daemon 486 encrypts the IP packet(s) received from the IP layer 478 and encapsulates them into tunnel data frames. The UNIX Daemon 486 then passes the tunnel data frames to the UNIX socket layer 474, through a socket associated with the tunnel connection. The tunnel data frames are then processed by the TCP layer 476, IP layer 478, data link layer 488, and physical layer 490 to be transmitted on the network 492. Since the resulting packets are not addressed to an IP address which the pseudo network adapter 480 has registered to convey, the packets will not be diverted to the pseudo network adapter 480.

Fig. 23 is a flow chart showing steps to initialize a example embodiment of a virtual private network. The steps shown in Fig. 23 are performed for example in the tunnel client 247 as shown in Fig. 14. At step 500 a tunnel application program executing in the tunnel client sends a tunnel relay frame to the tunnel server. At step 502 the tunnel application program sends a tunnel key exchange/authentication REQUEST frame to the tunnel server. The tunnel application in the tunnel server ignores the contents of the client data field in the tunnel key exchange/authentication REQUEST frame. The tunnel application in the tunnel server fills in the client data field in the tunnel key exchange/authentication RESPONSE frame with Dynamic Host Configuration Protocol (DHCP) information, for example including the following information in standard DHCP format:

1) IP Address for tunnel client Pseudo Network Adapter
2) IP Address for tunnel server Pseudo Network Adapter
3) Routes to nodes on the private network physically connected to the tunnel server which are to be reachable over the tunnel connection.

At step 504 the tunnel application receives a tunnel key exchange/authentication RESPONSE frame from the tunnel server. The client data field 508 in the tunnel connection response is made available to the pseudo network adapter in the tunnel client. The tunnel application in the tunnel client tells the TCP/IP stack that the pseudo network adapter in the tunnel client is active. The pseudo network adapter in the tunnel client is active and ready to be initialized at step 510.

The tunnel client system is configured such that it must obtain an IP address for the tunnel client pseudo network adapter dynamically. Therefore the TCP/IP stack in the tunnel client broadcasts a DHCP request packet through the pseudo network adapter. Accordingly, at step 512 the pseudo network adapter in the client receives a conventional DHCP request packet from the TCP/IP stack requesting a dynamically allocated IP address to associate with the pseudo network adapter. The pseudo network adapter passes the DHCP request packet to the DHCP server emulator within the pseudo network adapter, which forms a DHCP response based on the client data 508 received from the tunnel application. The DHCP response includes the IP address for the client pseudo adapter provided by the tunnel server in the client data. At step 514 the pseudo network adapter passes the DHCP response to the TCP/IP stack.

At step 520, the tunnel application modifies the routing tables within the tunnel client TCP/IP stack to indicate that the routes to the nodes attached to the private network to which the tunnel server is attached all are reachable only through the pseudo network adapter in the tunnel server. The IP address of the pseudo network adapter in the tunnel server provided in the client data is in this way specified as a gateway to the nodes on the private network to which the tunnel server is attached. In this way those remote nodes are viewed by the TCP/IP stack as being reachable via the virtual private network through the client pseudo network adapter.

At step 516 the pseudo network adapter in the tunnel client receives an ARP request for a physical address associated with the IP address of the pseudo network adapter in the tunnel server. The pseudo network adapter passes the ARP request to the ARP server emulator, which forms an ARP reply indicating a reserved physical address to be associated with the IP address of the pseudo network adapter in the tunnel server. At step 518 the pseudo network adapter passes the ARP response to the TCP/IP stack in the tunnel client. In response to the ARP response, the TCP/IP stack determines that packets addressed to any node on the virtual private network must be initially transmitted through the pseudo network adapter.

In an example embodiment the present system reserves two physical addresses to be associated with the pseudo network adapter in the client and the pseudo network adapter in the server respectively. These reserved physical addresses are used in responses to ARP requests passed through the pseudo network adapter for physical addresses corresponding to the IP addresses for the pseudo network adapter in the client and the pseudo network adapter in the server respectively. The reserved physical addresses should have a high likelihood of not being used in any actual network interface.

While the invention has been described with reference to specific example embodiments, the description is not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as other embodiments of the invention, will be apparent to persons skilled in the art upon reference to this description. Specifically, while various embodiments have been described using the TCP/IP protocol stack, the invention may advantageously be applied where other communications protocols are used. Also, while various flow charts have shown steps performed in an example order, various implementations may use altered orders of step in order to apply the invention. And further, while certain specific software and/or hardware platforms have been used in the description, the invention may be applied on other platforms with similar advantage. It is therefore contemplated that the appended claims will cover any such modifications or embodiments which fall within the scope of the invention.

Claims

1. A pseudo network adapter providing a virtual private network, comprising:

   an interface for capturing packets from a local communications protocol stack for transmission on said virtual private network, said interface appearing to said local communications protocol stack as a network adapter device driver for a network adapter connected to said virtual private network;
   a first server emulator, providing a first reply packet responsive to a first request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said first request packet requesting a network layer address for said pseudo network adapter, said first reply indicating a network layer address for said pseudo network adapter; and
   a second server emulator, providing a second reply packet responsive to an second request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said second request packet requesting a physical address corresponding to a network layer address of a second pseudo network adapter, said second pseudo network adapter located on a remote server node, said second reply indicating a predetermined, reserved physical address.

2. The pseudo network adapter of claim 1, further comprising a means for indicating to said local communications protocol stack that said predetermined, reserved physical address is reachable through said pseudo network adapter, wherein said means for indicating modifies a data structure in said local communications protocol stack indicating which nodes or networks are reachable through each network interface of the local system.

3. The pseudo network adapter of claim 1, further comprising a means for indicating to said local communications protocol stack that one or more nodes on a remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node.

4. The pseudo network adapter of claim 1, further comprising:

a transmit path for processing data packets captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network;
an encryption engine, within said transmit path, for encrypting said data packets;
an encapsulation engine, within said transmit path, for encapsulating said encrypted data packets into tunnel data frames; and
a means for passing said tunnel data frames back to said local communications protocol stack for transmission to a physical network adapter on said remote server node.

5. The pseudo network adapter of claim 4, wherein said transmit path further includes means for storing a digest value in a digest field in each of said tunnel data frames, said digest value equal to an output of a keyed hash function applied to said data packet encapsulated within said tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to said remote server node.

6. The pseudo network adapter of claim 4, wherein said transmit path further includes means for processing an Ethernet header in each one of said captured data packets, said processing of said Ethernet header including removing said Ethernet header.

7. The pseudo network adapter of claim 1, further comprising:

an interface into a transport layer of said local communications protocol stack for capturing received data packets from said remote server node.

8. The pseudo network adapter of claim 7, further comprising:

a receive path for processing received data packets captured by said interface into said transport layer of said local communications protocol stack for capturing received data packets from said remote server node;
an decapsulation engine, within said receive path, for decapsulating said received data packets by removing a tunnel frame header;
an decryption engine, within said receive path, for decrypting said received data packets; and
a means for passing said received data packets back to said local communications protocol stack for delivery to a user.

9. A method for providing a pseudo network adapter for a virtual private network, comprising the steps of:

capturing packets from a local communications protocol stack for transmission on said virtual private network, said capturing through an interface appearing to said local communications stack as a network adapter device driver for a network adapter connected to said virtual private network;
issuing a first reply packet responsive to a first request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said first request packet requesting a network layer address for said pseudo network adapter, said first reply indicating a network layer address for said pseudo network adapter; and
issuing a second reply packet responsive to a second request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said second request packet requesting a physical address corresponding to a network layer address of a second pseudo network adapter, said second pseudo network adapter located on a remote server node, said ARP Reply indicating a predetermined, reserved physical address.

10. The method of claim 9, further comprising indicating to said local communications protocol stack that said predetermined, reserved physical address is reachable through said pseudo network adapter, wherein said step of indicating to said local communications protocol stack modifies a data structure in said local communications protocol stack indicating which nodes or networks are reachable through each network interface of the local system.

11. The method of claim 9, further comprising indicating to said local communications protocol stack that one or more nodes on a remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node, wherein said step of indicating to said local communications protocol stack that one or more nodes on said remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node modifies a network layer routing table in said local communications protocol stack.

12. The method of claim 9, further comprising:

processing data packets captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network in a transmit data path;

encrypting said data packets in an encryption engine, within said transmit path;

encapsulating said encrypted data packets into tunnel data frames by an encapsulation engine, within said transmit path; and

passing said tunnel data frames back to said local communications protocol stack for transmission to a physical network adapter on said remote server node, wherein said transmit path further includes storing a digest value in a digest field in each of said tunnel data frames, said digest value equal to an output of a keyed hash function applied to said data packet encapsulated within said tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to said remote server node.

13. The method of claim 12, wherein said transmit path further includes processing an Ethernet header in each one of said captured data packets, said processing of said Ethernet header including removing said Ethernet header.

14. The method of claim 9, further comprising capturing received data packets from said remote server node through an interface into a transport layer of said local communications protocol stack, further comprising:

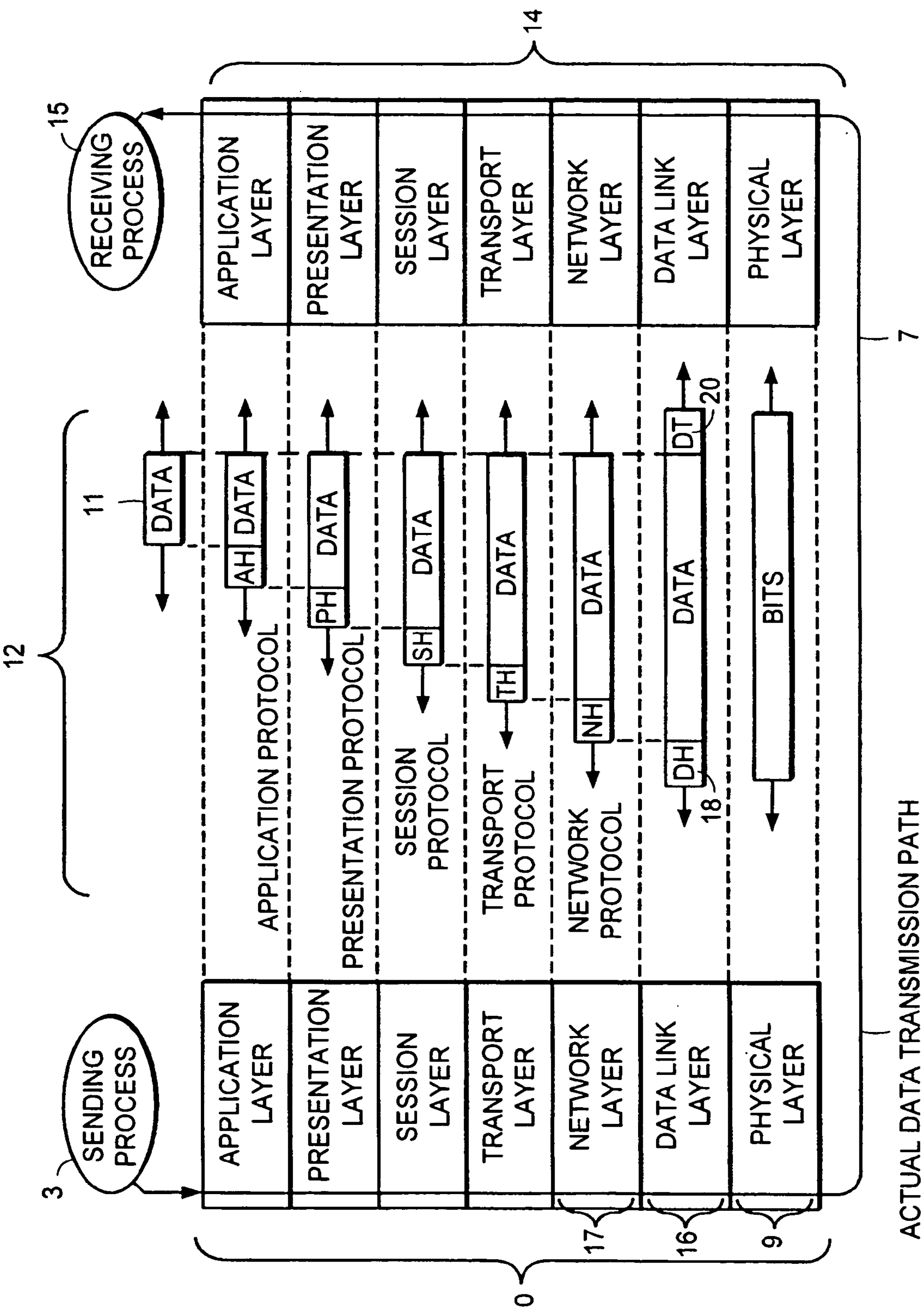
processing received data packets captured by said interface into said transport layer of said local communications protocol stack for capturing received data packets from said remote server node in a receive path;

decapsulating said received data packets by removing a tunnel frame header in an decapsulation engine, within said receive path;

decrypting said received data packets in a decryption engine within said receive path; and

passing said received data frames packets back to said local communications protocol stack for delivery to a user.

15. The method of claim 9, wherein said network layer address for said pseudo network adapter and said predetermined, reserved physical address is communicated to said pseudo network adapter from said remote server node as client data in a connection response frame.



SENDING PROCESS
3
RECEIVING PROCESS
15
APPLICATION LAYER
PRESENTATION LAYER
SESSION LAYER
TRANSPORT LAYER
NETWORK LAYER
DATA LINK LAYER
PHYSICAL LAYER
APPLICATION PROTOCOL
PRESENTATION PROTOCOL
SESSION PROTOCOL
TRANSPORT PROTOCOL
NETWORK PROTOCOL
DATA
AH
PH
SH
TH
NH
DH
DT
BITS
11
12
14
10
17
16
9
18
20
7
ACTUAL DATA TRANSMISSION PATH

FIG. 1

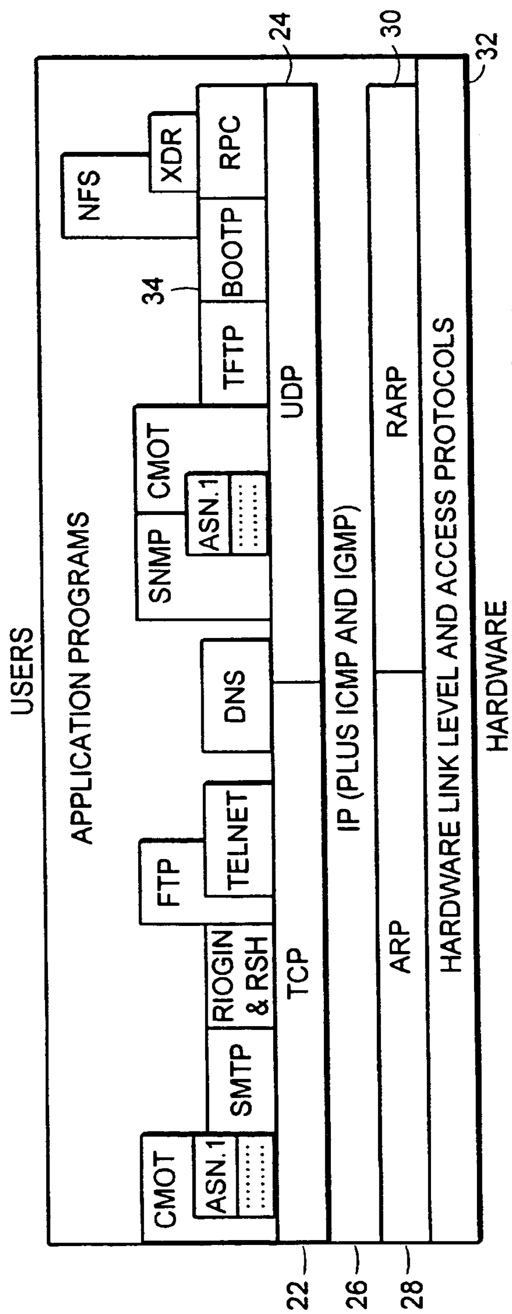

USERS
APPLICATION PROGRAMS
CMOT
ASN.1
SMTP
RIOGIN & RSH
FTP
TELNET
DNS
SNMP
CMOT
ASN.1
TFTP
BOOTP
NFS
XDR
RPC
34
24
TCP
UDP
IP (PLUS ICMP AND IGMP)
ARP
RARP
30
HARDWARE LINK LEVEL AND ACCESS PROTOCOLS
HARDWARE
32
22
26
28

FIG. 2

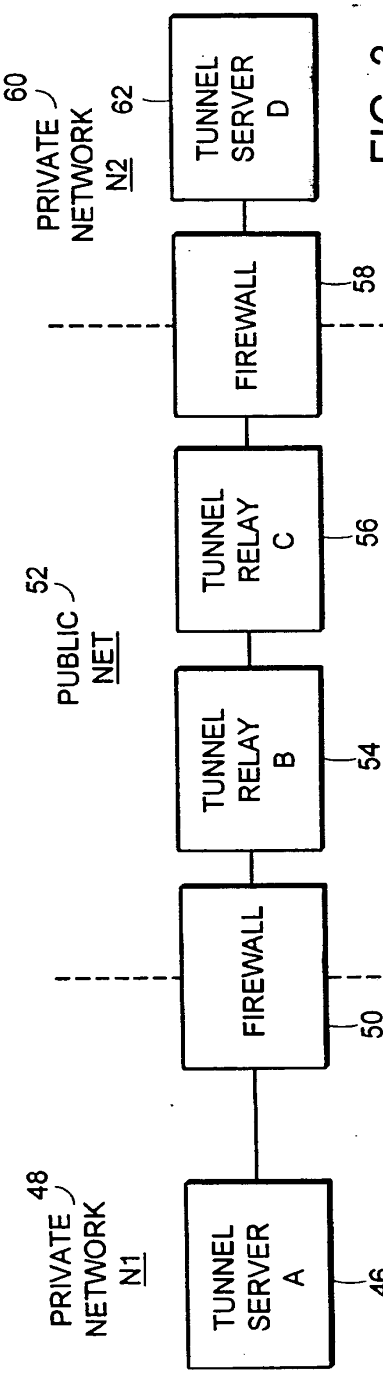

PRIVATE NETWORK N1
48
TUNNEL SERVER A
46
FIREWALL
50
PUBLIC NET
52
TUNNEL RELAY B
54
TUNNEL RELAY C
56
FIREWALL
58
PRIVATE NETWORK N2
60
62
TUNNEL SERVER D

FIG. 3

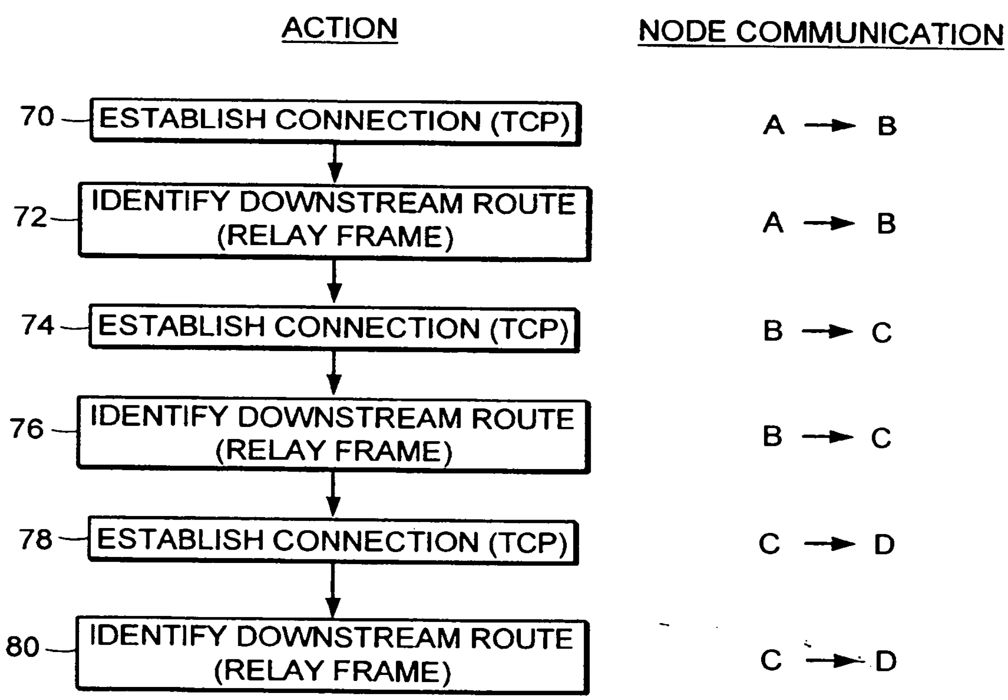

ACTION
NODE COMMUNICATION
70
ESTABLISH CONNECTION (TCP)
A → B
72
IDENTIFY DOWNSTREAM ROUTE (RELAY FRAME)
A → B
74
ESTABLISH CONNECTION (TCP)
B → C
76
IDENTIFY DOWNSTREAM ROUTE (RELAY FRAME)
B → C
78
ESTABLISH CONNECTION (TCP)
C → D
80
IDENTIFY DOWNSTREAM ROUTE (RELAY FRAME)
C → D

FIG. 4

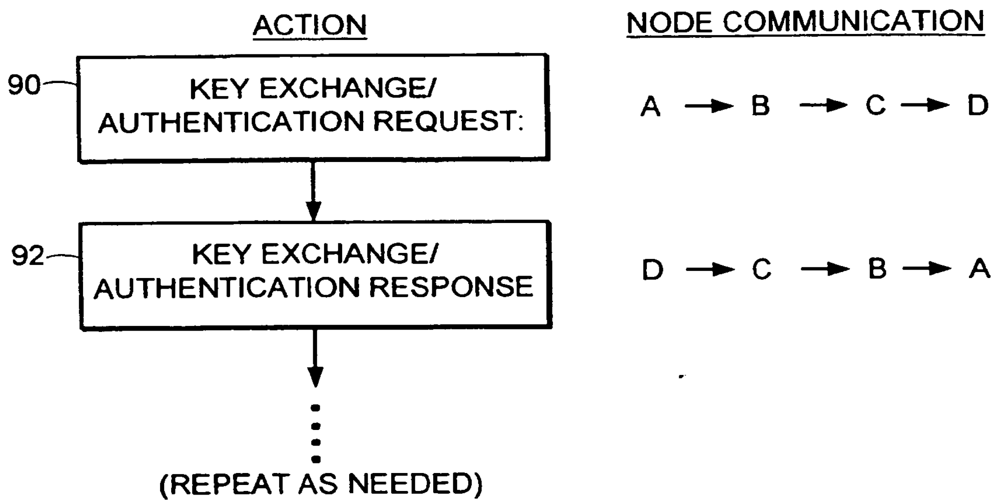

ACTION
NODE COMMUNICATION
90
KEY EXCHANGE/
AUTHENTICATION REQUEST:
A → B → C → D
92
KEY EXCHANGE/
AUTHENTICATION RESPONSE
D → C → B → A
(REPEAT AS NEEDED)

FIG. 5

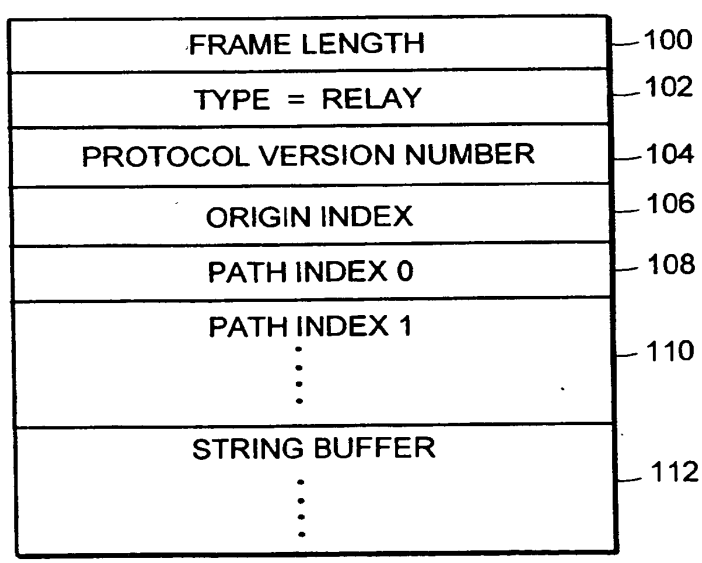

FRAME LENGTH
100
TYPE = RELAY
102
PROTOCOL VERSION NUMBER
104
ORIGIN INDEX
106
PATH INDEX 0
108
PATH INDEX 1
110
STRING BUFFER
112

FIG. 6

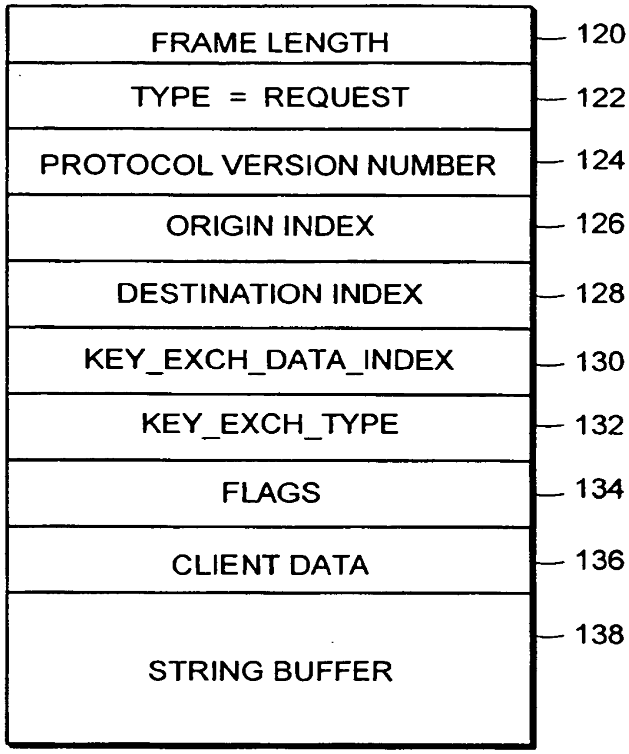

FRAME LENGTH
120
TYPE = REQUEST
122
PROTOCOL VERSION NUMBER
124
ORIGIN INDEX
126
DESTINATION INDEX
128
KEY_EXCH_DATA_INDEX
130
KEY_EXCH_TYPE
132
FLAGS
134
CLIENT DATA
136
STRING BUFFER
138

FIG. 7

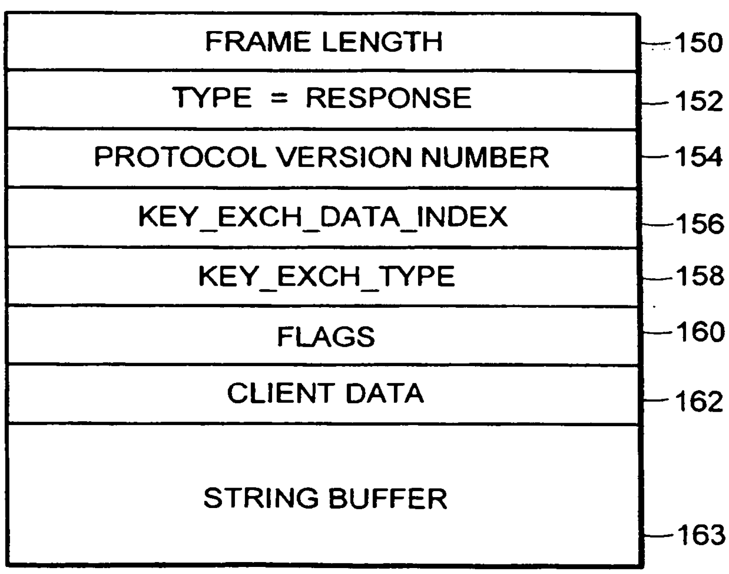

FRAME LENGTH
150
TYPE = RESPONSE
152
PROTOCOL VERSION NUMBER
154
KEY_EXCH_DATA_INDEX
156
KEY_EXCH_TYPE
158
FLAGS
160
CLIENT DATA
162
STRING BUFFER
163

FIG. 8

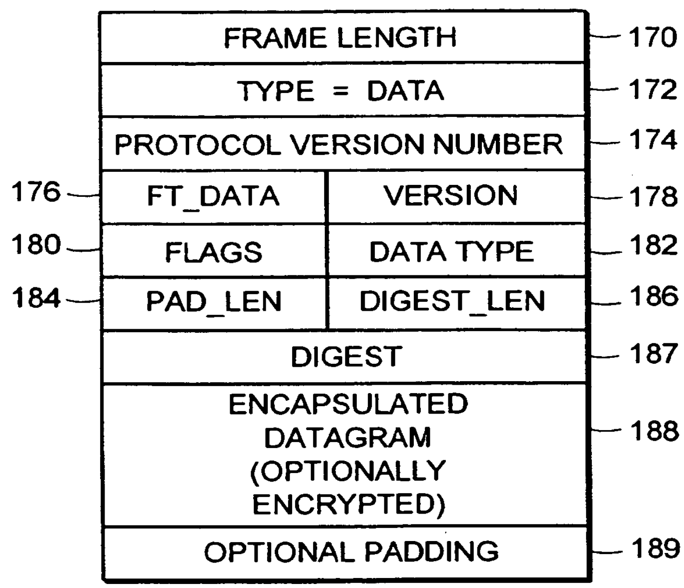

FRAME LENGTH
170
TYPE = DATA
172
PROTOCOL VERSION NUMBER
174
176
FT_DATA
VERSION
178
180
FLAGS
DATA TYPE
182
184
PAD_LEN
DIGEST_LEN
186
DIGEST
187
ENCAPSULATED DATAGRAM (OPTIONALLY ENCRYPTED)
188
OPTIONAL PADDING
189

FIG. 9

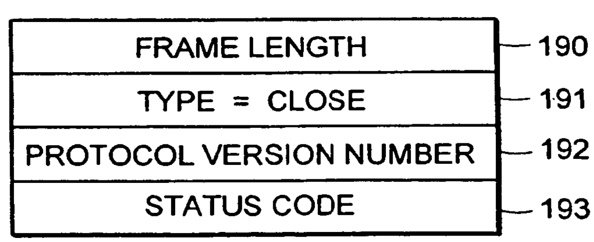

FRAME LENGTH
190
TYPE = CLOSE
191
PROTOCOL VERSION NUMBER
192
STATUS CODE
193

FIG. 10

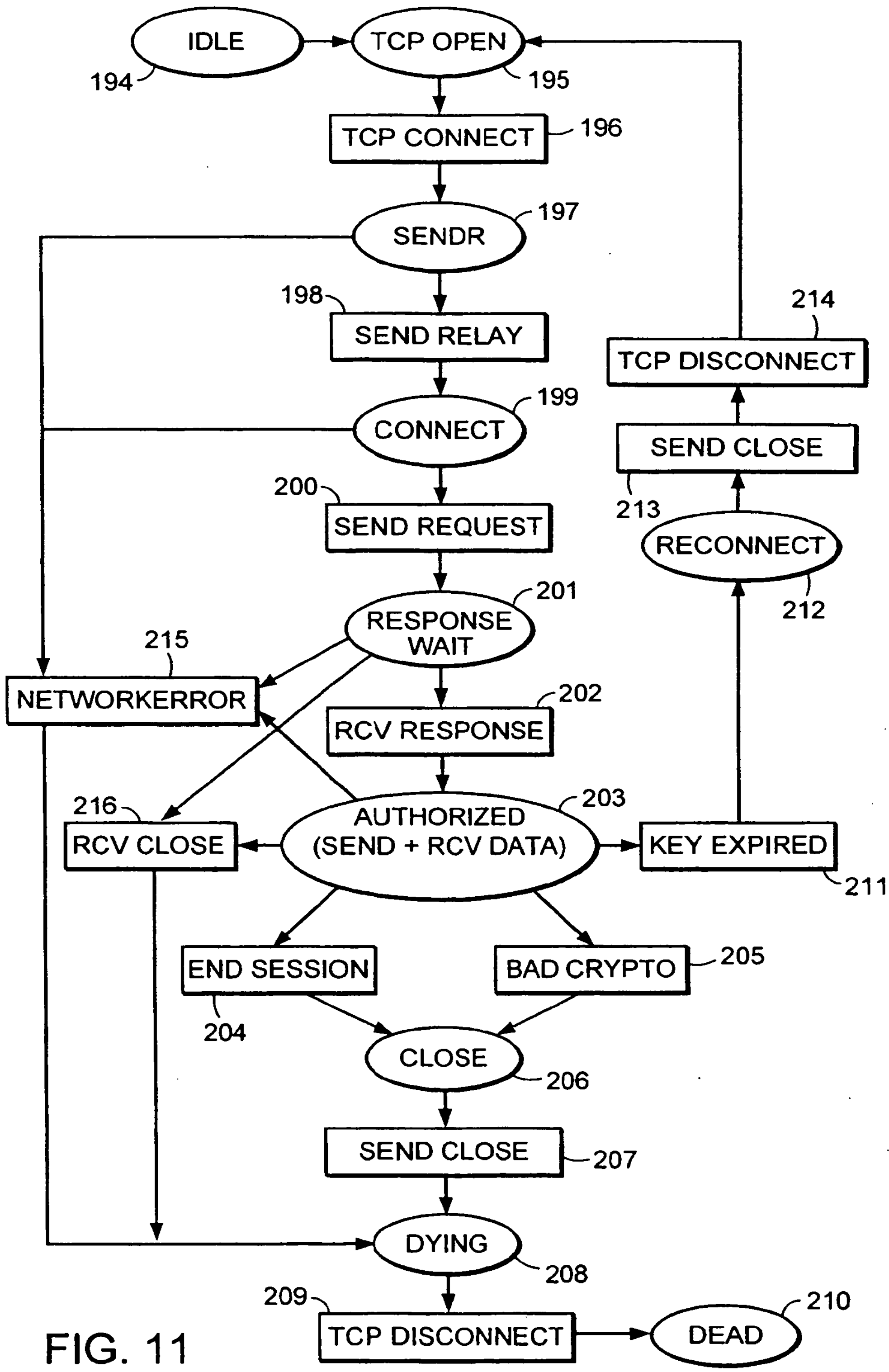
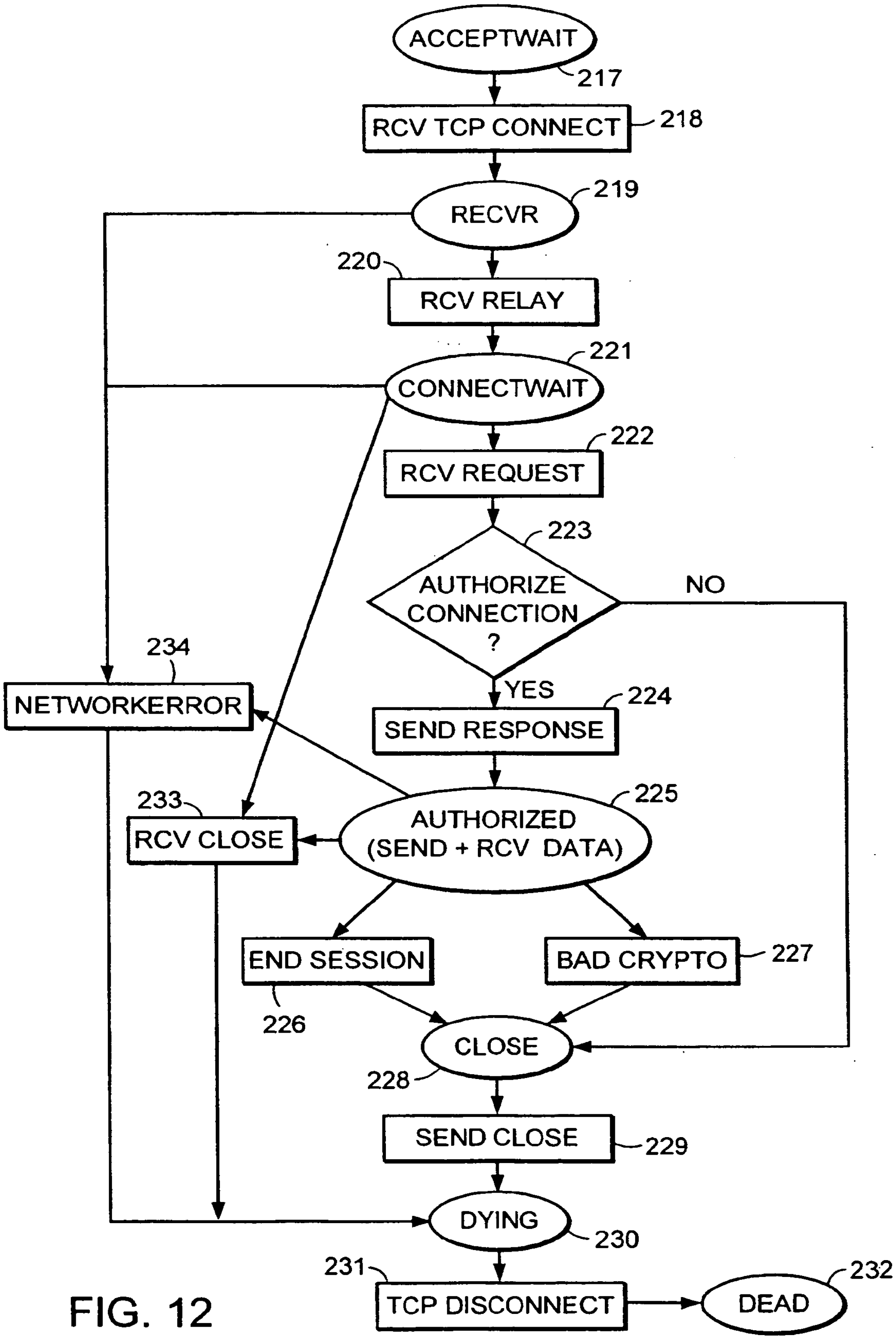

IDLE
194
TCP OPEN
195
TCP CONNECT
196
SENDR
197
198
SEND RELAY
CONNECT
199
200
SEND REQUEST
RESPONSE WAIT
201
215
NETWORKERROR
RCV RESPONSE
202
216
RCV CLOSE
AUTHORIZED (SEND + RCV DATA)
203
KEY EXPIRED
211
END SESSION
204
BAD CRYPTO
205
CLOSE
206
SEND CLOSE
207
DYING
208
209
TCP DISCONNECT
DEAD
210
214
TCP DISCONNECT
SEND CLOSE
213
RECONNECT
212

FIG. 11



ACCEPTWAIT
217
RCV TCP CONNECT
218
RECVR
219
220
RCV RELAY
CONNECTWAIT
221
RCV REQUEST
222
223
AUTHORIZE CONNECTION ?
NO
YES
234
NETWORKERROR
SEND RESPONSE
224
233
RCV CLOSE
AUTHORIZED (SEND + RCV DATA)
225
END SESSION
226
BAD CRYPTO
227
CLOSE
228
SEND CLOSE
229
DYING
230
231
TCP DISCONNECT
DEAD
232

FIG. 12

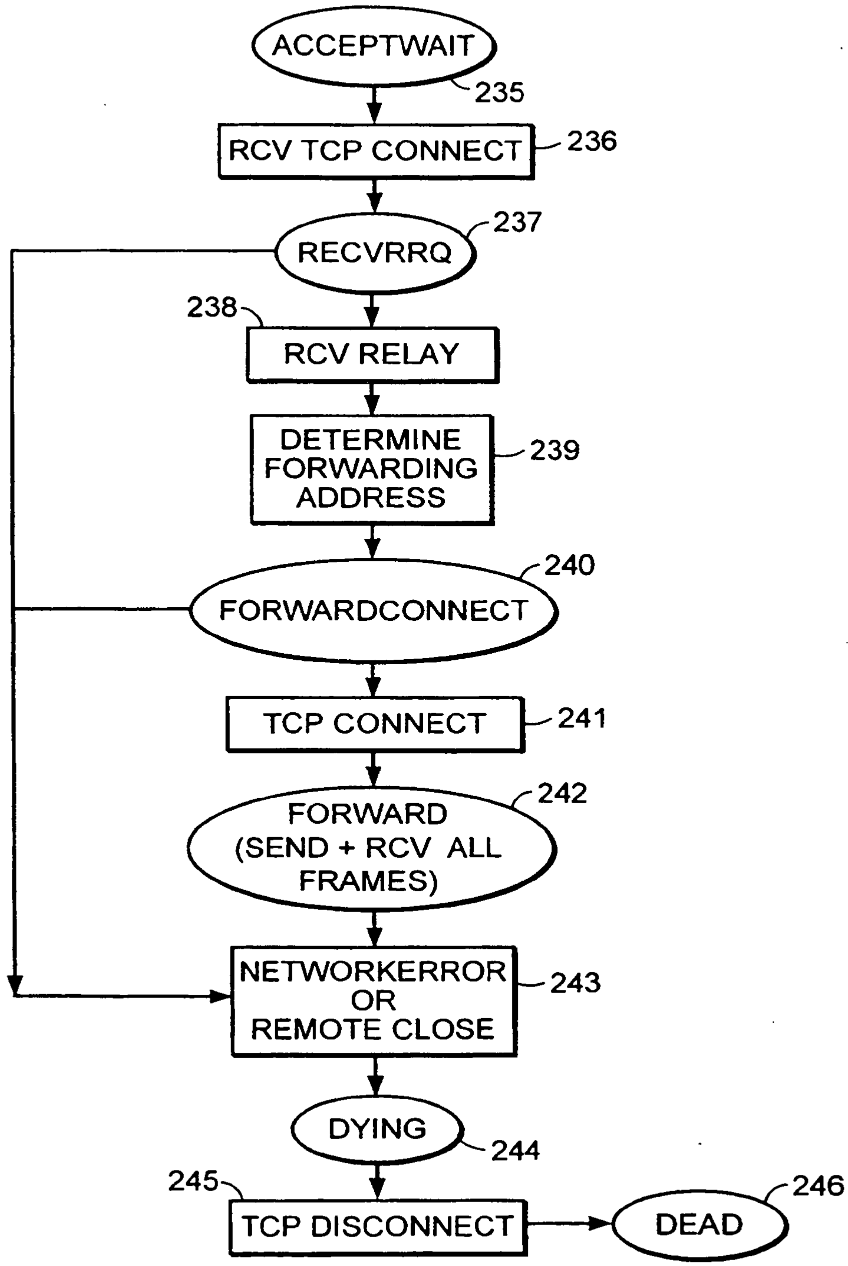

ACCEPTWAIT
235
RCV TCP CONNECT
236
RECVRRQ
237
238
RCV RELAY
DETERMINE FORWARDING ADDRESS
239
FORWARDCONNECT
240
TCP CONNECT
241
FORWARD (SEND + RCV ALL FRAMES)
242
NETWORKERROR OR REMOTE CLOSE
243
DYING
244
245
TCP DISCONNECT
DEAD
246

FIG. 13

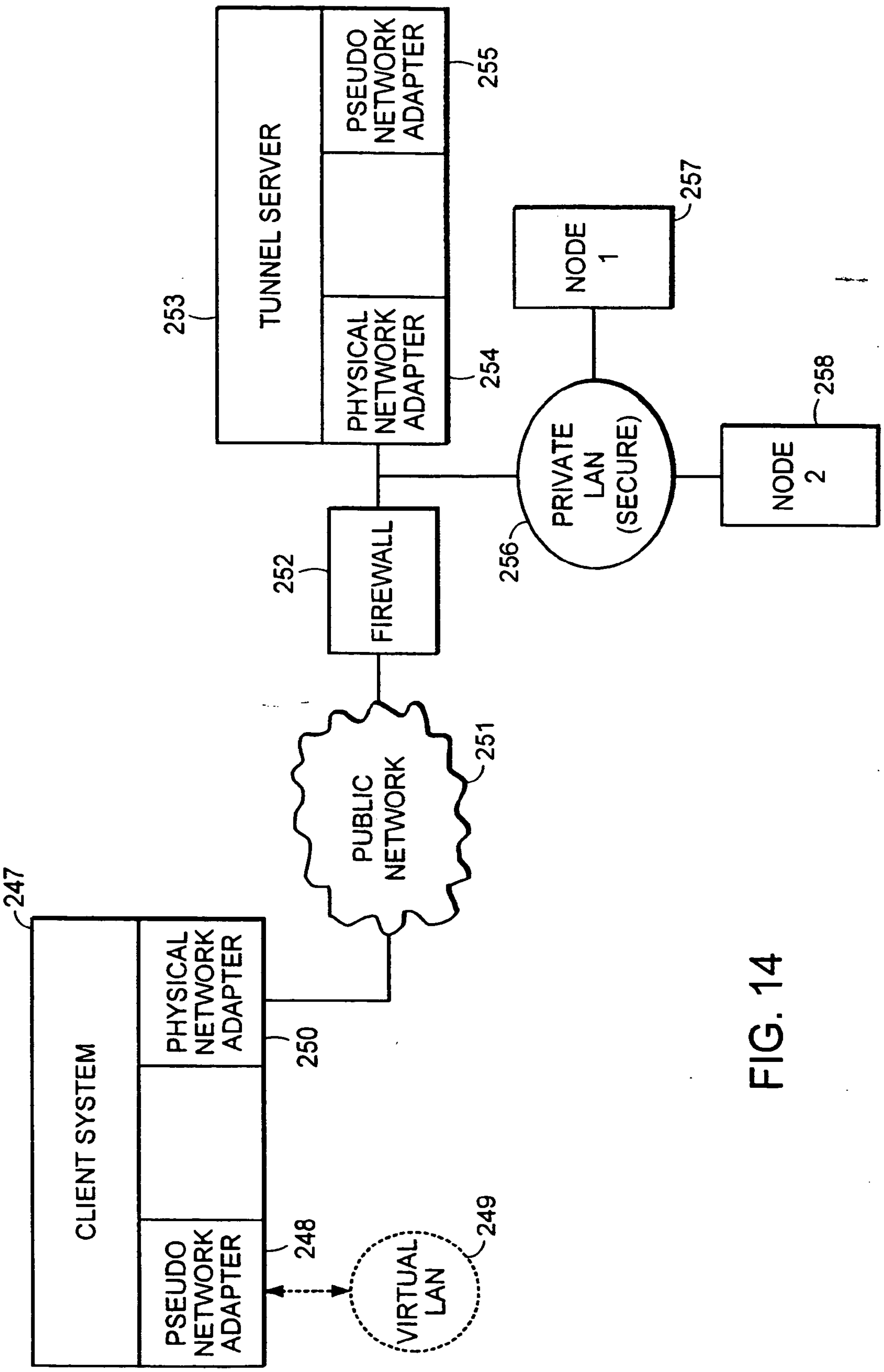

CLIENT SYSTEM
247
PSEUDO NETWORK ADAPTER
248
PHYSICAL NETWORK ADAPTER
250
VIRTUAL LAN
249
PUBLIC NETWORK
251
FIREWALL
252
TUNNEL SERVER
253
PHYSICAL NETWORK ADAPTER
254
PSEUDO NETWORK ADAPTER
255
PRIVATE LAN (SECURE)
256
NODE 1
257
NODE 2
258

FIG. 14

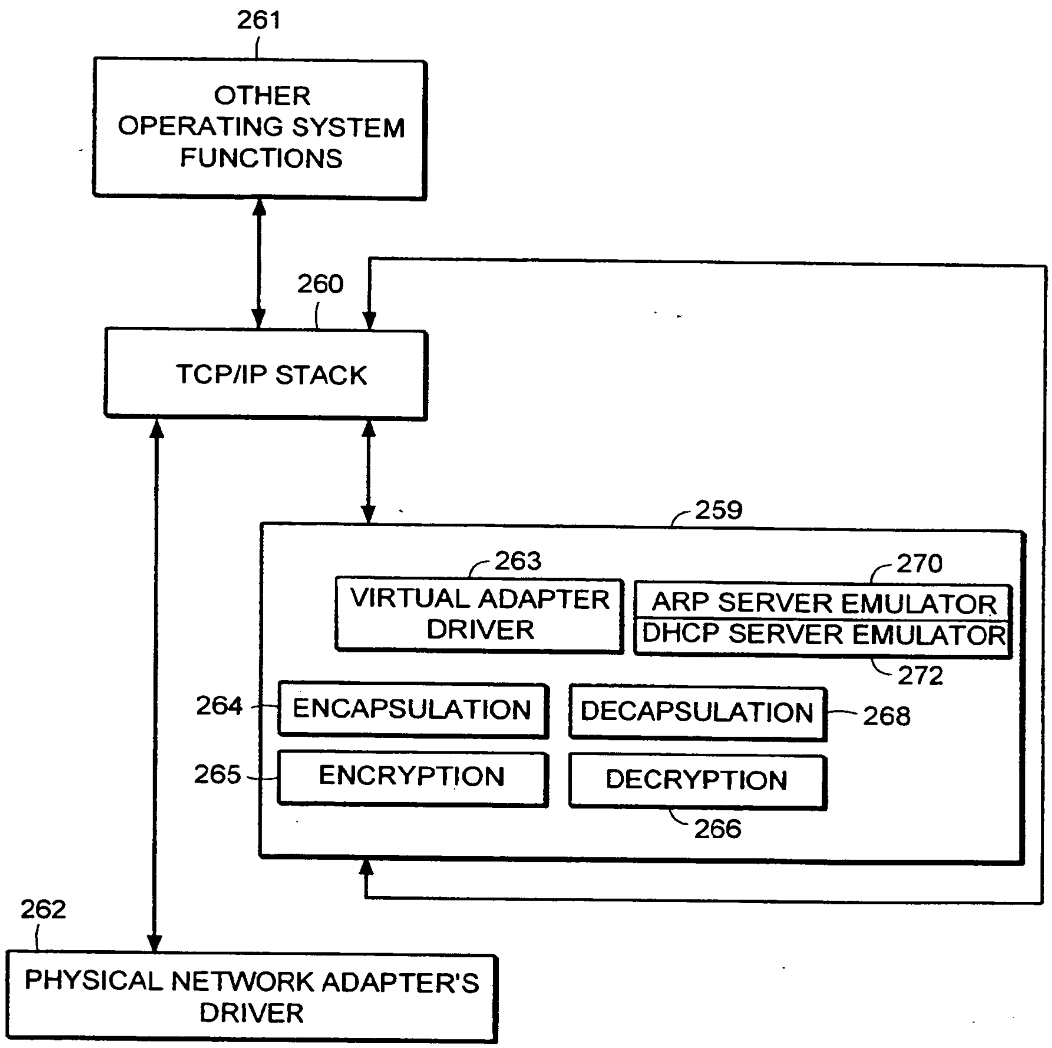

261
OTHER OPERATING SYSTEM FUNCTIONS
260
TCP/IP STACK
259
263
VIRTUAL ADAPTER DRIVER
270
ARP SERVER EMULATOR
DHCP SERVER EMULATOR
272
264
ENCAPSULATION
DECAPSULATION
268
265
ENCRYPTION
DECRYPTION
266
262
PHYSICAL NETWORK ADAPTER'S DRIVER

FIG. 15

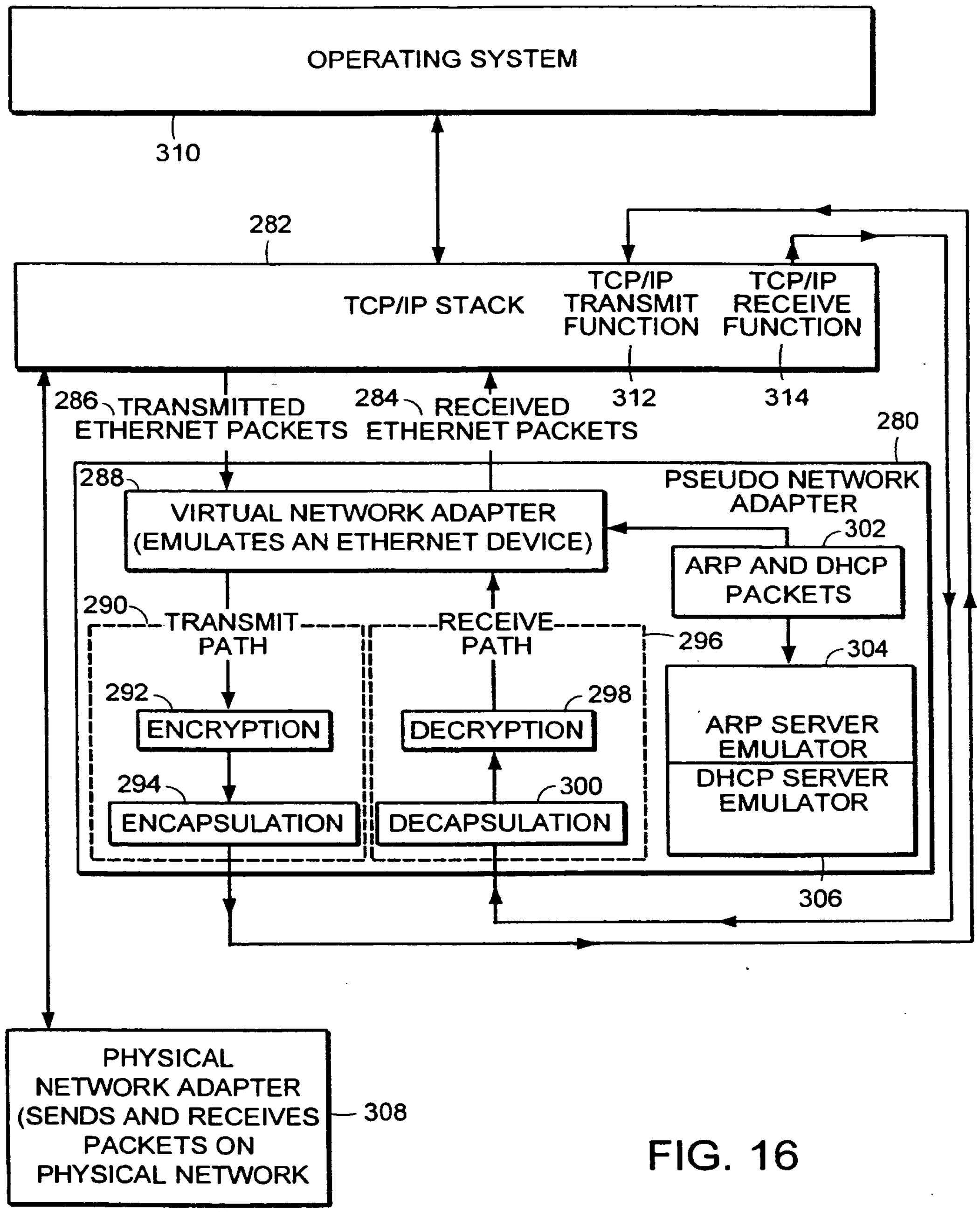

OPERATING SYSTEM
310
282
TCP/IP STACK
TCP/IP TRANSMIT FUNCTION
TCP/IP RECEIVE FUNCTION
312
314
286 TRANSMITTED ETHERNET PACKETS
284 RECEIVED ETHERNET PACKETS
280
288
PSEUDO NETWORK ADAPTER
VIRTUAL NETWORK ADAPTER (EMULATES AN ETHERNET DEVICE)
302
ARP AND DHCP PACKETS
290
TRANSMIT PATH
RECEIVE PATH
296
304
292
ENCRYPTION
298
DECRYPTION
ARP SERVER EMULATOR
DHCP SERVER EMULATOR
294
ENCAPSULATION
300
DECAPSULATION
306
PHYSICAL NETWORK ADAPTER (SENDS AND RECEIVES PACKETS ON PHYSICAL NETWORK
308

FIG. 16

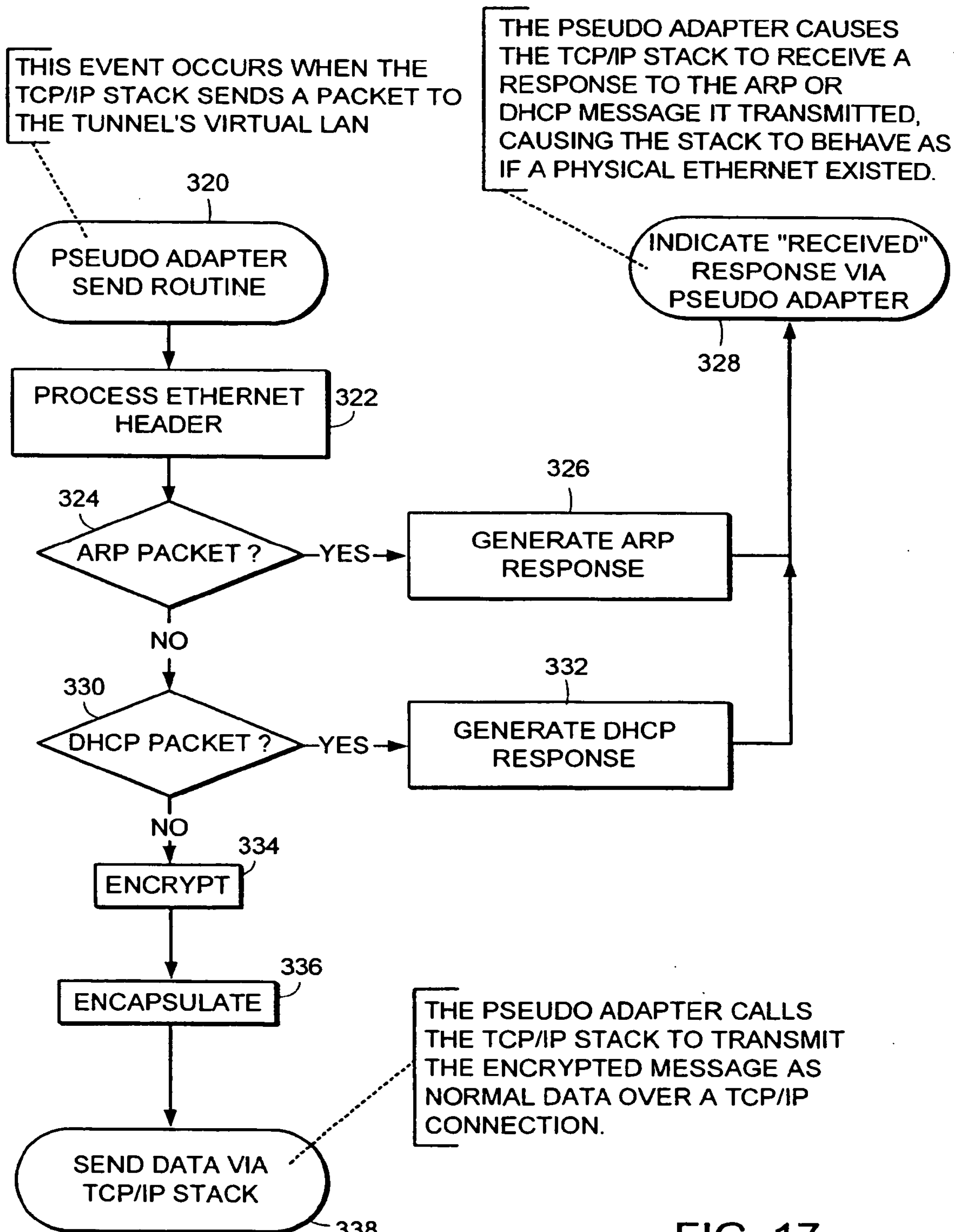

THIS EVENT OCCURS WHEN THE TCP/IP STACK SENDS A PACKET TO THE TUNNEL'S VIRTUAL LAN
THE PSEUDO ADAPTER CAUSES THE TCP/IP STACK TO RECEIVE A RESPONSE TO THE ARP OR DHCP MESSAGE IT TRANSMITTED, CAUSING THE STACK TO BEHAVE AS IF A PHYSICAL ETHERNET EXISTED.
320
PSEUDO ADAPTER SEND ROUTINE
INDICATE "RECEIVED" RESPONSE VIA PSEUDO ADAPTER
328
PROCESS ETHERNET HEADER
322
326
324
ARP PACKET ?
YES
GENERATE ARP RESPONSE
NO
332
330
DHCP PACKET ?
YES
GENERATE DHCP RESPONSE
NO
334
ENCRYPT
336
ENCAPSULATE
THE PSEUDO ADAPTER CALLS THE TCP/IP STACK TO TRANSMIT THE ENCRYPTED MESSAGE AS NORMAL DATA OVER A TCP/IP CONNECTION.
SEND DATA VIA TCP/IP STACK
338

FIG. 17

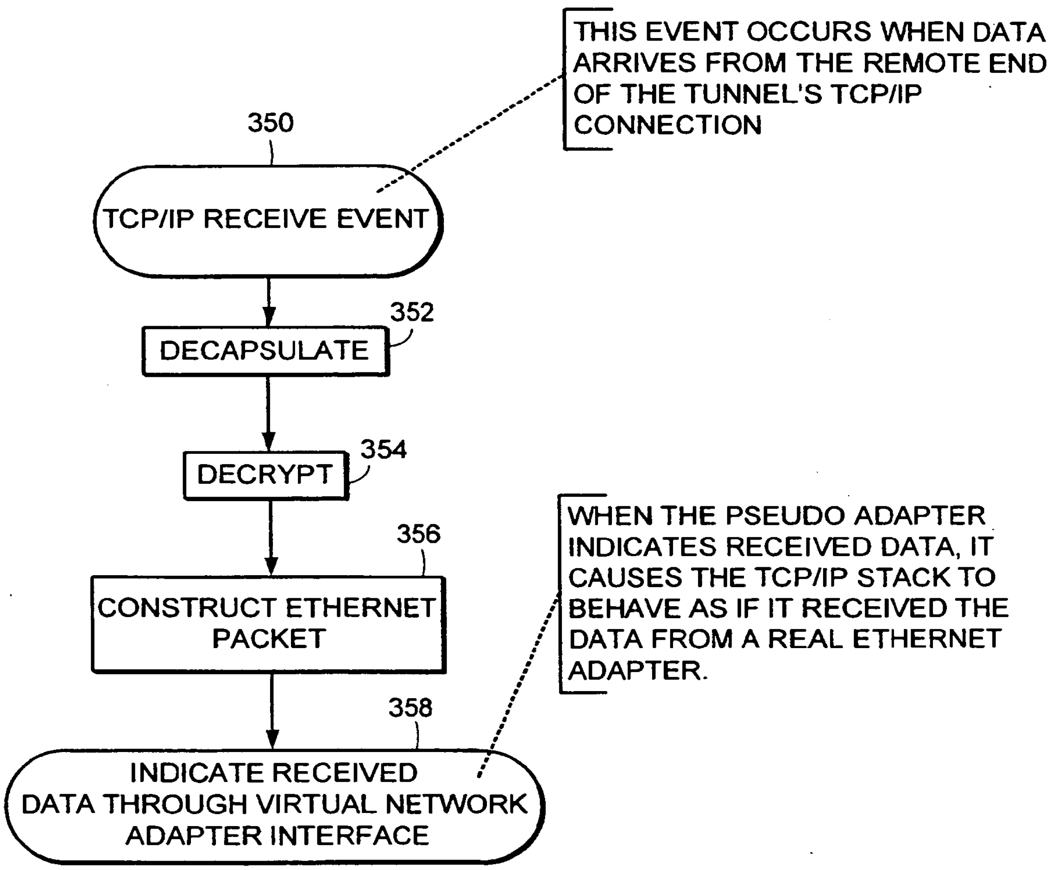

THIS EVENT OCCURS WHEN DATA ARRIVES FROM THE REMOTE END OF THE TUNNEL'S TCP/IP CONNECTION
350
TCP/IP RECEIVE EVENT
352
DECAPSULATE
354
DECRYPT
356
CONSTRUCT ETHERNET PACKET
WHEN THE PSEUDO ADAPTER INDICATES RECEIVED DATA, IT CAUSES THE TCP/IP STACK TO BEHAVE AS IF IT RECEIVED THE DATA FROM A REAL ETHERNET ADAPTER.
358
INDICATE RECEIVED DATA THROUGH VIRTUAL NETWORK ADAPTER INTERFACE

FIG. 18

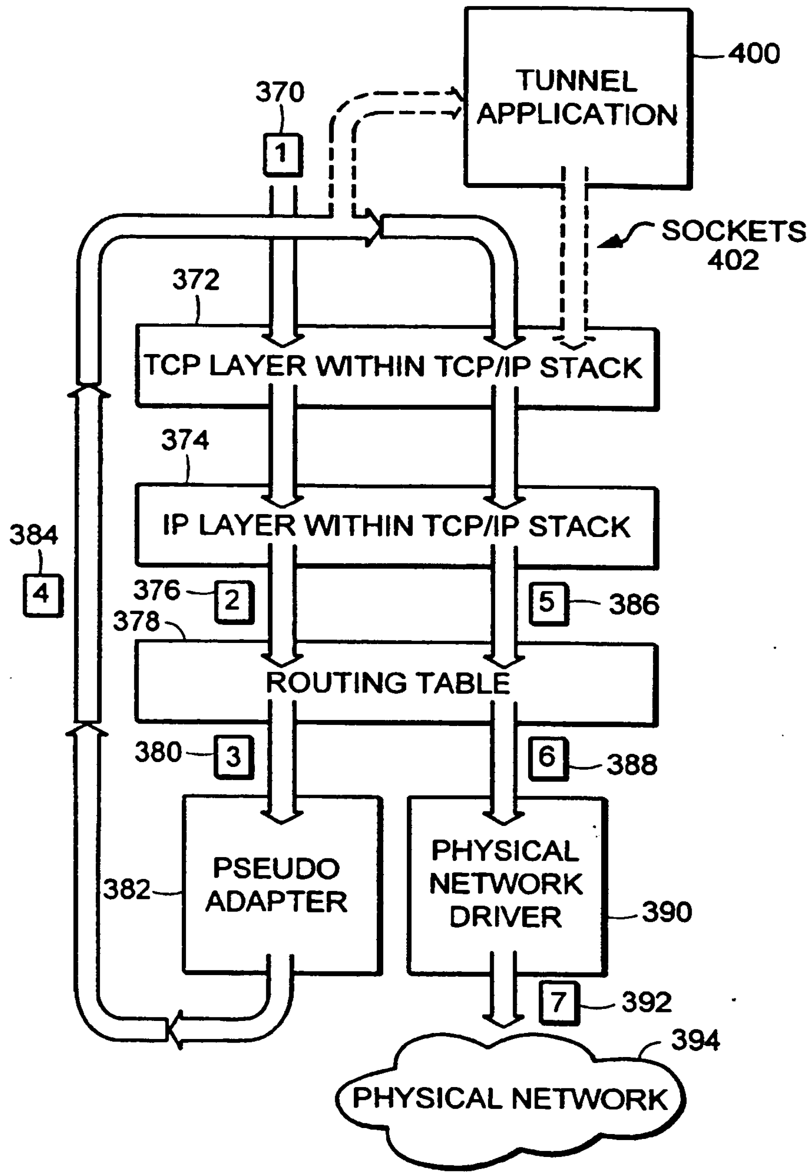

TUNNEL APPLICATION
400
370
1
SOCKETS 402
372
TCP LAYER WITHIN TCP/IP STACK
374
IP LAYER WITHIN TCP/IP STACK
384
4
376
2
378
5
386
ROUTING TABLE
380
3
6
388
PSEUDO ADAPTER
382
PHYSICAL NETWORK DRIVER
390
7
392
394
PHYSICAL NETWORK

FIG. 19

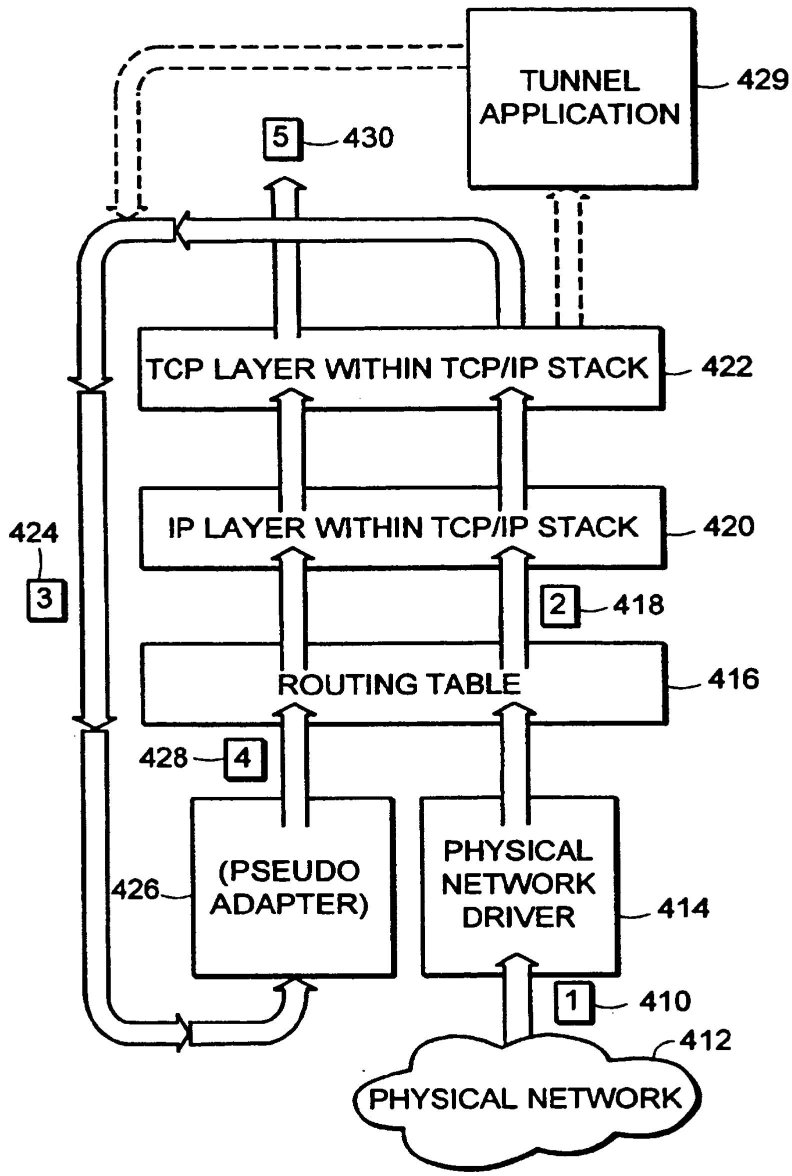

TUNNEL APPLICATION
429
5
430
TCP LAYER WITHIN TCP/IP STACK
422
424
3
IP LAYER WITHIN TCP/IP STACK
420
2
418
ROUTING TABLE
416
428
4
426
(PSEUDO ADAPTER)
PHYSICAL NETWORK DRIVER
414
1
410
412
PHYSICAL NETWORK

FIG. 20

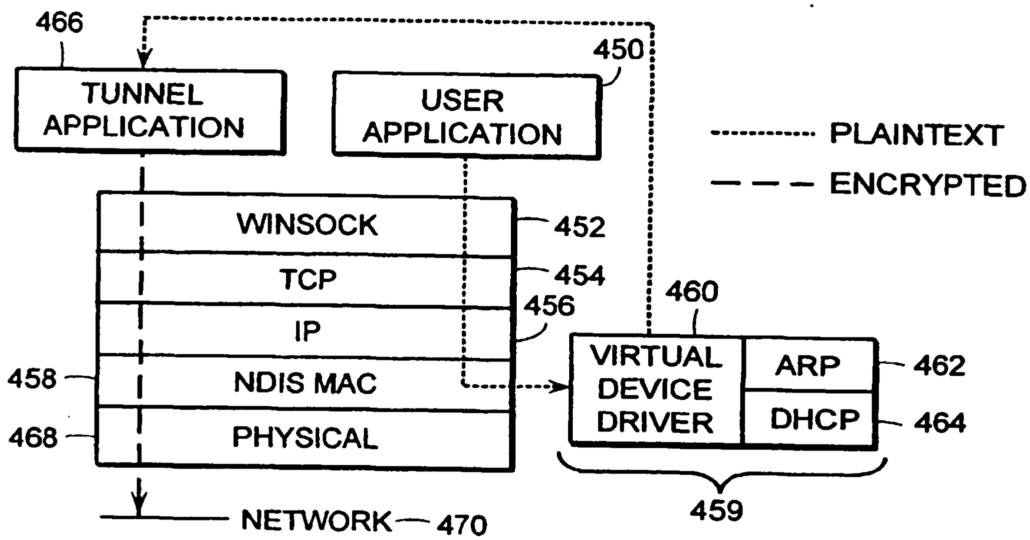

466
TUNNEL APPLICATION
USER APPLICATION
450
PLAINTEXT
ENCRYPTED
WINSOCK
452
TCP
454
IP
456
460
458
NDIS MAC
VIRTUAL DEVICE DRIVER
ARP
462
468
PHYSICAL
DHCP
464
NETWORK
470
459

FIG. 21

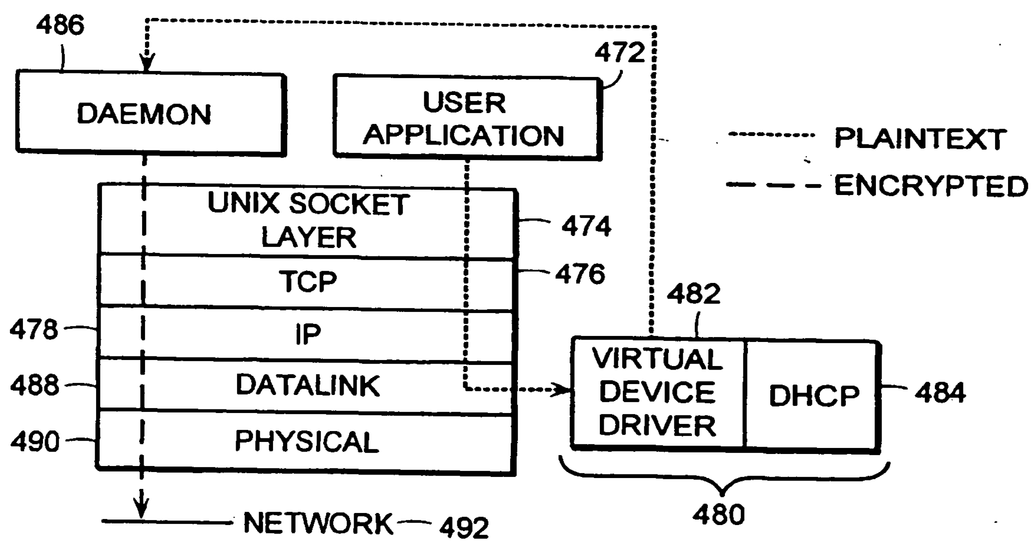

486
DAEMON
USER APPLICATION
472
PLAINTEXT
ENCRYPTED
UNIX SOCKET LAYER
474
TCP
476
482
478
IP
488
DATALINK
VIRTUAL DEVICE DRIVER
DHCP
484
490
PHYSICAL
NETWORK
492
480

FIG. 22

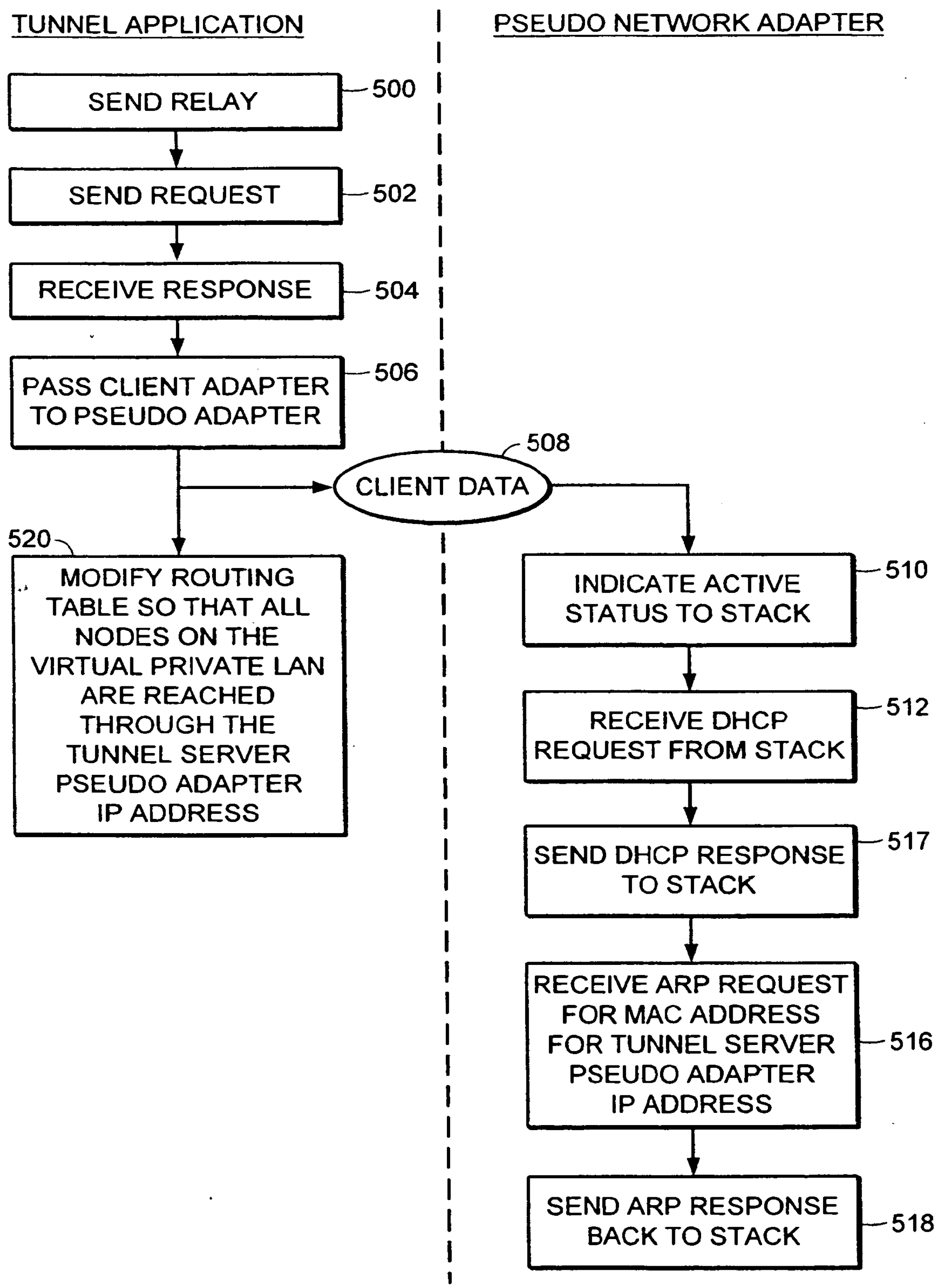

TUNNEL APPLICATION
PSEUDO NETWORK ADAPTER
SEND RELAY
500
SEND REQUEST
502
RECEIVE RESPONSE
504
PASS CLIENT ADAPTER TO PSEUDO ADAPTER
506
CLIENT DATA
508
520
MODIFY ROUTING TABLE SO THAT ALL NODES ON THE VIRTUAL PRIVATE LAN ARE REACHED THROUGH THE TUNNEL SERVER PSEUDO ADAPTER IP ADDRESS
INDICATE ACTIVE STATUS TO STACK
510
RECEIVE DHCP REQUEST FROM STACK
512
SEND DHCP RESPONSE TO STACK
517
RECEIVE ARP REQUEST FOR MAC ADDRESS FOR TUNNEL SERVER PSEUDO ADAPTER IP ADDRESS
516
SEND ARP RESPONSE BACK TO STACK
518

FIG. 23

10/082,285

(12) UK Patent Application (19) GB (11) 2 317 792 (13) A

(43) Date of A Publication 01.04.1998

(21) Application No 9719816.2

(22) Date of Filing 17.09.1997

(30) Priority Data
(31) 08715343 (32) 18.09.1996 (33) US
08715668 18.09.1996

(71) Applicant(s)
Secure Computing Corporation

(Incorporated in USA - Delaware)

2675 Long Lake Road, Roseville, Minnesota 55113-2536, United States of America

(72) Inventor(s)
Spence Minear
Edward B Stockwell
Troy De Jongh

(74) Agent and/or Address for Service
Beresford & Co
2-5 Warwick Court, High Holborn, LONDON, WC1R 5DJ, United Kingdom

(51) INT CL⁶
H04L 9/00

(52) UK CL (Edition P)
H4P PPEB
U1S S2124 S2209

(56) Documents Cited
WO 97/26735 A1 WO 97/26734 A1 WO 97/26731 A1
WO 97/23972 A1 WO 97/13340 A1

(58) Field of Search
UK CL (Edition P) **H4P PDCSA PDCSC PPEB**
INT CL⁶ **H04L 9/00 9/32 29/06 29/08**
Online: WPI, INSPEC

(54) Virtual Private Network for encrypted firewall

(57) A system (10) for regulating the flow of messages through a firewall (18) having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer where if the message is not encrypted, it passes the unencrypted message up the network protocol stack to an application level proxy (50), and if the message is encrypted, it decrypts the message and passes the decrypted message up the network protocol stack to the application level proxy. The step of decrypting the message includes the step of executing a process at the IP layer to decrypt the message.

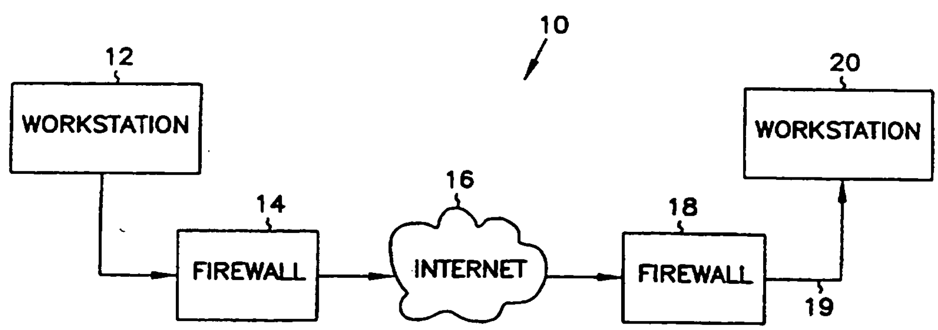


FIG. 1

GB 2 317 792 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

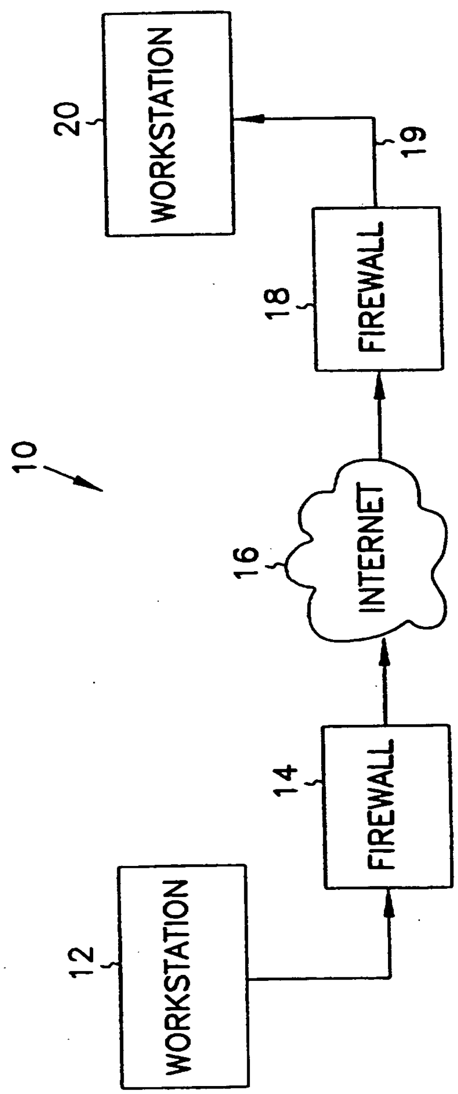
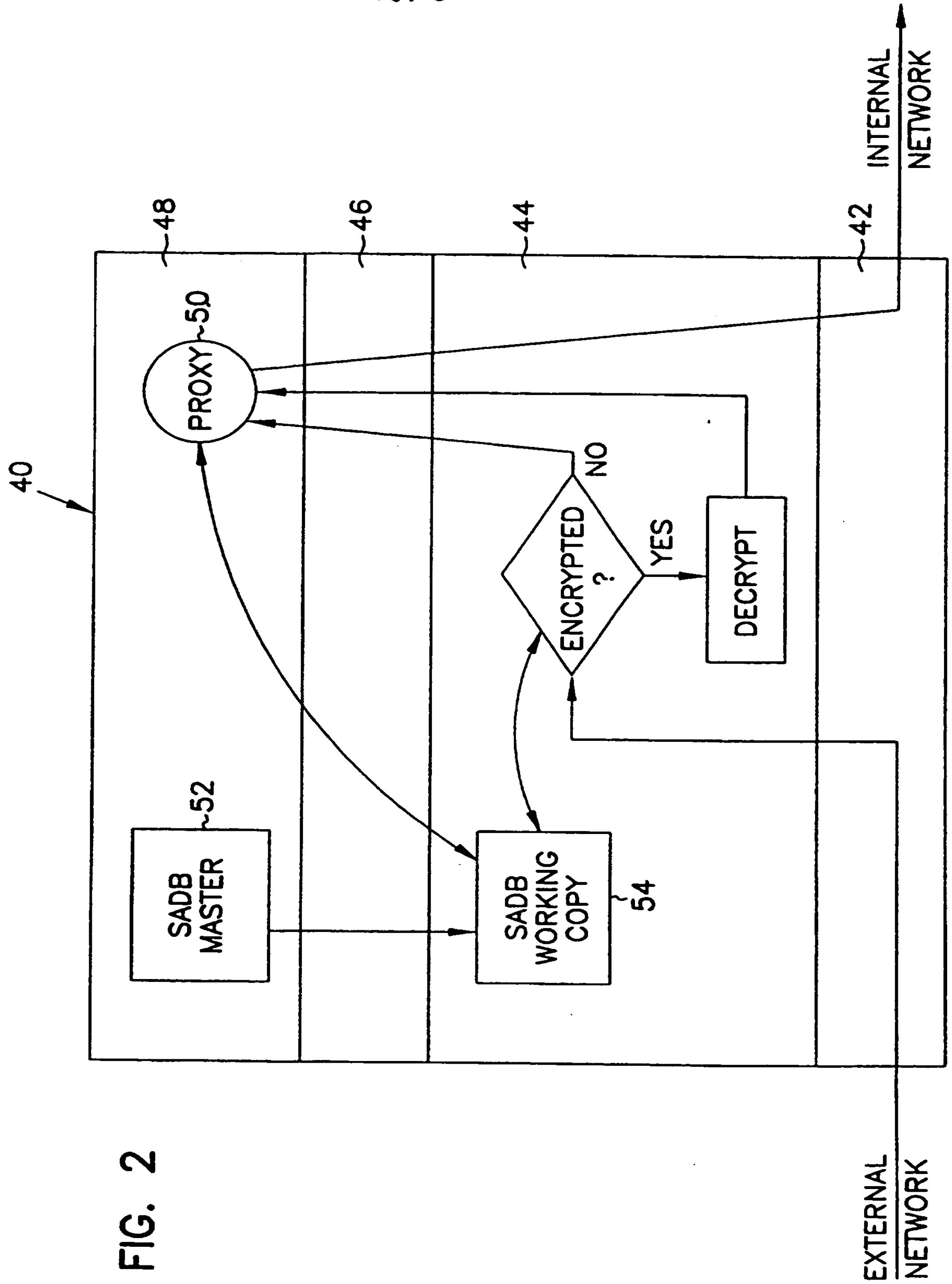

10
12
WORKSTATION
14
FIREWALL
16
INTERNET
18
FIREWALL
19
20
WORKSTATION

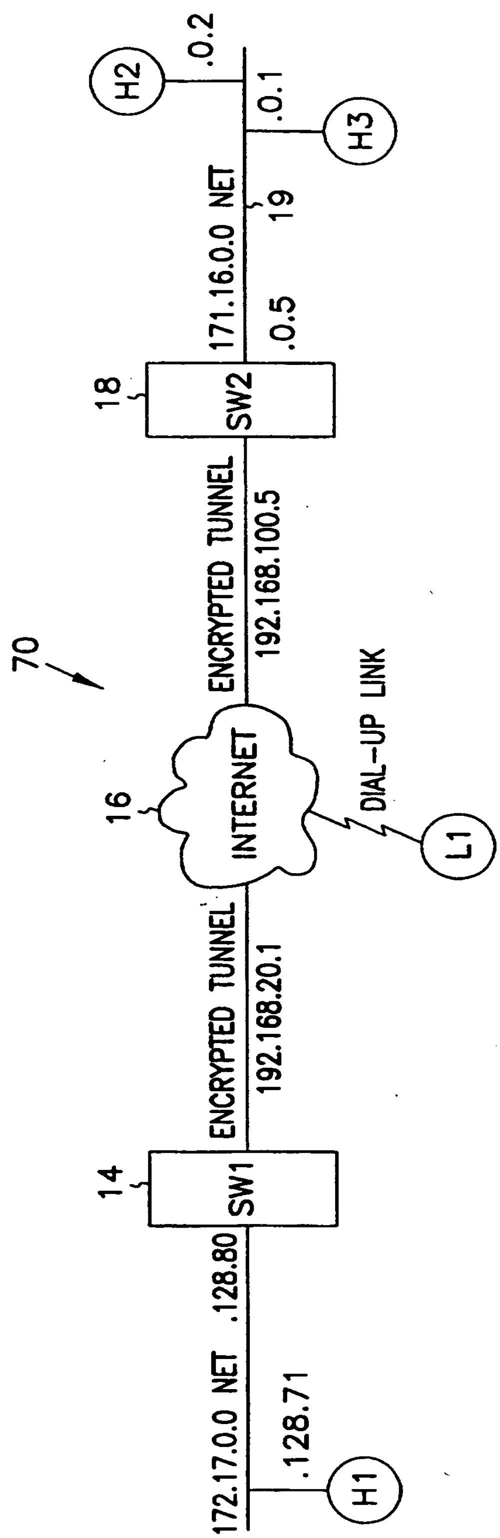
FIG. 1



40
48
46
44
42
PROXY
50
ENCRYPTED ?
NO
YES
DECRYPT
SADB MASTER
52
SADB WORKING COPY
54
INTERNAL NETWORK
EXTERNAL NETWORK

FIG. 2

3/5



70
H2
.0.2
.0.1
H3
171.16.0.0 NET
19
.0.5
18
SW2
ENCRYPTED TUNNEL
192.168.100.5
DIAL-UP LINK
16
INTERNET
L1
ENCRYPTED TUNNEL
192.168.20.1
14
SW1
.128.80
172.17.0.0 NET
.128.71
H1

FIG. 3

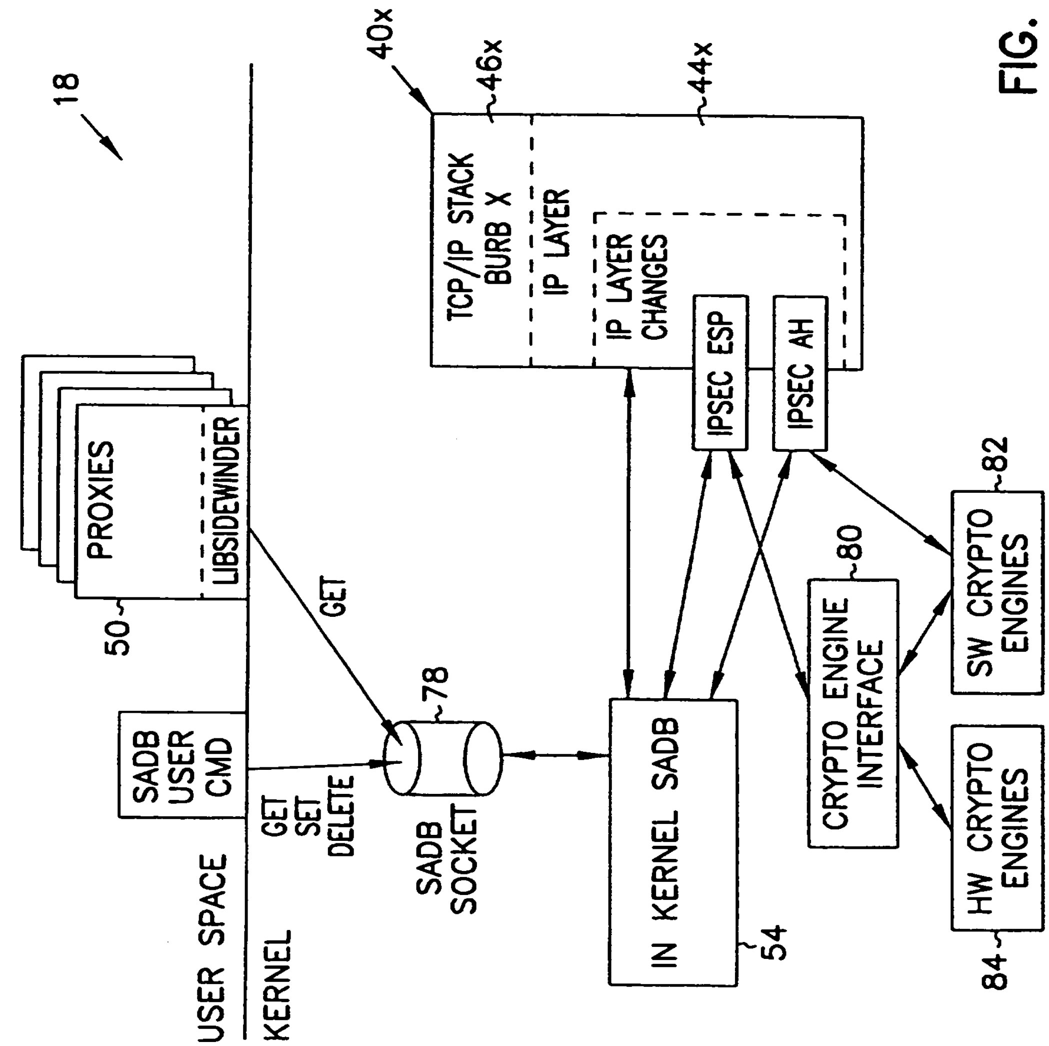

18
40x
46x
44x
TCP/IP STACK
BURB X
IP LAYER
IP LAYER
CHANGES
IPSEC ESP
IPSEC AH
PROXIES
LIBSIDEWINDER
50
USER SPACE
KERNEL
SADB
USER
CMD
GET
SET
DELETE
GET
78
SADB
SOCKET
IN KERNEL SADB
54
CRYPTO ENGINE
INTERFACE
80
SW CRYPTO
ENGINES
82
HW CRYPTO
ENGINES
84

FIG. 4

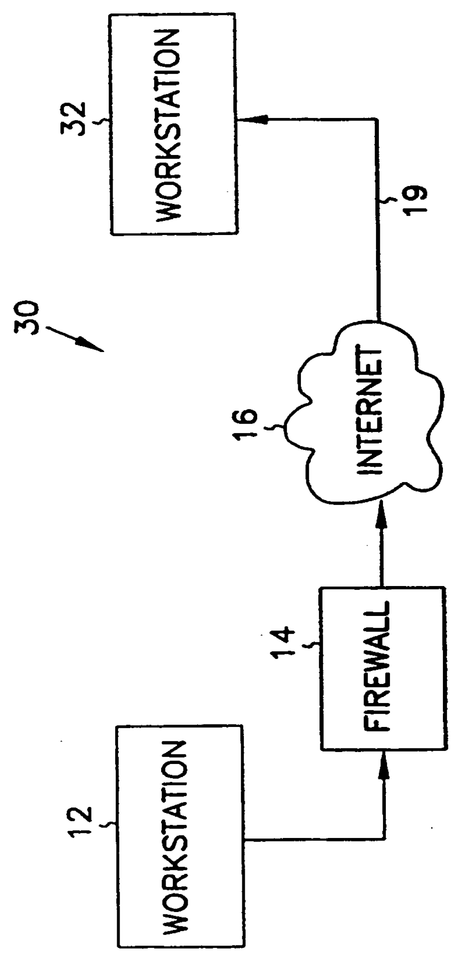

30
12
WORKSTATION
14
FIREWALL
16
INTERNET
19
32
WORKSTATION

FIG. 5

2317792

VIRTUAL PRIVATE NETWORK ON APPLICATION GATEWAY

Background of the Invention

Field of the Invention

The present invention pertains generally to network communications, and in particular to a system and method for securely transferring information between firewalls over an unprotected network.

Background Information

Firewalls have become an increasingly important part of network design. Firewalls provide protection of valuable resources on a private network while allowing communication and access with systems located on an unprotected network such as the Internet. In addition, they operate to block attacks on a private network arriving from the unprotected network by providing a single connection with limited services. A well designed firewall limits the security problems of an Internet connection to a single firewall computer system. This allows an organization to focus their network security efforts on the definition of the security policy enforced by the firewall. An example of a firewall is given in "SYSTEM AND METHOD FOR PROVIDING SECURE INTERNETWORK SERVICES" by Boebert et al. (PCT Published Application No. WO 96/13113, published on May 2, 1996), the description of which is hereby incorporated by reference. Another description of a firewall is provided by Dan Thomsen in "Type Enforcement: the new security model", *Proceedings: Multimedia: Full-Service Impact on Business, Education, and the Home*, SPIE Vol. 2617, p. 143, August 1996. Yet another such system is described in "SYSTEM AND METHOD FOR ACHIEVING NETWORK SEPARATION" by Gooderum et al. (PCT Published Application No. WO 97/29413, published on August 14, 1997), the description of which is hereby incorporated by reference. All the above systems are examples of application level gateways. Application level gateways use proxies or other such mechanisms operating at the application layer to process traffic through the firewall. As such, they can review not only the message traffic but also message content. In addition, they provide authentication and identification services, access control and auditing.

Data to be transferred on unprotected networks like the Internet is susceptible to electronic eavesdropping and accidental (or deliberate) corruption. Although a firewall can protect data within a private network from attacks launched from the unprotected network, even that data is vulnerable to both eavesdropping and corruption when transferred from the private network to an external machine. To address this danger, the Internet Engineering Task Force (IETF) developed a standard for protecting data transferred between firewalls over an unprotected network. The Internet Protocol Security (IPSEC) standard calls for encrypting data before it leaves the first firewall, and then decrypting the data when it is received by the second firewall. The decrypted data is then delivered to its destination, usually a user workstation connected to the second firewall. For this reason IPSEC encryption is sometimes called *firewall-to-firewall encryption* (FFE) and the connection between a workstation connected to the first firewall and a client or server connected to the second firewall is termed a *virtual private network*, or VPN.

The two main components of IPSEC security are data encryption and sender authentication. Data encryption increases the cost and time required for the eavesdropping party to read the transmitted data. Sender authentication ensures that the destination system can verify whether or not the encrypted data was actually sent from the workstation that it was supposed to be sent from. The IPSEC standard defines an encapsulated payload (ESP) as the mechanism used to transfer encrypted data. The standard defines an authentication header (AH) as the mechanism for establishing the sending workstation's identity.

Through the proper use of encryption, the problems of eavesdropping and corruption can be avoided; in effect, a protected connection is established from the internal network connected to one firewall through to an internal network connected to the second firewall. In addition, IPSEC can be used to provide a protected connection to an external computing system such as a portable personal computer.

IPSEC encryption and decryption work within the IP layer of the network protocol stack. This means that all communication between two IP addresses will be protected because all interfirewall communication must go through the IP layer. Such an approach is preferable over encryption and decryption at higher levels in the network protocol stack since when encryption is performed at layers higher than the IP layer more work is required to ensure that all supported communication is properly protected. In addition, since IPSEC encryption is handled below the Transport layer, IPSEC can encrypt data sent by any application. IPSEC therefore becomes a transparent add-on to such protocols as TCP and UDP.

Since, however, IPSEC decryption occurs at the IP layer, it can be difficult to port IPSEC to an application level gateway while still maintaining control at the proxy over authentication, message content, access control and auditing. Although the IPSEC specification in RFC 1825 suggests the use of a mandatory access control mechanism in a multi-level secure (MLS) network to compare a security level associated with the message with the security level of the receiving process, such an approach provides only limited utility in an application level gateway environment. In fact, implementations on application level gateways to date have simply relied on the fact that the message was IPSEC-encrypted as assurance that the message is legitimate and have simply decoded and forwarded the message to its destination. This creates, however, a potential chink in the firewall by assuming that the encrypted communication has access to all services.

What is needed is a method of handling IPSEC messages within an application level gateway which overcomes the above deficiencies. The method should allow control over access by an IPSEC connection to individual services within the internal network.

Summary of the Invention

The present invention is a system and method for regulating the flow of messages through a firewall having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of determining, at the IP layer, if a message is encrypted, if the message is not encrypted, passing the unencrypted message up the network protocol stack to an application level proxy, and if the message is encrypted, decrypting the message and passing the decrypted message up the network protocol stack to the application level proxy, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message.

According to another aspect of the present invention, a system and method is described for authenticating the sender of a message within a computer system having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of determining, at the IP layer, if the message is encrypted, if the message is encrypted, decrypting the message, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message, passing the decrypted message up the network protocol stack to an application level proxy, determining an authentication protocol appropriate for the message, and executing the authentication protocol to authenticate the sender of the message.

Brief Description of the Drawings

In the following detailed description of example embodiments of the invention, reference is made to the accompanying drawings which form a part hereof, and which is shown by way of illustration only, specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

In the drawings, where like numerals refer to like components throughout the several views:

Figure 1 is a functional block diagram of an application level gateway-implemented firewall-to-firewall encryption scheme according to the present invention;

Figure 2 is a block diagram showing access control checking of both encrypted and unencrypted messages in network protocol stack according to the present invention;

Figure 3 is a block diagram of a representative application level gateway-implemented firewall-to-firewall encryption scheme;

Figure 4 is a block diagram of one embodiment of a network-separated protocol stack implementing IPSEC according to the present invention; and

Figure 5 is a functional block diagram of a firewall-to-workstation encryption scheme according to the present invention.

Description of the Preferred Embodiments

In the following detailed description of the preferred embodiment, references made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific preferred embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that structural, logical, physical, architectural, and electrical changes may be made without departing from the spirit and scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims and their equivalents.

A system 10 which can be used for firewall-to-firewall encryption (FFE) is shown in Figure 1. In Figure 1, system 10 includes a workstation 12 communicating through a firewall 14 to an unprotected network 16 such as the Internet. System 10 also includes a workstation 20 communicating through a firewall 18 to unprotected network 16. In one embodiment, firewall 18 is an application level gateway.

As noted above, IPSEC encryption and decryption work within the IP layer of the network protocol stack. This means that all communications between two IP addresses will be protected because all interfirewall communication must pass through the IP layer. IPSEC takes the standard

Internet packet and converts it into a carrier packet. The carrier packet is designed to do two things: to conceal the contents of the original packet (encryption) and to provide a mechanism by which the receiving firewall can verify the source of the packet (authentication). In one embodiment of the present invention, each IPSEC carrier packet includes both an authentication header used to authenticate the sending machine and an encapsulated payload containing encrypted data. The authentication header and the encapsulated payload features of IPSEC can, however, be used independently. As required in RFC 1825, DES-CBC is provided for use in encrypting the encapsulated payload while the authentication header uses keyed MD5.

To use IPSEC, you must create a *security association* (SA) for each destination IP address. In one embodiment, each SA contains the following information:

- Security Parameters Index (SPI) - The index used to find a SA on receipt of an IPSEC datagram.
- Destination IP address - The address used to find the SA and trigger use of IPSEC processing on output.
- The peer SPI - The SPI value to put on a IPSEC datagram on output.
- The peer IP address - The destination IP address to be put into the packet header if IPSEC Tunnel mode is used.
- The Encryption Security Payload (ESP) algorithm to be used.
- The ESP key to used for decryption of input datagrams.
- The ESP key to used for encryption of output datagrams.
- The authentication (AH) algorithm to be used.
- The AH key to be used for validation of input packets.
- The AH key to be used for generation of the authentication data for output datagrams.

The combination of a given Security Parameter Index and Destination IP address uniquely identifies a particular "Security Association." In one embodiment, the sending firewall uses the sending userid and Destination Address to select an appropriate Security Association (and hence SPI value). The receiving firewall uses the combination of SPI value and Source address to obtain the appropriate Security Association.

A security association is normally one-way. An authenticated communications session between two firewalls will normally have two Security Parameter Indexes in use (one in each direction). The combination of a particular Security Parameter Index and a particular Destination Address uniquely identifies the Security Association.

More information on the specifics of an IPSEC FFE implementation can be obtained from the standards developed by the IPSEC work group and documented in *Security Architecture for IP* (RFC 1825) and in RFC's 1826-1829.

When a datagram is received from unprotected network 16 or is to be transmitted to a destination across unprotected network 16, the firewall must be able to determine the algorithms, keys, etc. that must be used to process the datagram correctly. In one embodiment, this information is obtained via a security association lookup. In one such embodiment, the lookup routine is passed several arguments: the source IP address if the datagram is being received from network 16 or the destination IP address if the datagram is to be transmitted across network 16, the SPI, and a flag that is used to indicate whether the lookup is being done to receive or transmit a datagram.

When an IPSEC datagram is received by firewall 18 from unprotected network 16, the SPI and source IP address are determined by looking in the datagram. In one embodiment a Security Association Database (SADB) stored within firewall 18 is searched for the entry with a matching SPI. In one such embodiment, security associations can be set up based on network address as well as a more granular host address. This allows the network administrator to create a security association between two firewalls with only a couple of lines in a configuration file on each machine. For such embodiments, the entry in the Security Association Database that has both the matching SPI and the longest address match is selected as the SA entry. In another such embodiment, each SA has a prefix length value associated with the address. An address match on a SA entry means that the addresses match for the number of bits specified by the prefix length value.

There are two exceptions to this search process. First, when an SA entry is set marked as being dynamic it implies that the user of this SA may not have a fixed IP address. In this case the match is fully determined by the SPI value. Thus it is necessary that the SPI values for such SA entries be unique in the SADB. The second exception is for SA entries marked as tunnel mode entries. In this case it is normally the case that the sending entity will hide its source address so that all that is visible on the public wire is the destination address. In this case, like in the case where the SA entries are for dynamic IP addresses, the search is done exclusively on the basis of the SPI.

When transmitting a datagram across unprotected network 16 the SADB is searched using only the destination address as an input. In this case the entry which has the longest address match is selected and returned to the calling routine.

In one embodiment, if firewall 18 receives datagrams which are identified as either an IP_PROTO_IPSEC_ESP or IP_PROTO_IPSEC_AH protocol datagram, there must be a corresponding SA in the SADB or else firewall 18 will drop the packet and an audit message will be generated. Such an occurrence might indicate a possible attack or it might simply be a symptom of an erroneous key entry in the Security Association Database.

In a system such as system 10, application level gateway firewall 18 acts as a buffer between unprotected network 16 and workstations such as workstation 20. Messages coming from unprotected network 16 are reviewed and a determination is made as to whether execution of an authentication and identification protocol is warranted. In contrast to previous systems, system 10 also performs this same determination on IPSEC-encrypted messages. If desired, the same authentication and identification can be made on messages to be transferred from workstation 20 to unprotected network 16. Figure 2 illustrates one way of authenticating both encrypted and unencrypted messages in a system such as system 10.

In the system of Figure 2 a network protocol stack 40 includes a physical layer 42, an Internet protocol (IP) layer 44, a Transport layer 46 and an application layer 48. Such a protocol stack exists, for instance on application level gateway firewall 18 of Figure 1. An application executing in application layer 48 can communicate to an application executing on another system by preparing a message and transmitting it through one of the existing transport services executing on transport layer 46. Transport layer 46 in turn uses a process executing in IP layer 44 to continue the transfer. Physical layer 42 provides the software needed to transfer data through the communication hardware (e.g., a network interface card or a modem). As noted above, IPSEC executes within IP layer 44. Encryption and authentication is transparent to the host as long as the network administrator has the Security Association Database correctly configured and a key management mechanism is in place on the firewall.

In application level gateway firewall 18, a proxy 50 operating within application layer 48 processes messages transferred between internal and external networks. All network-to-network traffic must pass through one of the proxies within application layer 48 before being the transfer across networks is allowed. A message arriving from external network 16 is examined at IP layer 44 and an SADB is queried to determine if the source address and SPI are associated with an SA. In the embodiment shown in Figure 2, an SADB Master copy 52 is maintained in persistent memory at application layer 48 while a copy 54 of SADB is maintained in volatile memory within the kernel. If the message is supposed to be encrypted, the message is decrypted based on the algorithm and key associated with the particular SA and the message is transferred up through transport layer 46 to proxy 50. Proxy 50 examines the source and destination addresses and the type of service desired and decides whether authentication of the sender is warranted. If so, proxy 50 initiates an authentication protocol. The protocol may be as simple as requesting a user name and password or it may include a challenge/response authentication process. Proxy 50 also looks to see whether the message coming in was encrypted or not and may factor that into whether a particular type of authentication is needed. In Telnet, for instance, user name/password authentication may be sufficient for an FFE link while the security policy may dictate that a more stringent challenge/response protocol is needed for unencrypted links. In that case, proxy 50 will be a Telnet proxy and it will base its authentication protocol on whether the link was encrypted or not.

Since IPSEC executes within IP layer 44 there is no need for host firewalls to update their applications. Users that already have IPSEC available on their own host machine will, however, have to request that the firewall administrator set up SA's in the SADB for their traffic.

In the embodiment shown in Figure 2, a working copy 54 of the Security Association Database consisting of all currently active SA's is kept resident in memory for ready access by IP layer processing as datagrams are received and transmitted. In addition, a working master copy 52 of the SADB is maintained in a file in nonvolatile memory. During system startup and initialization processing the content of all of the required SA's in master SADB 52 is added to the working copy 54 stored in kernel memory.

In one embodiment, firewall 18 maintains different levels of security on internal and external network interfaces. It is desirable for a firewall to have different levels of security on both the internal and external interfaces. In one embodiment, firewall 18 supports three different levels, numbered 0 through 2. These levels provide a simple policy mechanism that controls permission for both in-bound and out-bound packets.

- Level 0 - do not allow any in-bound or out-bound traffic unless there is a security association between the source and destination.

- Level 1 - Allow both in-bound and out-bound non-IPSEC traffic but force the use of IPSEC if a SA exists for the address. (To support this firewall 18 must look for a SA for each in-bound datagram.)
- Level 2 - allow NULL security associations to exist. NULL associations are just like normal security associations, except no encryption or authentication transform is performed on in-bound or out-bound packets that correspond to this NULL association. With Level 2 enabled, the machine will still receive unprotected traffic, but it will not transmit unless Level 1 is enabled.

The default protection level established when the Security Association Database (SADB) is initialized at boot time is 1 for in-bound traffic and 2 for out-bound traffic.

An Access Control List, or ACL, is a list of rules that regulate the flow of Internet connections through a firewall. These rules control how a firewall's servers and proxies will react to connection attempts. When a server or proxy receives an incoming connection, it performs an ACL check on that connection.

An ACL check compares a set of parameters associated with the connection against a list of ACL rules. The rules determine whether the connection is allowed or denied. A rule can also have one or more side effects. A side effect causes the proxy to change its behavior in some fashion. For example, a common side effect is to redirect the destination IP address to an alternate machine. In addition to IP connection attempts, ACL checks can also made on the console logins and on logins made from serial ports. Finally, ACL checks can also be made on behalf of IP access devices, such as a Cisco box, through the use of the industry standard TACACS+ protocol.

In one embodiment, the ACL is managed by an acld daemon running in the kernel of firewalls 10 and 30. The acld daemon receives two types of requests, one to query the ACL and one to administer it. In one such embodiment, the ACL is stored in a relational database such as the Oracle database for fast access. By using such a database, query execution is asynchronous and many queries can be executing concurrently. In addition, these types of databases are designed to manipulate long lists of rules quickly and efficiently. These qualities ensure that a given query cannot hang up the process that issued the query for any appreciable time (> 1-2 seconds).

In one such embodiment, the database can hold up to 100,000 users and up to 10,000 hosts but can be scaled up to the capacity of the underlying database engine. The results of an ACL check is cached, allowing repeated checks to be turned around very quickly.

Applications on firewalls 10 and 30 can query acld to determine if a given connection attempt should be allowed to succeed. In one embodiment, the types of applications (i.e. "agents") that can make ACL queries can be divided into four classes:

1) Proxies. These allow connections to pass through firewall 10 or 30 in order to provide access to a remote service. They include tnauthp (authenticated telnet proxy), pftp (FTP proxy), httpp (HTTP proxy), and tcpgsp (TCP generic service proxy).
2) Servers. These provide a service on the firewall itself. They include ftpd and httpd.
3) Login agents. Login agent is a program on the firewall that can create a Unix shell. It is not considered a server because it cannot receive IP connections. One example is /usr/bin/login when used to create a dialup session or a console session on firewall 10 or 30. Another example is the command *srole*.
4) Network Access Servers (NAS). NAS is a remote IP access device, typically a dialup box manufactured by such companies as Cisco or Bridge. The NAS usually provides dialup telnet service and may also provide SLIP or PPP service.

Proxies, servers, login agents, and NASes make queries to acld to determine if a given connection attempt should be allowed to succeed. All of the agents except NAS make their queries directly. NAS, because it is remote, must communicate via an auxiliary daemon that typically uses an industry standard protocol such as RADIUS or TACACS+. The auxiliary daemon (e.g., tacradd) in turn forwards the query to local acld.

As a side effect of the query, acld tells the agent if authentication is needed. If no authentication is needed, the connection proceeds immediately. Otherwise acld provides (as another side effect) a list of allowed authentication methods that the user can choose from. The agent can present a menu of choices or simply pick the first authentication method by default. Typical authentication methods include plain password, SNK DSS, SDI SecurID, LOCKout DES, and LOCKout FORTEZZA. In one embodiment, the list of allowed authentication methods varies depending on the host name, user name, time of day, or any combination thereof.

In the case of a Level 0 policy, it would be safe to assume that all incoming traffic is encrypted or authenticated. In the case of Levels 1 through 2, a determination must be made whether or not a security association exists for a given peer. Otherwise an application may believe that in-bound traffic has been authenticated when it really has not. (That is why it is necessary to look for an SA on input of each non-IPSEC datagram.)

In one embodiment, a flag which accompanies the message as it is sent from IP layer 44 to proxy 50 indicates whether the incoming message was or was not encrypted. In another embodiment, proxy 50 accesses Security Association Database 54 (the table in the kernel can be queried via an SADB routing socket (PF-SADB)) to determine whether or not a security association exists for a given peer.. The SADB socket is much like a routing socket found in the stock BSD 4.4 kernel (protocol family PF-ROUTE) except that PF-SADB sockets are used to maintain the Security Association Database (SADB) instead of the routing table. Because the private keys used for encryption, decryption, and keyed authentication are stored in this table, access must be strictly prohibited and allowed to only administrators and key management daemons. Care must be taken when allowing user-level daemons access to `/dev/mem` or `/dev/kmem` as well, since the keys are stored in kernel memory and could be exposed with some creative hacking.

In one embodiment, a command-line tool called `sadb` is used to support the generation and maintenance of in-kernel version 54 of SADB. The primary interface between this tool and the SADB is the PF-SADB socket. The kernel provides socket processing to receive client requests to add, update, or change entries in in-kernel SADB 54. As noted above, the default protection level established when the Security Association Database (SADB) is initialized at boot time is 1 for in-bound traffic and 2 for out-bound traffic. This may be changed by the use of the `sadb` command.

The existing `sadb` command was derived from the NIST implementation of IPSEC. As noted above, this tool is much like `route` in that it uses a special socket to pass data structures in and out of the kernel. There are three commands recognized by the `sadb` command: *get, set, delete*. The following simple shell script supports adding and removing a single SA entry to SADB 54. It shows one embodiment of a parameter order for adding a SA to the SADB.

```
# ! /bin/sh
if [ $# -ne 1 ]
then
      echo "usage: $0 <on>|<off>" >&2
      exit 1
fi
ONOFF=$1

addsa ()
{
IPADDRESS=$2
PEERADDRESS=0.0.0.0
PREFIXLEN=0                   # Num of bits, 0 => full 32
bit match
LOCALADDRESS=0.0.0.0
REALADDRESS=0.0.0.0
PORT=0
PROTOCOL=0
UID=0
DESALG=1                      # I = DES-CBC
IVLEN=4                       # bytes
DESKEY=0b0b0b0b0b0b0b0b
DESKEYLEN=8                   # bytes
AHALG=1                       # 1 = MD5
AHKEY=30313233343536373031323334353637
AHKEYLEN=16                   # bytes
LOCAL_SPI=$1
```

```
PEER_SPI=$1
TUNNEL_MODE=0
AHRESULTLEN=4
COMBINED_MODE=1              # On output, 1 = ESP, then
AH; 0 = AH, then ESP
DYNAMIC_FLAG=0

if [ "$ONOFF" = "on"
then
     ./sadb add dst $IPADDRESS $PREFIXLEN $LOCAL_SPI
$UID $PEERADDRESS $PEER_SPI $TUNNEL_MODE $LOCALADDRESS
$REALADDRESS $PROTOCOL $PORT $DESALG $IVLEN $DESKEYLEN
$DESKEY $DESKEYLEN $DESKEY $AHALG $AHKEYLEN $AHKEY
$AHKEYLEN $AHKEY $AHRESULTLEN $COMBINED_MODE
$DYNAMIC_FLAG
else
     ./sadb delete dst $IPADDRESS $LOCAL-SPI
fi
}

#    Get down to work:
addsa 500 172.17.128.115          # number6.sctc.com
```

The current status of in-kernel SADB 54 can be obtained with the sadb command. The get option allows dumping the entire SADB or a single entry. In one embodiment, the complete dump approach uses /dev/kmem to find the information. The information may be presented as follows:

```
# sadb get dst

Local-SPI Address-Family Destination-Addr
Preflx_length UID
     Peer-Address Peer-SPI Transport-Type
     Local-Address Real-Address
     Protocol Port
     ESP_Alg_ID ESP_IVEC_Length
          ESP_Enc_Key_length ESP_Enc_ESP_Key
          ESP_Dec_Key_length ESP_Dec_ESP_Key
     AH_Alg_ID AH_Data_Length
          AH_Gen_Key_Length AH_Gen_Key
          AH_Check_Key_Length AH_Check_Key
     Combined_Mode  Dynamic_Flag
```

```
-----------------------------------------------------------------
-
500 INET: number6.sctc.com 0 0
     0.0.0.0   500 Transport(0) 0
     0.0.0.0 0.0.0.0
     None None
     DES/CBC-RFC1829(1) 4
          8 0b0b0b0b0b0b0b0b
          8 0b0b0b0b0b0b0b0b
     MD5-RFC1828(1) 4
          16 30313233343536373031323334353637
          16 30313233343536373031323334353637
     ESP+AH(1) 0
501 INET: spokes.sctc.com 0 0
     0.0.0.0   501 Transport(0) 0
     0.0.0.0.0.0.0.0
     None None
     DES/CBC-RFC1829(1) 4
          8 0b0b0b0b0b0b0b0b
          8 0b0b0b0b0b0b0b0b
     MD5-RFC1828(l) 4
          16 30313233343536373031323334353637
          16 30313233343536373031323334353637
     ESP+AH(1) 0

End of list.
```

When a new entry is added to in-kernel SADB 54, the add process first checks to see that no existing entry will match the values provided in the new entry. If no match is found then the entry is added to the end of the existing SADB list.

To illustrate the use and administration of an FFE, we'll go through an example using FFE 70 in Figure 3. Firewalls 14 and 18 are both application level gateway firewalls implemented according to the present invention. Workstations H2 and H3 both want to communicate with Hl. For the administrator of firewalls 14 and 18, this is easy to accomplish. The administrator sets up a line something like this (we'll only show the IP address part and SPI parts of the SA, since they're the trickiest values to configure. Also, assume that we are using tunnel mode):

```
#  Hypothetical SW1 Config File
```

```
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
realsrcaddr= localaddr= key=

# The following entry sets up a tunnel between hosts
behind SW1
# and hosts behind SW2.
src=172.16.0.0 localSPI=666 peer=192.168.100.5
peerSPI=777 \
     realsrcaddr=192.168.100.5 localaddrs=0.0.0.0
     key=0xdeadbeeffadebabe

#  Hypothetical SW2 Config File
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
     realsrcaddr= localaddr= key=

#  The following entry sets up a tunnel between hosts
behind SW1 and
#  hosts behind SW2.
src=172.17.0.0 localSPI=777 peer=192.168.20.1
peerSPI=666 \
     realsrcaddr=192.168.20.1 localaddr=0.0.0.0 \
     key=0xdeadbeeffadebabe
```

With this setup, all traffic is encrypted using one key, no matter who is talking to whom. For example, traffic from H2 to Hl as well as traffic from H3 to Hl will be encrypted with one key. Although this setup is small and simple, it may not be enough.

What happens if H2 cannot trust H3? In this case, the administrator can set up security associations at the host level. In this case, we have to rely on the SPI field of the SA, since the receiving firewall cannot tell from the datagram header which host behind the sending firewall sent the packet. Since the SPI is stored in IPSEC datagrams, we can do a lookup to obtain its value. Below are the sample configuration files for both firewalls again, but this time, each host combination communicates with a different key. Moreover, H2 excludes H3 from communications with Hl, and H3 excludes H2 in the same way.

```
#  Hypothetical SW1 Config File
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
realsrcaddr= localaddr= key=

# The following entry sets up a secure link between H2
and H1
src=172.16.0.2 localSPI=666 peer=192.168.100.5
peerSPI=777 \
      realsrcaddr=192.168.100.5
localaddrs=178.17.128.71 \
      key=0x0a0a0a0a0a0a0a0a

# The following entry sets up a secure link between H3
and H1
src=172.16.0.1 localSPI=555 peer=192.168.100.5
peerSPI=888 \
      realsrcaddr=192.168.100.5
localaddrs=178.17.128.71 \
      key=0x0b0b0b0b0b0b0b0b

#  Hypothetical SW2 Config File
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
realsrcaddr= localaddr= key=

# The following entry sets up a secure link between H2
and H1
src=172.17.128.71 localSPI=777 peer=192.168.20.1
peerSPI=666 \
      realsrcaddr=192.168.20.1 localaddrs=172.16.0.2 \
      key=0x0a0a0a0a0a0a0a0a

# The following entry sets up a secure link between H3
and H1
src=172.17.128.71 localSPI=888 peer=192.168.20.1
peerSPI=555 \
      realsrcaddr=192.168.20.1 localaddrs=172.16.0.1 \
      key=0x0b0b0b0b0b0b0b0b
```

Figure 4 is a block diagram showing in more detail one embodiment of an IPSEC-enabled application level gateway firewall 18. Application level gateway firewall 18 provides access control checking of both encrypted and unencrypted messages in a more secure environment due to its network-separated architecture. Network separation divides a system into a set of independent regions or burbs, with a domain and a protocol stack assigned to each burb. Each protocol stack 40x has its own independent set of data structures, including routing information and protocol information. A given socket will be bound to a single protocol stack at creation time and no data can pass between protocol stacks 40 without going through proxy space. A proxy 50 therefore acts as the go-between for transfers between domains. Because of this, a malicious attacker who gains control of one of the regions is prevented from being able to compromise processes executing in other regions. Network separation and its application to an application level gateway is described in "SYSTEM AND METHOD FOR ACHIEVING NETWORK SEPARATION", U.S. Application No. 08/599,232, filed February 9, 1996 by Gooderum et al.

In the system shown in Figure 4, the in-bound and out-bound datagram processing of a security association continues to follow the conventions defined by the network separation model. Thus all datagrams received on or sent to a given burb remain in that burb once decrypted. In one such embodiment SADB socket 78 has been defined to have the type 'sadb'. Each proxy 50 that requires access to SADB socket 78 to execute its query as to whether the received message was encrypted must have create permission to the sadb type.

The following is list of specific requirements that a system such as is shown in Figure 4 must provide. Many of the requirements were discussed in the information provided earlier in this document.

1. Firewall applications may query the IPSEC subsystem to determine if traffic with a given address is guaranteed to be encrypted.
2. Receipt of an unencrypted datagram from an address that has a SA results in the datagram being dropped and an audit message being generated.
3. On receipt of encrypted protocol datagrams the SADB searches will be done using the SPI as the primary key. The source address will a secondary key. The SA returned by the search will be the SA which matches the SPI exactly and has the longest match with the address.

4. A search of the SADB for a SPI that finds an entry that is marked as SA for a dynamic IP will not consider the address in the search process.
5. A search of the SADB for a SPI that finds an entry that is marked as a SA for a tunnel mode connection will to consider the address if it is (0.0.0.0) i.e INADDR.
6. On receipt of a non-IPSEC datagram the SADB will be searched for an entry that matches the src address. If a SA is found the datagram will be dropped and an audit message sent.
7. SADB searches on output will be done using the DST address as key. If more than one SA entry in the SADB has that address the first one with the maximum address match will be returned.
8. The SADB must be structured so that searches are fast regardless if the search is done by SPI or by address.
9. The SADB must provide support for connections to a site with a fixed SPI but changing IP address. SA entries for such connections will be referred to as Dynamic Address Sites, or just Dynamic entries.
10. When a dynamic entry is found by a SPI search, the current datagram's SRC address, which is required to ensure that the return datagrams are properly encrypted, will be recorded in the SA only after the AH checking has passed successfully. (This is because if the address is recorded before AH passes then an attacker can cause return packets of an outgoing connection to be transmitted in the clear.)
11. A failure of an AH check on a dynamic entry results in an audit message.
12. In an embodiment where the firewall requires that all connections use both AH and ESP, on receipt the order should be AH first ESP second.
13. The processing structure on both input and output should try to minimize the number of SADB required lookups.

Returning to Figure 4, in one embodiment firewall 18 includes a crypto engine interface 80 used to encrypt an IPSEC payload. Crypto engine interface 80 may be connected to a software encryption engine 82 or to a hardware encryption engine 84. Engines 82 and 84 perform the actual encryption function using, for example, DES-CBC. In addition, software encryption engine 82 may include the keyed MD5 algorithm used for AH.

In one embodiment, crypto engine interface 80 is a utility which provides a consistent interface between the software and hardware encryption engines. As shown in Figure 4, in one such embodiment interface 80 only supports the use of the use of hardware cryptographic engine 84 for IPSEC ESP processing. The significant design issue that interface 80 must deal with is that use of a hardware encryption engine requires that the processing be down in disjoint steps operating in different interrupt contexts as engine 84 completes the various processing steps.

The required information is stored in a request structure that is bound to the IP datagram being processed. The request is of type `crypto_request_t`. This structure is quite large and definitely does not contain a minimum state set.

In addition to the definition of the request data structure, this software implementing interface 80 provides two functions which isolate the decision of which cryptographic engine to use. The `crypt_des_encrypt` function is for use by the IP output processing to encrypt a datagram. The `crypt_des_decrypt` function is for use by the IP input processing to decrypt a datagram. If hardware encryption engine 84 is present and other hardware usage criteria are met the request is enqueued on a hardware processing queue and a return code indicating that the cryptographic processing is in progress is returned. If software engine 82 is used, the return code indicates that the cryptographic processing is complete. In the former case, the continuation of the IP processing is delayed until after hardware encryption is done. Otherwise it is completed as immediately in the same processing stream.

There are two software cryptographic engines 82 provided in the IPSEC software. One provides the MD5 algorithm used by the IPSEC AH processing, and the other provides the DES algorithm used by the IPSEC ESP processing. This software can be obtained from the US Government IPSEC implementation.

In one embodiment hardware cryptographic engine 84 is provided by a Cylink SafeNode processing board. The interface to this hardware card is provided by the Cylink device driver. A significant aspect of the Cylink card that plays a major part in the design of the IPSEC Cylink driver is that the card functions much like a low level subroutine interface and requires software support to initiate each processing step. Thus to encrypt or decrypt an individual datagram there are a minimum of two steps, one to set the DES initialization vector and one to do the encryption. Since the IP processing can not suspend itself and wait while the hardware completes and then be rescheduled by the hardware interrupt handler, in one embodiment a finite state machine is used to tie sequences of hardware processing elements together. In one such embodiment the interrupt handler looks at the current state, executes a defined after state function, transitions to the state and then executes that state's start function.

One function, `cyl_enqueue_request`, is used to initiate either an encrypt or a decrypt action. This function is designed to be called by cryptographic engine interface 80. All of the information required to initiate the processing as well as the function to be performed after the encryption operation is completed is provided in the request structure. This function will enqueue the request on the hardware request queue and start the hardware processing if necessary.

A system 30 which can be used for firewall-to-workstation encryption is shown in Figure 5. In Figure 5, system 30 includes a workstation 12 communicating through a firewall 14 to an unprotected network 16 such as the Internet. System 30 also includes a workstation 32 communicating directly with firewall 14 through unprotected network 16. Firewall 14 is an application level gateway incorporating IPSEC handling as described above. (It should be noted that IPSEC security cannot be used to authenticate the personal identity of the sender for a firewall to firewall transfer. When IPSEC is used, however, on a single user machine such as a portable personal computer, IPSEC usage should be protected with a personal identification number (PIN). In these cases IPSEC can be used to help with user identification to the firewall.)

According to the IPSEC RFC's, you can use either tunnel or transport mode with this embodiment based on your security needs. In certain situations, the communications must be sent in tunnel mode to hide unregistered addresses.

Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiment shown. This application is intended to cover any adaptations or variations of the present invention. Therefore, it is intended that this invention be limited only by the claims and the equivalents thereof.

What is claimed is:

1. A method of regulating the flow of messages through a firewall having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:

determining, at the IP layer, if a message is encrypted;

if the message is not encrypted, passing the unencrypted message up the network protocol stack to an application level proxy; and

if the message is encrypted, decrypting the message and passing the decrypted message up the network protocol stack to the application level proxy, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message.

2. A method of authenticating the sender of a message within a computer system having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:

determining, at the IP layer, if the message is encrypted;

if the message is encrypted, decrypting the message, wherein the step of decrypting the message includes the step of executing a process at the IP layer to decrypt the message;

passing the decrypted message up the network protocol stack to an application level proxy;

determining an authentication protocol appropriate for the message; and

executing the authentication protocol to authenticate the sender of the message.

3. The method according to claim 2 wherein the step of determining an authentication protocol appropriate for the message includes the steps of:

determining a source IP address associated with the message; and

determining the authentication protocol associated with the source IP address.

4. The method according to claim 2 wherein the message includes security parameters index and wherein the step of determining an authentication protocol appropriate for the message includes the steps of:

determining the authentication protocol associated with a dynamic IP address, wherein the step of determining the authentication protocol includes the step of looking up a security association based on the security parameters index;

determining a current address associated with the dynamic source IP address; and

binding the current address to the security parameters index.

5. A firewall, comprising:

a first communications interface;

a second communications interface;

a network protocol stack connected to the first and the second communications interfaces, wherein the network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

a decryption procedure, operating at the IP layer, wherein the decryption procedure decrypts encrypted messages received at one of said first and second communications interfaces and outputs decrypted messages; and

a proxy, connected to the transport layer of said network protocol stack, wherein the proxy receives decrypted messages from the decryption procedure and executes an authentication protocol based on the content of the decrypted message.

6. A firewall, comprising:

a first communications interface;

a second communications interface;

a first network protocol stack connccted to the first communications interface, wherein the first network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

a second network protocol stack connected to the second communications interface, wherein the second network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

a decryption procedure, operating at the IP layer of the first network protocol stack, the decryption procedure receiving encrypted messages received by said first communications interface and outputting decrypted messages; and

a proxy, connected to the transport layers of said first and second network protocol stacks, the proxy receiving decrypted messages from the decryption procedure and executing an authentication protocol based on the content of the decrypted message.

7. The firewall according to claim 6 wherein the firewall further includes:

a third communications interface; and

a third network protocol stack connected to the third communications interface and to the proxy, wherein the third network protocol stack includes an Internet Protocol (IP) layer and a transport layer and wherein the second and third network protocol stacks are restricted to first and second burbs, respectively.

8. A method of establishing a virtual private network between a first and a second network, wherein each network includes an application level gateway firewall which uses a proxy operating at the application layer to process traffic through the firewall, wherein each firewall includes a network protocol stack and wherein each network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:

transferring a connection request from the first network to the second network;

determining, at the IP layer of the network protocol stack of the second network's firewall, if the connection request is encrypted;

if the connection request is encrypted, decrypting the request, wherein the step of decrypting the request includes the step of executing a procedure at the IP layer of the second network's firewall to decrypt the message;

passing the connection request up the network protocol stack to an application level proxy;

determining an authentication protocol appropriate for the connection request;

executing the authentication protocol to authenticate the connection request; and

if the connection request is authentic, establishing an active connection between the first and second networks.

9. The method according to claim 8 wherein the step of executing the authentication protocol includes the step of executing program code within the firewall of the second network to mimic a challenge/response protocol executing on a server internal to the second network.

10. The method according to claim 8 wherein the step of executing the authentication protocol includes the step of executing program code to execute the authentication protocol in line to the session.

11. The method according to claim 8 wherein the step of determining an authentication protocol includes the step of determining if the connection request arrived encrypted and selecting the authentication protocol based on whether the connection request was encrypted or not encrypted.

The Patent Office

28

Application No:	GB 9719816.2	**Examiner:**	B.J.SPEAR
Claims searched:	1-11	**Date of search:**	21 January 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): H4P (PPEB,PDCSA,PDCSC)

Int Cl (Ed.6): H04L 9/00, 9/32, 29/06, 29/08

Other: Online:WPI, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
XP	WO97/26734A1	(Raptor Systems) Whole document, eg Figs 1,3 and pages 6-12	1,2.5,6,8 at least
XP	WO97/26731A1	(Raptor Systems) Whole document, eg Figs 1,3 and pages 7-12	1,2.5,6,8 at least
XP	WO97/26735A1	(Raptor Systems) Whole document, eg Figs 1,3 and pages 4-10	1,2.5,6,8 at least
XP	WO97/23972A1	(V-ONE Corp) Whole document, eg Figs 1,2 and claim 1.	1,2.5,6,8 at least
XP	WO97/13340A1	(Digital Secured Networks) Whole document, eg pages 7-13	1,2.5,6,8 at least

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		

An Executive Agency of the Department of Trade and Industry

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

#3
T.D.
11/28/03

In re Application of	
Edmond Colby MUNGER *et al.*	
Serial No.:	10/259,494
Filed:	September 30, 2002
For:	Improvement To An Agile Network Protocol For Secure Communications With Assured System Availability

Group Art Unit: 2152

Examiner: Unassigned

Atty. Dkt. No. 000479.00082

RECEIVED
JUN 26 2003
Technology Center 2100

REQUEST TO CORRECT INVENTORSHIP

Commissioner of Patents
P.O. Box 1450
Alexandria, VA 22313

Pursuant to 37 C.F.R. § 1.48 (b), Applicants hereby request correction of inventorship of the above-captioned application as follows:

Delete inventor Douglas Charles Schmidt.

REMARKS

The present application is a divisional application of 09/504,783 (the '783 application), now U.S. Pat 6,502,135, issued December 31, 2002 (the '135 patent). During prosecution of the '783 application, the Office issued a restriction requirement. At least some claims directed to inventor Schmidt's invention were elected in the '783 application that issued as the '135 patent, however, his invention is not claimed in the cancelled claims of the '783 application that are being pursued in the present application.

The Office is hereby authorized to charge any required fee for this Request To Correct Inventorship to the undersigned's Deposit Account No. 19-0733. If the Examiner has any questions, the examiner is requested to contact the undersigned at (202) 824-3153.

Respectfully submitted,

BANNER & WITCOFF, LTD.

Dated: June 23, 2003

Ross A. Dannenberg
Reg. No. 49,024
1001 G Street, N.W.
Washington, D.C. 20001-4597
(202) 824-3000

GP 2152

PTO/SB/21 (08-00)

Please type a plus sign (+) inside this box → [+]

Approved for use through 10/31/2002. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/259,494
Filing Date	September 30, 2002
First Named Inventor	Edmond Colby Munger
Group Art Unit	2152
Examiner Name	TBA
Total Number of Pages in This Submission	
Attorney Docket Number	000479.00082

ENCLOSURES *(check all that apply)*

- [x] Fee Transmittal Form
  - [ ] Fee Attached
- [ ] Amendment / Response
  - [ ] After Final
  - [ ] Affidavits/declaration(s)
- [ ] Extension of Time Request
- [ ] Express Abandonment Request
- [ ] Information Disclosure Statement
- [ ] Certified Copy of Priority Document(s)
- [ ] Response to Missing Parts/ Incomplete Application
  - [ ] Response to Missing Parts under 37 CFR 1.52 or 1.53
- [x] Assignment Papers *(for an Application)*
- [ ] Drawing(s)
- [ ] Licensing-related Papers
- [ ] Petition
- [ ] Petition to Convert to a Provisional Application
- [ ] Power of Attorney, Revocation Change of Correspondence Address
- [ ] Terminal Disclaimer
- [ ] Request for Refund
- [ ] CD, Number of CD(s) _____
- [ ] After Allowance Communication to Group
- [ ] Appeal Communication to Board of Appeals and Interferences
- [ ] Appeal Communication to Group *(Appeal Notice, Brief, Reply Brief)*
- [ ] Proprietary Information
- [ ] Status Letter
- [x] Other Enclosure(s) *(please identify below)*:

Request To Correct Inventorship

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Ross A. Dannenberg, Reg. No. 49,024
Signature	Ross D
Date	June 23, 2003

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on this date:

Typed or printed name			
Signature		Date	

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be send to the Chief Information Officer, U.S. Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Washington, DC 20231.

L Number	Hits	Search Text	DB	Time stamp
1	5630	encrypt$ same channel	USPAT; US-PGPUB	2004/02/22 13:33
2	2312	(encrypt$ same channel) and secure same communication	USPAT; US-PGPUB	2004/02/22 13:33
3	279	((encrypt$ same channel) and secure same communication) and DNS	USPAT; US-PGPUB	2004/02/22 13:33
4	150	(((encrypt$ same channel) and secure same communication) and DNS) and authoriz$ same access$	USPAT; US-PGPUB	2004/02/22 13:34
5	78	((((encrypt$ same channel) and secure same communication) and DNS) and authoriz$ same access$) and proxy same server	USPAT; US-PGPUB	2004/02/22 13:34
6	60	(((((encrypt$ same channel) and secure same communication) and DNS) and authoriz$ same access$) and proxy same server) and client and target	USPAT; US-PGPUB	2004/02/22 13:35
7	25	((((((encrypt$ same channel) and secure same communication) and DNS) and authoriz$ same access$) and proxy same server) and client and target) and DNS same request	USPAT; US-PGPUB	2004/02/22 13:35

380/

713/160
189
190

L Number	Hits	Search Text	DB	Time stamp
1	1	("6618761").PN.	USPAT; US-PGPUB	2004/06/18 10:18
2	6144	encrypt$ same channel	USPAT; US-PGPUB	2004/06/18 11:30
3	474	(encrypt$ same channel) and DNS	USPAT; US-PGPUB	2004/06/18 11:30
4	307	((encrypt$ same channel) and DNS) and client and server	USPAT; US-PGPUB	2004/06/18 11:31
5	217	(((encrypt$ same channel) and DNS) and client and server) and authoriz$ same access$	USPAT; US-PGPUB	2004/06/18 11:32
7	36	((((encrypt$ same channel) and DNS) and client and server) and authoriz$ same access$) and DNS same request) and establish$ same encrypt$ same channel	USPAT; US-PGPUB	2004/06/18 11:33
6	82	((((encrypt$ same channel) and DNS) and client and server) and authoriz$ same access$) and DNS same request	USPAT; US-PGPUB	2004/06/18 11:43

2

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

22907 7590 06/24/2004

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

EXAMINER	
LIM, KRISNA	
ART UNIT	**PAPER NUMBER**
2153	4

DATE MAILED: 06/24/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)
	10/259,494	MUNGER ET AL.
	Examiner	**Art Unit**
	Krisna Lim	2153

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-20 is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) 1-20 is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date 9/30/02
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

1. Claims 1-20 are presented for examination.

2. The title of the invention is neither descriptive nor precise. A new title is required which should include, using twenty words or fewer, claimed features that differentiate the invention from the Prior Art. The title should reflect the gist of or the improvement of the present invention.

3. The disclosure is objected to because of the following informalities:

(a) On page 1, the text of the first paragraph should be updated with the current status of the cited applications such as U.S. Patent Application Serial No., a filing date, U.S. Patent No., and the issued date. Appropriate correction is required.

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-20 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Strentzsch et al. [U.S. Patent No. 6,256,671].

6. Strentzsch et al. disclose (e.g., see Figs. 1-7) the invention substantially as claimed. Taking claim 1, 3 and 6 as an exemplary claims, the reference discloses a method for establishing an encrypted channel between a client and a target computer (if a source is allowed to access a host corresponding to the host name, then providing the address to the source, col. 14, lines 18-21), comprising the steps of: I) intercepting a DNS request sent by the client (e.g., see 505 of Fig. 5, col. 5, line 55, to col. 8, line 60); ii) based on the DNS request, establishing the encrypted channel between the client and the target (e.g., see 510 to 530 of Fig. 5, col. 5, line 55, to col. 8, line 60).

7. While Strentzsch discloses: a) a more secure way to control access by the user to host (target) systems on the network; b) typical access control programs indicate to the user that, due to the access management setting, the user is prevented from accessing the desired host system; and c) a network access control for returning address to the source after checking to make sure that the user is allowed to access the host, checking whether or not the IP address corresponding to the received host name can be located and checking whether or not the user allowed to access IP address (e.g., see col. 1, line 28, to col. 2, line 11), Strentzsch does not explicitly mention the term "establishing the encrypted channel between the client and the target." It would have been obvious to one of ordinary skill in the art at the time the invention was made to recognize that in order to control access by the user to the host in a secure way the communication channel between the client (user) and the target (host) must be encrypted. Moreover, such encryption feature is a well-known feature in the art (e.g., see any computer dictionary for the definition).

8. As to claim 2, Strentzsch et al. disclose the steps of: a) determining whether the client is authorized to access the target (e.g., see 510 to 530 of Fig. 5, col. 5, line 55, to col. 8, line 60); b) when the client is authorized to access the target, initiating the encrypted channel (see paragraph 7 above for the teaching of encryption feature); and c) when the client is not authorized to access the target, sending an error message to the client (e.g., see 510 to 530 of Fig. 5, col. 5, line 55, to col. 8, line 60).

9. As to claim 5, Strentzsch et al. disclose the step of a DNS proxy server (160) intercepting the DNS request sent by the client (e.g., see 510 to 530 of Fig. 5, col. 5, line 55, to col. 8, line 60).

10. Claims 7-20 are similar in scope as of claims 1-6, and therefore claims 7-20 are rejected for the same reasons set forth above for claims 1-6.

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

The references are cited in the Form PTO-892 for the applicant's review.

A shortened statutory period for response to this action is set to expire 3 (three) months and 0 (zero) days from the mail date of this letter. Failure to respond within the period for response will result in **ABANDONMENT** of the application (see 35 U.S.C 133, M.P.E.P 710.02, 710.02(b)).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Examiner Krisna Lim whose telephone number is (703) 305-9672. The examiner can normally be reached on Monday-Friday from 7:30 to 5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Mr. Glenton Burgess, can be reached at (703) 305-4772. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is 703-305-9700

Communications via Internet e-mail regarding this application, other than those under 35 U.S.C. 132 or which otherwise require a signature, may be used by the applicant and should be addressed to [glen.burgess@uspto.gov].

All Internet e-mail communication will be made of record in the application file. PTO employees do not engage in Internet communications where there exists a possibility that sensitive information could be identified or exchanged unless the record includes a properly signed express waiver of the confidentiality requirement of 35 U.S.C. 122. This is more clearly set forth in the Interim Internet Usage Policy published in the Office Gazette of the Patent and Trademark on February 25, 1997 at 1195 OG 89.

kl

June 22, 2004

KRISNA LIM
PRIMARY EXAMINER

Notice of References Cited	Application/Control No. 10/259,494	Applicant(s)/Patent Under Reexamination MUNGER ET AL.	
	Examiner Krisna Lim	Art Unit 2153	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-6,256,671	07-2001	Strentzsch et al.	709/227
	B	US-6,332,158	12-2001	Risley et al.	709/219
	C	US-5,164,986	11-1992	Bright, Michael W.	380/273
	D	US-6,079,020	06-2000	Liu, Quentin C.	713/201
	E	US-6,425,003	07-2002	Herzog et al.	709/223
	F	US-6,751,738	06-2004	Wesinger et al.	713/201
	G	US-6,606,708	08-2003	Devine et al.	713/201
	H	US-2003/0196122	10-2003	Wesinger et al.	713/201
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00082	SERIAL NUMBER DIV of 09/504,783
	APPLICANT Edmond Colby Munger et al.	
	FILING DATE September 30, 2002	GROUP ART UNIT 2153

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
KL	6,119,171	9/2000	Alkhatib	---	---	

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
KL	199 24 575	12/2/99	DE				
KL	0 838 930	4/29/98	EPO				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

KL	Search Report (dated 8/23/02), International Application No. PCT/US01/13260
	Donald E. Eastlake, 3rd, "Domain Name System Security Extensions", INTERNET DRAFT, April 1998, pages 1-51
	D. B. Chapman et al., "Building Internet Firewalls", November 1995, pages 278-375
	P. Srisuresh et al., "DNS extensions to Network address Translators (DNS_ALG)", INTERNET DRAFT, July 1998, pages 1-27
	James E. Bellaire, "New Statement of Rules – Naming Internet Domains", Internet Newsgroup, July 30, 1995, 1 page
	D. Clark, "US Calls for Private Domain-Name System", Computer, IEEE Computer Society, August 1, 1998, pages 22-25
	August Bequai, "Balancing Legal Concerns Over Crime and Security in Cyberspace", Computer & Security, Vol. 17, No. 4, 1998, pages 293-298
	Rich Winkel, "CAQ: Networking With Spooks: The NET & The Control Of Information", Internet Newsgroup, June 21, 1997, 4 pages

EXAMINER	KRISNA LIM	DATE CONSIDERED	3/10/04

EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00082	SERIAL NUMBER DIV of 09/504,783
	APPLICANT Edmond Colby Munger et al.	
	FILING DATE September 30, 2002	GROUP ART UNIT 2153

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
KL	5,588,060	12/24/96	Aziz			
KL	5,689,566	11/18/9	Nguyen			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
KL	199 24 575	12/2/99	DE				
	0 838 930	4/29/98	EPO				
	2 317 792	4/1/98	GB				
	0 814 589	12/29/97	EPO				
✓	WO 98/27783	6/25/98	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

KL	Search Report (dated 6/18/02), International Application No. PCT/US01/13260
	Search Report (dated 6/28/02), International Application No. PCT/US01/13261
	Donald E. Eastlake, "Domain Name System Security Extensions", DNS Security Working Group, April 1998, 51 pages
	D. B. Chapman et al., "Building Internet Firewalls", November 1995, pages 278-297 and pages 351-375
	P. Srisuresh et al., "DNS extensions to Network Address Translators", July 1998, 27 pages
	Laurie Wells, "Security Icon", October 19, 1998, 1 page
	W. Stallings, "Cryptography And Network Security", 2nd Edition, Chapter 13, IP Security, June 8, 1998, pages 399-440
✓	W. Stallings, "New Cryptography and Network Security Book", June 8, 1998, 3 pages

EXAMINER KRISNA LIM	DATE CONSIDERED 2/20/04
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.	

PTO-1449 (Modified) U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE INFORMATION DISCLOSURE STATEMENT BY APPLICANT	ATTY. DOCKET NO. 000479.00082	SERIAL NUMBER DIV of 09/504,783
	APPLICANT Edmond Colby Munger et al.	
	FILING DATE September 30, 2002	GROUP ART UNIT 2153

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
KL	0 858 189	8/12/98	EPO				
KL	WO 01 50688	7/12/01	PCT				
KL	WO 98 59470	12/30/98	PCT				
KL	WO 99 48303	9/23/99	PCT				
KL	WO 99 38081	7/29/99	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

KL	Search Report (dated 8/20/02), International Application No. PCT/US01/04340
KL	Shree Murthy et al., "Congestion-Oriented Shortest Multipath Routing", Proceedings of IEEE INFOCOM, 1996, pages 1028-1036
KL	Jim Jones et al., "Distributed Denial of Service Attacks: Defenses", Global Integrity Corporation, 2000, pages 1-14
KL	FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security: Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER Krisna Lim	DATE CONSIDERED 2/20/04
EXAMINER: Initial citation if reference was considered. Draw line through citation if not in conformance to MPEP 609 and not considered. Include copy of this form with next communication to applicant.	

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

Edmund Colby MUNGER *et al.*

Serial No.: 10/259,494

Filed: September 30, 2002

For: Establishment of a Secure Communication Link Based on a Domain Name Service (DNS) Request (As Amended)

Group Art Unit: 2153

Examiner: Krisna Lim

Atty. Dkt. No. 000479.00082

AMENDMENT

U.S. Patent and Trademark Office
220 20th Street S.
Customer Window
Crystal Plaza Two, Lobby, Room 1B03
Arlington, VA 22202

Sir:

In response to the Office Action mailed June 24, 2004, please amend the instant application as follows:

Amendments to the Specification begin on page 2 of this paper.

Amendments to the Claims are reflected in the Listing of Claims, which begins on page 3 of this paper.

Remarks/Arguments begin on page 9 of this paper.

It is believed that no fee is required for this submission. If any fees are required or if an overpayment is made, the Commissioner is authorized to debit or credit our Deposit Account No. 19-0733, accordingly.

Amendments to the Specification:

Please replace the title with the following rewritten title:

ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

Please replace paragraph [01] on page 1 with the following rewritten paragraph:

[01] This application is a divisional application of 09/504,783 (filed February 15, 2000), now U.S. Pat. No. 6,502,135, issued December 31, 2002, which claims priority from and is a continuation-in-part of previously filed U.S. application serial number 09/429,643 (filed October 29, 1999). The subject matter of the '643 application, which is bodily incorporated herein, derives from provisional U.S. application numbers 60/106,261 (filed October 30, 1998) and 60/137,704 (filed June 7, 1999).

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Original) A method for establishing an encrypted channel between a client and a target computer, comprising the steps of:

(i) intercepting a DNS request sent by the client; and

(ii) based on the DNS request, establishing the encrypted channel between the client and the target.

2. (Original) The method of claim 1, wherein step (ii) comprises steps of:

a) determining whether the client is authorized to access the target;

b) when the client is authorized to access the target, initiating the encrypted channel; and

c) when the client is not authorized to access the target, sending an error message to the client.

3. (Original) The method of claim 2, wherein step b) comprises sending encrypted channel parameters to the client.

4. (Original) The method of claim 1, wherein step (ii) occurs in a communication protocol independently of an application program.

5. (Original) The method of claim 1, wherein step (i) comprises a DNS proxy server intercepting the DNS request sent by the client.

6. (Original) The method of claim 1, wherein step (ii) comprises establishing the encrypted channel responsive to intercepting a DNS request for a domain name comprising a predetermined domain name extension.

7. (Original) A method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name extension.

8. (Original) The method of claim 7, wherein the creating step is performed in a communication protocol independently of an application program.

9. (Original) A method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel in response to detecting a request for access to a predetermined IP address.

10. (Original) The method of claim 9, wherein the creating step is performed in a communication protocol independently of an application program.

11. (Original) A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:

(i) determining whether the intercepted DNS request corresponds to a secure server;

(ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

12. (Original) The data processing device of claim 11, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

13. (Original) The data processing device of claim 12, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

14. (Original) The data processing device of claim 13, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

15. (Original) A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

16. (Original) A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) intercepting a DNS request sent by a client;

(ii) determining whether the intercepted DNS request corresponds to a secure server;

(iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

17. (Original) The computer readable medium of claim 16, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

18. (Original) The computer readable medium of claim 17, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

19. (Original) The computer readable medium of claim 18, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

20. (Original) A computer readable medium comprising computer readable instructions that, when executed, cause a domain name server (DNS) proxy module to intercept DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

21. (New) The method of claim 1, wherein step (ii) comprises establishing an IP address hopping scheme between the client and the target.

22. (New) The method of claim 1, wherein step (ii) avoids sending a true IP address of the target to the client.

23. (New) The method of claim 7, wherein creating the encrypted channel comprises establishing an IP address hopping scheme between the client and the secure host.

24. (New) The method of claim 7, wherein creating the encrypted channel avoids sending a true IP address of the secure host to the client.

25. (New) The method of claim 9, wherein automatically creating the encrypted channel comprises establishing an IP address hopping scheme between the client and the secure host.

26. (New) The method of claim 9, wherein automatically creating the encrypted channel avoids sending a true IP address of the secure host to the client.

27. (New) The data processing device of claim 11, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

28. (New) The data processing device of claim 11, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

29. (New) The data processing device of claim 15, wherein automatically initiating an encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

30. (New) The data processing device of claim 15, wherein automatically initiating an encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

31. (New) The computer readable medium of claim 16, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

32. (New) The computer readable medium of claim 16, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

33. (New) The computer readable medium of claim 20, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

34. (New) The computer readable medium of claim 20, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

REMARKS/ARGUMENTS

The office action of June 24, 2004 has been carefully reviewed and these remarks are responsive thereto. Reconsideration and allowance of the instant application are respectfully requested. The specification has been amended to update the status of cited pending applications. The title has been amended to be more descriptive. Claims 21-24 have been added. Claims 1-34 thus remain pending in this application.

Amendments to the Specification

The Office Action objects to the title as being neither descriptive nor precise. Applicants have amended the title to be more descriptive, and respectfully request the objection to the title be withdrawn.

The Office Action objects to paragraph 1 of the disclosure in the specification, requiring the first paragraph to be updated to reflect the current status of the cited applications. Applicants have amended the first paragraph of the specification accordingly, and respectfully request the objection to the disclosure be withdrawn.

Rejections Under 35 U.S.C. § 103

Claims 1-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Strentzsch *et al.* (U.S. Pat. No. 6,256,671, hereinafter Strentzsch). The rejection is respectfully traversed.

In order to reject a claim as obvious under § 103(a), three criteria must exist: 1) there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine the reference teachings; 2) there must be a reasonable expectation of success; and 3) the prior art reference(s) must teach or suggest all the claim limitations. *See* MPEP § 706.02 (j); *In re Vaeck*, 947 F.2d 488 (Fed. Cir. 1991).

Non-Analogous Arts

With respect to the first criteria, in order to rely on a reference as a basis for rejection under 35 U.S.C. § 103, the reference must either be in the field of an applicant's endeavor or, if not, then be reasonably pertinent to the particular problem with which the invention was concerned. Neither is the case here.

Strentzsch, generally, provides a method and apparatus for *preventing* network access by manipulating a domain name system, whereas the presently claimed invention *establishes* secure network connections. That is, as the Office Action correctly indicates, Strentzsch is in a field of endeavor concerned with completely preventing users' or computers' access to systems or to particular web sites, as is evident from the context of the disclosure in Strentzsch. For example, Strentzsch states:

> Network administrators frequently want to limit individuals' abilities to access various host systems. For example, a parent may want to prevent his or her children from accessing host systems storing content unsuitable for children. By way of another example, an employer may want to prevent employees from accessing particular host systems using the employer's equipment.

Strentzsch, col. 1, lines 34-42. Strentzsch provides descriptions and details of events that occur when a user is *not* allowed to access a particular domain name or IP address at all. Strentzsch describes a method and apparatus that *prevents* a user from accessing an IP address by giving the user a DNS error message that the host could not be found. Strentzsch, col. 6, line 58 – col. 7, line 33; col. 9, lines 18 – 28. Nowhere does Strentzsch imply or suggest that the DNS method and apparatus described therein is used in the context of *establishing* a secure connection or for Internet privacy and/or security. Indeed, *preventing* access is quite inapposite to *establishing* access.

Aspects of the present invention, on the other hand, are concerned with, and should be considered to be in the art of, the establishment of a secure connection between computers. The use of DNS to establish the secure connection is a solution to this problem posed by the presently claimed invention. Specification, ¶¶ 191-206. Furthermore, aspects of the present invention address a security flaw in the DNS system actually used by Strentzsch. Namely, nefarious listeners on the Internet could intercept the DNS REQ and DNS RESP packets in Strentzsch and thus learn what IP addresses the user was contacting. For example, if a user wanted to set up a secure communication path with a web site having the name "Target.com," when the user's browser contacted a DNS to find the IP address for that web site, the true IP address of that web site would be revealed over the Internet as part of the common DNS inquiry in Strentzsch. This would hamper anonymous communications on the Internet. Spec., ¶ 194.

In a comparable 'analogous art' scenario, an application and an issued patent were found to be in non-analogous arts even though both were directed to Single Inline Memory Modules (SIMMs). The application and the reference were not necessarily in the same field of endeavor simply because they both related to memories, or even because they both related to SIMM memory modules. The reference to a SIMM for an industrial controller was found to be in a different field of endeavor than the claimed subject matter because the reference taught memory circuits in which modules of different sizes may be added or replaced, whereas the claims concerned compact memory modules. See MPEP § 2141.01(a).

Likewise, it is clear that even though the present application and Strentzsch both use DNS intercepts, they do so to solve different problems. Strentzsch *prevents* persons (such as children and employees) and/or computers from accessing web sites deemed not allowed, whereas the pending claims all concern the *establishment of a secure communication link*, and area in which Strentzsch is silent. Stated another way, Strentzsch uses DNS intercepts to *prevent* access to particular IP addresses. When the user is allowed access to an IP address, the DNS request in Strentzsch behaves as normal, disregarding issues of internet privacy and security addressed by the present invention. That is, the present invention uses DNS as a solution to the problem of finding a secure and private way to establish a secure communication link between two computers. Applicants' thus respectfully submit that Strentzsch and the present application are not in analogous arts.

Modification of Strentzsch Defeats its Purpose

In addition to the above, modification of Strentzsch in such a way as apparently alleged by the Office Action would render Strentzsch unsatisfactory for its intended purpose. See MPEP 2143.01. That is, Strentzsch completely prevents users and computers from accessing specific IP addresses. Modifying Strentzsch to establish a secure connection between two computers is at odds with preventing access to an IP address.

Request for Clarification

The Office Action states, at paragraph 7, that Strentzsch does not mention the establishment of a secure channel between the client and the target based on the DNS request. The Office Action goes on to state that "it would have been obvious to one of ordinary skill in the art at the time the invention was made to recognize that in order to control access by the user to the host in a secure way the communication channel between the client (user) and the target (host) *must be* encrypted. Moreover, such encryption feature is a well known feature in the art (e.g., see any computer dictionary for the definition)." Office Action, ¶. 7 (emphasis added). It is unclear whether the examiner is stating that it is inherent that secure communications are described, due to the Office Action's use of "must be" language. To the extent this is an inherency rejection (which seems illogical in a § 103 rejection), the use of encryption is not inherent because it is not necessary. Indeed, Strentzsch accomplishes its task, i.e., preventing access to an IP address, without the use of encryption. To any extent the rejection is not based on inherency, the Office Action seems to state that it would have been obvious to modify Strentzsch in view of "any computer dictionary" without providing any details as to how or which one. Applicants thus have no fair opportunity to respond. If the rejection is maintained, clarification is respectfully requested, including citation to an actual reference showing encryption and the motivation to establish a secure communication link based on a DNS request.

Lack of Motivation to Combine

In view of the above argument, one of ordinary skill in the art would not be motivated to look to Strentzsch for a solution to the problem of establishing a secure connection between two computers. The prior art must show the encryption and the desirability to modify the primary reference. Indeed, the only suggestion or motivation to create a secure communication link based on a DNS request is Applicants' own specification and/or claims, which cannot be used as a motivating factor to combine or modify prior art. The Federal Circuit has repeatedly stated that the limitations of a claim in a pending application cannot be used as a blueprint to piece together prior art in hindsight, *In re Dembiczak*, 50 U.S.P.Q.2d 1614 (Fed. Cir. 1999), and that the Patent

Office should *rigourously* apply the requirement that a teaching or motivation to combine prior art references needs to be provided. *Id.* (emphasis added).

Additionally, when ascertaining the differences between the prior art and the claims at issue, one must consider the claims and the prior art as a whole, not merely whether the differences themselves would have been obvious. See MPEP § 2141.02. The claims of the present invention, as a whole, are directed to the establishment of a secure communication link between two computers. This further supports the above arguments that the claims as a whole are not obvious in view of Strentzsch because Strentzsch is not concerned with creating a secure connection, but rather is concerned with preventing access to certain IP addresses.

Independent claim 1 of the present invention, on the other hand, recites a method for *establishing* an encrypted channel comprising, *inter alia*, "based on the DNS request, establishing the encrypted channel between the client and the target." That is, claim 1 as a whole is directed to establishing an encrypted channel, which is not taught or suggested by Strentzsch.

Independent claim 7 is similarly directed to a method for establishing an encrypted channel between a client and a secure host comprising "the step of automatically creating the encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name extension."

Independent claim 9 is similarly directed to a method for establishing an encrypted channel comprising "the step of automatically creating the encrypted channel in response to detecting a request for access to a predetermined IP address."

Independent claim 11 is directed to a data processing device that automatically initiates an encrypted channel between a client and a secure server when an intercepted DNS request corresponds to the secure server.

Independent claim 15 is directed to a data processing device that has a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

Independent claim 16 is directed to a computer readable medium storing DNS proxy server instructions for execution on a computer that, when an intercepted DNS request corresponds to a secure server, automatically initiates an encrypted channel between a client and the secure server.

Independent claim 20 is directed to a computer readable medium comprising computer readable instructions that, when executed, cause a domain name server (DNS) proxy module to intercept DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

Thus, Applicants respectfully submit that that there is no motivation or suggestion to combine Strentzsch, which discloses a method for completely preventing access to specific IP addresses, with the level of knowledge of one of ordinary skill in the art, to arrive at the claimed invention, which is directed to establishing a secure channel for communication between two computers, as claimed in the pending claims. Quite simply, the Office Action has not made a *prima facie* case of obviousness, and this rejection is respectfully traversed.

New Claims

Applicants have added new claims 21-34 to more broadly and completely claim that which the inventors consider their invention. No new matter has been added.

CONCLUSION

All rejections having been addressed, applicant respectfully submits that the instant application is in condition for allowance, and respectfully solicits prompt notification of the same. However, if for any reason the Examiner believes the application is not in condition for allowance or there are any questions, the examiner is requested to contact the undersigned at (202) 824-3153.

Respectfully submitted,

BANNER & WITCOFF, LTD.

Dated this 13 day of Sept., 2004

By: Ross Dannenberg

Ross Dannenberg, Registration No. 49,024

1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel: (202) 824-3000
Fax: (202) 824-3001

RAD/mmd

2153 41

PTO/SB/21 (02-04)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

OIPE JCW SEP 1 3 2004 PATENT & TRADEMARK OFFICE

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/259,494
Filing Date	September 30, 2002
First Named Inventor	Edmund Colby Munger
Art Unit	2153
Examiner Name	Krisna Lim
Total Number of Pages in This Submission	
Attorney Docket Number	000479.00082

ENCLOSURES ***(check all that apply)***

- ☒ Fee Transmittal Form
  - ☐ Fee Attached
- ☒ Amendment / Reply
  - ☐ After Final
  - ☐ Affidavits/declaration(s)
- ☐ Extension of Time Request
- ☐ Express Abandonment Request
- ☐ Information Disclosure Statement
- ☐ Certified Copy of Priority Document(s)
- ☐ Response to Missing Parts/ Incomplete Application
- ☐ Response to Missing Parts under 37 CFR 1.52 or 1.53
- ☐ Drawing(s)
- ☐ Licensing-related Papers
- ☐ Petition
- ☐ Petition to Convert to a Provisional Application
- ☐ Power of Attorney, Revocation Change of Correspondence Address
- ☐ Terminal Disclaimer
- ☐ Request for Refund
- ☐ CD, Number of CD(s) _____
- ☐ After Allowance Communication to Group
- ☐ Appeal Communication to Board of Appeals and Interferences
- ☐ Appeal Communication to Group **(Appeal Notice, Brief, Reply Brief)**
- ☐ Proprietary Information
- ☐ Status Letter
- ☐ Other Enclosure(s) *(please identify below)*:

Remarks

RECEIVED
SEP 1 6 2004
Technology Center 2100

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Ross A. Dannenberg, Reg. No. 49,024
Signature	Ross D[signature]
Date	September 13, 2004

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Typed or printed name			
Signature		Date	

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/17 (10-03)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE

FEE TRANSMITTAL for FY 2004

Effective 10/01/2003. Patent fees are subject to annual revision.

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 252

Complete if Known	
Application Number	10/259,494
Filing Date	September 30, 2002
First Named Inventor	Edmund Colby Munger
Examiner Name	Krisna Lim
Art Unit	2153
Attorney Docket No.	000479.00082

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit card ☐ Money Order ☐ Other ☐ None

☒ Deposit Account:

Deposit Account Number: 19-0733

Deposit Account Name: Banner & Witcoff, LTD.

The Director is authorized to: *(check all that apply)*
☒ Charge fee(s) indicated below ☒ Credit any overpayments
☒ Charge any additional fee(s) during the pendency of this application
☐ Charge fee(s) indicated below, **except for the filing fee**
to the above-identified deposit account.

FEE CALCULATION

1. BASIC FILING FEE

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	
1002	340	2002	170	Design filing fee	
1003	530	2003	265	Plant filing fee	
1004	770	2004	385	Reissue filing fee	
1005	160	2005	80	Provisional filing fee	

SUBTOTAL (1) ($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

			Extra Claims		Fee from below		Fee Paid
Total Claims	34	-20 ** =	14	X	18	=	252
Independent Claims	7	-7 ** =	0	X		=	0
Multiple Dependent	0			X	290	=	0

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description
1202	18	2202	9	Claims in excess of 20
1201	86	2201	43	Independent claims in excess of 3
1203	290	2203	145	Multiple dependent claim, if not paid
1204	86	2204	43	** Reissue independent claims over original patent
1205	18	2205	9	** Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) ($) 252

***or number previously paid, if greater; For Reissues, see above*

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Large Entity Fee ($)	Small Entity Fee Code	Small Entity Fee ($)	Fee Description	Fee Paid
1051	130	2051	65	Surcharge - late filing fee or oath	
1052	50	2052	25	Surcharge - late provisional filing fee or cover sheet.	
1053	130	1053	130	Non-English specification	
1812	2,520	1812	2,520	For filing a request for reexamination	
1804	920*	1804	920*	Requesting publication of SIR prior to Examiner action	
1805	1,840*	1805	1,840*	Requesting publication of SIR after Examiner action	
1251	110	2251	55	Extension for reply within first month	
1252	420	2252	210	Extension for reply within second month	
1253	950	2253	475	Extension for reply within third month	
1254	1,480	2254	740	Extension for reply within fourth month	
1255	2,010	2255	1,005	Extension for reply within fifth month	
1401	330	2401	165	Notice of Appeal	
1402	330	2402	165	Filing a brief in support of an appeal	
1403	290	2403	145	Request for oral hearing	
1451	1,510	1451	1,510	Petition to institute a public use proceeding	
1452	110	2452	55	Petition to revive – unavoidable	
1453	1,330	2453	665	Petition to revive – unintentional	
1501	1,330	2501	665	Utility issue fee (or reissue)	
1502	480	2502	240	Design issue fee	
1503	640	2503	320	Plant issue fee	
1460	130	1460	130	Petitions to the Commissioner	
1807	50	1807	50	Processing fee under 37 CFR 1.17 (q)	
1806	180	1806	180	Submission of Information Disclosure Stmt	
8021	40	8021	40	Recording each patent assignment per property (times number of properties)	
1809	770	2809	385	Filing a submission after final rejection (37 CFR § 1.129(a))	
1810	770	2810	385	For each additional invention to be examined (37 CFR § 1.129(b))	
1801	770	2801	385	Request for Continued Examination (RCE)	
1802	900	1802	900	Request for expedited examination of a design application	

Other fee (specify) ____

*Reduced by Basic Filing Fee Paid

SUBTOTAL (3) ($) 0

SUBMITTED BY Complete *(if applicable)*

Name (Print/Type)	Ross A. Dannenberg	*Registration No. (Attorney/Agent)*	49,024	*Telephone*	(202) 824-3000
Signature	[signature]			*Date*	September 13, 2004

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2001

Application or Docket Number: 000479.00082

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	20	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	20 minus 20=	* 0
INDEPENDENT CLAIMS	7 minus 3 =	* 4
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐ OR OTHER THAN SMALL ENTITY

RATE	FEE		RATE	FEE
BASIC FEE	370.00	OR	BASIC FEE	740.00
X$ 9=		OR	X$18=	
X42=		OR	X84=	336
+140=		OR	+280=	
TOTAL		OR	TOTAL	992

CLAIMS AS AMENDED - PART II

SMALL ENTITY OR OTHER THAN SMALL ENTITY

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 34	Minus	** 20	= 14
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	Pd
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875 (Rev. 8/01) ☆U.S GPO:2001 482-124/33187 Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

22907 7590 10/18/2005

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

EXAMINER	
LIM, KRISNA	
ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 10/18/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No.	Applicant(s)
	10/259,494	MUNGER ET AL.
	Examiner	Art Unit
	Krisna Lim	2153

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 September 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-34* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

1. Claims 1-34 are pending for examination.

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

3. Claims1-34 are rejected under 35 U.S.C. 102(a) as being anticipated by Sistanizadeh et al. [U.S. Patent No. 5,790,548].

4. Sistanizadch et al. anticipated the invention substantially as claimed. Taking claim 9 as an exemplary claim, the reference disclosed a method for establishing an encrypted channel between a client and a secure host , comprising the step of automatically creating the encrypted channel in response to detecting a request for access to a predetermined IP address (e.g., see the abstract, lines 8-15).

5. As to claim 10, Sistanizacdch et al. anticipated the creating step is performed in a communication protocol independently of an application program (e.g., see the abstract, lines 8-15, col. 9, lines 38-45).

6. As to claim 25, Sistanizacdch et al. anticipated automatically creating the encrypted channel comprises establishing an IP address hopping scheme between the client and the secure host (e.g., see the abstract, lines 8-15, col. 9, line 26, to col. 10, line 60).

7. As to claim 26, Sistanizacdch et al. anticipated automatically creating the encrypted channel avoid sending the true IP address (security and protection of user information privacy) of the secure host to the client (e.g., see col. 9, lines 34-46).

8. Sistanizadch et al. anticipated the invention substantially as claimed. Taking claim 7 as an exemplary claim, the reference disclosed a method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel upon intercepting a DNS request for a domain name (e.g., see col. 11, line 26, to col. 12, line 46).

9. As to claim 8, Sistanizacdch et al. anticipated the creating step is performed in a communication protocol independently of an application program (e.g., see the abstract, lines 8-15, col. 9, lines 38-45).

10. As to claim 23, Sistanizacdch et al. anticipated automatically creating the encrypted channel comprises establishing an IP address hopping scheme between the client and the secure host (e.g., see the abstract, lines 8-15, col. 9, line 26, to col. 10, line 60).

11. As to claim 24, Sistanizacdch et al. anticipated automatically creating the encrypted channel avoid sending the true IP address (security and protection of user information privacy) of the secure host to the client (e.g., see col. 9, lines 34-46).

12. Sistanizadch et al. anticipated the invention substantially as claimed. Taking claim 1 as an exemplary claim, the reference disclosed a method for establishing an encrypted channel between a client and a target computer, comprising the step of: (i) intercepting a DNS request sent by the client (e.g., see [1] of Fig. 6, col. 11, lines 56-58. DNS Request for "Host" of Fig. 7); and

(ii) based on the DNS request, establishing the encrypted channel between the client and the target (e.g., see the abstract, lines 8-15, col. 11, line 27, to col. 12, line 46).

13. As to claim 2, Sistanizacdch et al. anticipated the steps of: a) determining whether the client is authorized to access the target (e.g., see "Verification/Authentication of Fig. 7, col. 10, lines 46-47, col. 12, lines 3-46); b) when the client is authorized to access the target, initiating the encrypted channel (e.g., see the abstract, lines 8-15, col. 11, line 27, to col. 12, line 46); and c) when the client is not authorized to access the target, sending an error message to the client (see col. 11, line 27, to col. 12, line 46).

14. As to claim 3, Sistanizacdch et al. anticipated the step of send encrypted channel parameters to the client (e.g., see col. 11, line 27, to col. 12, line 46).

15. As to claim 4, Sistanizacdch et al. anticipated the creating step is performed in a communication protocol independently of an application program (e.g., see the abstract, lines 8-15, col. 9, lines 38-45).

16. As to claim 5, Sistanizacdch et al. anticipated a DNS proxy server intercepting the DNS request send by the client (e.g., see "DNS request for Host" from PC 710 to the DNS 714 of Fig. 7, col. 12, lines 3-46).

17. As to claim 6, Sistanizadch et al. anticipated establishing an encrypted channel responsive to intercepting a DNS request for a domain name comprising a predetermined domain name extensive (e.g., see col. 11, line 26, to col. 12, line 46).

18. As to claim 21, Sistanizacdch et al. anticipated automatically creating the encrypted channel comprises establishing an IP address hopping scheme

between the client and the secure host (e.g., see the abstract, lines 8-15, col. 9, line 26, to col. 10, line 60).

19. As to claim 22, Sistanizacdch et al. anticipated automatically creating the encrypted channel avoid sending the true IP address (security and protection of user information privacy) of the secure host to the client (e.g., see col. 9, lines 34-46).

20. As to claim 11-20 and 27-34, they are similar in scope as of claims 1-10 and 21-26, and therefore claims 11-20 and 27-34 are rejected for the same reasons set forth above for claims 1-10 and 21-26.

21. Applicant's arguments with respect to claims 1-34 have been considered but are moot in view of the new ground(s) of rejection.

22. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

The references are cited in the Form PTO-892 for the applicant's review.

A shortened statutory period for response to this action is set to expire 3 (three) months and 0 (zero) days from the mail date of this letter. Failure to respond within the period for response will result in **ABANDONMENT** of the application (see 35 U.S.C 133, M.P.E.P 710.02, 710.02(b)).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Krisna Lim whose telephone number is 571-272-3956 The examiner can normally be reached on Monday to Wednesday and Friday from 7:30 AM to 4:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton Burgess, can be reached on 571-272-3949. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

KI

January 19, 2005

KRISNA LIM
PRIMARY EXAMINER

Notice of References Cited	Application/Control No. 10/259,494	Applicant(s)/Patent Under Reexamination MUNGER ET AL.	
	Examiner Krisna Lim	Art Unit 2153	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-5,790,548	08-1998	Sistanizadeh et al.	370/401
	B	US-6,119,234	09-2000	Aziz et al.	713/201
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Index of Claims

Application No.	Applicant(s)
10/259,494	MUNGER ET AL.
Examiner	**Art Unit**
Krisna Lim	2153

√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	+	Restricted	I	Interference	O	Objected

Claim Final	Claim Original	Date 10/3/05	Date	Date	Date	Date	Date	Date	Date	Date
	1	✓								
	2	↓								
	3	↓								
	4	↓								
	5	↓								
	6	↓								
	7	↓								
	8	↓								
	9	↓								
	10	↓								
	11	↓								
	12	↓								
	13	↓								
	14	↓								
	15	↓								
	16	↓								
	17	↓								
	18	↓								
	19	↓								
	20	↓								
	21	↓								
	22	↓								
	23	↓								
	24	↓								
	25	↓								
	26	↓								
	27	↓								
	28	↓								
	29	↓								
	30	↓								
	31	↓								
	32	↓								
	33	↓								
	34	↓								
	35									
	36									
	37									
	38									
	39									
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49									
	50									
	51									
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	94									
	95									
	96									
	97									
	98									
	99									
	100									
	101									
	102									
	103									
	104									
	105									
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

U.S. Patent and Trademark Office

Part of Paper No. 20050119

RECEIVED
CENTRAL FAX CENTER
JAN 06 2006

Appln. No.: 10/259,494
Amendment dated September 13, 2004
Reply to Office Action of October 18, 2005

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	
Edmund Colby MUNGER *et al.*	
Serial No.:	10/259,494
Filed:	September 30, 2002
For:	Establishment of a Secure Communication Link Based on a Domain Name Service (DNS) Request (As Amended)

Group Art Unit: 2153

Examiner: Krisna Lim

Atty. Dkt. No. 000479.00082

AMENDMENT

U.S. Patent and Trademark Office
220 20th Street S.
Customer Window
Crystal Plaza Two, Lobby, Room 1B03
Arlington, VA 22202

I hereby certify that this correspondence is being facsimile transmitted to the Patent and Trademark Office on January 6, 2006, to (571) 273-8300.
Signature: /Ross Dannenberg/
Ross A. Dannenberg, Reg. No. 49,024

Sir:

In response to the Office Action mailed October 18, 2005, please amend the instant application as follows:

Amendments to the Claims are reflected in the Listing of Claims, which begins on page 2 of this paper.

Remarks/Arguments begin on page 8 of this paper.

It is believed that no fee is required for this submission. If any fees are required or if an overpayment is made, the Commissioner is authorized to debit or credit our Deposit Account No. 19-0733, accordingly.

This listing of claims is replaces all previous listings of claims provided in this application:

Listing of Claims:

1. (Original) A method for establishing an encrypted channel between a client and a target computer, comprising the steps of:

(i) intercepting a DNS request sent by the client; and

(ii) based on the DNS request, establishing the encrypted channel between the client and the target.

2. (Currently Amended) The method of claim 1, wherein step (ii) comprises steps of:

a) determining whether the client is authorized to access the target identified by the DNS request;

b) when the client is authorized to access the target, initiating the encrypted channel; and

c) when the client is not authorized to access the target, sending an error message to the client.

3. (Original) The method of claim 2, wherein step b) comprises sending encrypted channel parameters to the client.

4. (Original) The method of claim 1, wherein step (ii) occurs in a communication protocol independently of an application program.

5. (Original) The method of claim 1, wherein step (i) comprises a DNS proxy server intercepting the DNS request sent by the client.

6. (Original) The method of claim 1, wherein step (ii) comprises establishing the encrypted channel responsive to intercepting a DNS request for a domain name comprising a predetermined domain name extension.

7. (Original) A method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name extension.

8. (Original) The method of claim 7, wherein the creating step is performed in a communication protocol independently of an application program.

9. (Original) A method for establishing an encrypted channel between a client and a secure host, comprising the step of automatically creating the encrypted channel in response to detecting a request for access to a predetermined IP address.

10. (Original) The method of claim 9, wherein the creating step is performed in a communication protocol independently of an application program.

11. (Original) A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:

(i) determining whether the intercepted DNS request corresponds to a secure server;

(ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

12. (Original) The data processing device of claim 11, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

13. (Original) The data processing device of claim 12, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

14. (Original) The data processing device of claim 13, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

15. (Original) A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

16. (Original) A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) intercepting a DNS request sent by a client;

(ii) determining whether the intercepted DNS request corresponds to a secure server;

(iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

17. (Original) The computer readable medium of claim 16, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

18. (Original) The computer readable medium of claim 17, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

19. (Original) The computer readable medium of claim 18, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

20. (Original) A computer readable medium comprising computer readable instructions that, when executed, cause a domain name server (DNS) proxy module to intercept DNS requests sent by a client and, for each intercepted DNS request, when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server and, if so, automatically initiates an encrypted channel between the client and the secure server.

21. (Previously Presented) The method of claim 1, wherein step (ii) comprises establishing an IP address hopping scheme between the client and the target.

22. (Previously Presented) The method of claim 1, wherein step (ii) avoids sending a true IP address of the target to the client.

23. (Previously Presented) The method of claim 7, wherein creating the encrypted channel comprises establishing an IP address hopping scheme between the client and the secure host.

24. (Previously Presented) The method of claim 7, wherein creating the encrypted channel avoids sending a true IP address of the secure host to the client.

25. (Previously Presented) The method of claim 9, wherein automatically creating the encrypted channel comprises establishing an IP address hopping scheme between the client and the secure host.

26. (Previously Presented) The method of claim 9, wherein automatically creating the encrypted channel avoids sending a true IP address of the secure host to the client.

27. (Previously Presented) The data processing device of claim 11, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

28. (Previously Presented) The data processing device of claim 11, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

29. (Previously Presented) The data processing device of claim 15, wherein automatically initiating an encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

30. (Previously Presented) The data processing device of claim 15, wherein automatically initiating an encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

31. (Previously Presented) The computer readable medium of claim 16, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

32. (Previously Presented) The computer readable medium of claim 16, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

33. (Previously Presented) The computer readable medium of claim 20, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

34. (Previously Presented) The computer readable medium of claim 20, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

REMARKS/ARGUMENTS

The office action of October 18, 2005, has been carefully reviewed and these remarks are responsive thereto. Applicants have amended claim 2. Claims 1-34 thus remain pending in this application, and reconsideration and allowance of the instant application are respectfully requested.

Rejections Under 35 U.S.C. § 102

Claims 1-34 stand rejected under 35 U.S.C. § 102(a) as being anticipated by Sistanizadeh et al. (U.S. Pat. No. 5,790,548, hereinafter Sistanizadeh). The rejection is respectfully traversed. As the Office Action addresses the claims out of order, Applicants will respond in kind for the examiner's convenience.

The Office Action initially rejects claim 9 as an exemplary claim. Claim 9 recites, *inter alia*, "automatically creating the encrypted channel *in response to* detecting a request for access to a predetermined IP address." (emphasis added). The Office Action further alleges that Sistanizadeh describes such a feature in the abstract, at lines 8-15. However, the abstract of Sistanizadeh recites:

> providers. A domain name server (DNS) and a dynamic host configuration protocol (DHCP) server are connected to the router to provide domain name to IP address translations and temporary assignment of IP addresses to said customer premises processor terminal. The customer or subscriber going on-line communicates with the DHCP using encryption and preferably public/private key encryption to both authenticate the customer and the DHCP. The DHCP

Sistanizadeh, Abstract, lines 8-15.

As is evident from the above reproduction, Sistanizadeh makes no mention of an encrypted channel created *in response to* detecting a request for access to a predetermined IP address. Instead, Sistanizadeh describes a system whereby the initial DNS request is made over an encrypted channel, not creating the encrypted channel in response to the DNS request. The two are quite inapposite from each other. Applicants have reviewed Sistanizadeh and respectfully submit that Sistanizadeh does not teach or suggest automatically creating the encrypted channel in response to detecting a request for access to a predetermined IP address, as recited in claim 9, at the cited portions or elsewhere. As such, the Office Action fails to establish a *prima facie* rejection of claim 9 under 35 U.S.C. § 102(a).

Claims 10, 25, and 26, dependent back to claim 9, are patentable over Sistanizadeh for at least the same reasons as independent claim 9.

In addition, with respect to claim 10, because Sistanizadeh does not teach or suggest the claimed creating step, Sistanizadeh does not teach or suggest performing the creating step in a communication protocol independently of an application program. Indeed, at the cited location of the reference, Sistanizadeh merely describes retrieving subscription information and updating this information using Simple Network Management Protocol, which is wholly dissimilar from automatically creating the encrypted channel in response to detecting a request for access to a predetermined IP address, using a communication protocol independently of an application program, as claimed.

With respect to claim 25, Sistanizadeh does not teach or suggest an IP address hopping scheme as claimed, but rather merely describes conventional server pooling.

With respect to claim 26, Sistanizadeh is deficient as well. Again, at the cited location, Sistanizadeh merely describes retrieving subscription information and updating this information using Simple Network Management Protocol, which is wholly dissimilar from avoiding sending a true IP address of the secure host to the client. Likewise, Sistanizadeh does not cure this deficiency elsewhere.

The Office Action next addresses claim 7. Applicants respectfully submit that claim 7 is allowable at least for similar reasons as claim 9, as the additionally cited portions of Sistanizadeh are deficient for the same reasons already discussed above. For example, nowhere in, col. 11, line 26 – col. 12, line 46, does Sistanizadeh discuss or even mention the creation of an encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name *extension*.

Claims 8, 23, and 24, dependent back to claim 7, are patentable over Sistanizadeh for at least the same reasons as independent claim 7.

In addition, claim 8 is further allowable for similar reasons as claim 10, discussed above.

Claim 23 is further allowable for similar reasons as claim 25, discussed above.

Claim 24 is further allowable for similar reasons as claim 26, discussed above.

The Office Action next addresses claim 1, which recites, *inter alia*, "intercepting a DNS request sent by the client; and based on the DNS request, establishing the encrypted channel between the client and the target." The Office Action cites col. 11, lines 56-58, which merely recites a conventional DNS request, and col. 11, line 27 – col. 12, line 46, already discussed above, as describing such features. Applicants therefore respectfully submit that claim 1 is not anticipated by Sistanizadeh for at least the same reasons as discussed above with respect to claims 7 and 9.

Claims 2-6 and 21-22, dependent back to claim 1, are patentable over Sistanizadeh for at least the same reasons as independent claim 1.

In addition, claim 2 is not anticipated by Sistanizadeh because Sistanizadeh does not teach or suggest determining whether the client is authorized to access the target computer identified by the DNS request.

Claim 4 is further allowable for similar reasons as claim 10.

Claim 6 is further allowable for similar reasons as claim 7.

Claim 21 is further allowable for similar reasons as claim 25.

Claim 22 is further allowable for similar reasons as claim 26.

The Office Action summarily rejects claims 11-20 and 27-34, alleging "they are similar in scope as of claims 1-10 and 21-26." However, such is not the case. For example, independent claim 11 recites, inter alia, "determining whether the intercepted DNS request corresponds to a secure server; when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server." Such features are not recited in independent claims 1, 7, or 9. The rejection of claim 11 is therefore deficient because the Office Action therefore fails to address all the recitations of claim 11. Claims 12-15 and 27-28 depend back to claim 11 and are allowable for at least the same reason.

Claim 16 recites, *inter alia*, "determining whether the intercepted DNS request corresponds to a secure server," which is different from the recitations of claims 1-10 and 21-26. The rejection of claim 16 and dependent claims 17-19 and 31-32 is therefore deficient as well.

Claim 20 recites, *inter alia*, "when the intercepted DNS request corresponds to a secure server, determines whether the client is authorized to access the secure server," which is different from the recitations of claims 1-10 and 21-26. The rejection of claim 20 and dependent claims 33-34 is therefore deficient as well.

CONCLUSION

All rejections having been addressed, applicant respectfully submits that the instant application is in condition for allowance, and respectfully solicits prompt notification of the same. However, if for any reason the Examiner believes the application is not in condition for allowance or there are any questions, the examiner is requested to contact the undersigned at (202) 824-3153.

Respectfully submitted,

BANNER & WITCOFF, LTD.

Dated this 6th day of Jan., 2006

By: /Ross Dannenberg/
Ross Dannenberg, Registration No. 49,024

1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel: (202) 824-3000
Fax: (202) 824-3001

RAD/mmd

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2001

Application or Docket Number: 000479.00082

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	20	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	20 minus 20=	* 0
INDEPENDENT CLAIMS	7 minus 3 =	* 4
MULTIPLE DEPENDENT CLAIM PRESENT	☐	

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐ OR OTHER THAN SMALL ENTITY

SMALL ENTITY RATE	SMALL ENTITY FEE		OTHER THAN SMALL ENTITY RATE	OTHER THAN SMALL ENTITY FEE
BASIC FEE	370.00	OR	BASIC FEE	740.00
X$ 9=		OR	X$18=	
X42=		OR	X84=	336
+140=		OR	+280=	
TOTAL		OR	TOTAL	992

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 34	Minus	** 20	= 14
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM	☐			

SMALL ENTITY RATE	ADDITIONAL FEE		OTHER THAN SMALL ENTITY RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	Pd
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

1/6/06

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 34	Minus	** (crossed out)	= (crossed out)
Independent	* 7	Minus	*** (crossed out)	= (crossed out)
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM	☐			

SMALL ENTITY RATE	ADDITIONAL FEE		OTHER THAN SMALL ENTITY RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	*	Minus	**	=
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM	☐			

SMALL ENTITY RATE	ADDITIONAL FEE		OTHER THAN SMALL ENTITY RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875 (Rev. 8/01) Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

22907 7590 03/28/2006

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

EXAMINER	
LIM, KRISNA	
ART UNIT	**PAPER NUMBER**
2153	

DATE MAILED: 03/28/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No. 10/259,494	Applicant(s) MUNGER ET AL.
	Examiner Krisna Lim	Art Unit 2153

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE _3_ MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _the amendmend filed 1/6/06_.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1-34_ is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-34_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

1. Claims 1-34 are still presented for examination.

2. The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Sistanizadeb et al. [U.S. Patent No. 5,790,548]. Examiner used this reference to reject all claims under 35 U.S.C. § 102 (a) but now Examiner uses this reference to reject all claims under 35 U.S.C. § 103(a)

4. In addition to the previous rejection, Examiner would like add additional statements as following:

While Sistanizadeh discloses the use of an encrypted channel and preferably public/private key encryption to both authenticate the customer and a dynamic host configuration protocol (DHCP), Sistanizadeb does not explicitly mention that this encrypted channel is automatically created in response to the DNS request. Since Sistanizadeb discloses the use of the encrypted channel between a client and a target for allow a client (e.g., a customer or a subscriber) to communication on-line with others, it would have been obvious to one of ordinary skill in the art to recognize that Sistanizadeb would have been obviously disclosed the feature of creation of the encrypted channel while specific circumstance of when to created this encrypted channel would have been a matter of choice. And, this specific circumstance would not be a patentably distinguishable.

5. Applicant's arguments filed 1/6/06 have been fully considered but they are not deemed to be persuasive.

In the remarks, applicants argued in substance that Sistanizadeb makes no mention of the encrypted channel is automatically created in response to the DNS request. In reply to, Sistanizadeh do discloses the use of an encrypted channel and preferably public/private key encryption to both authenticate the customer and a dynamic host configuration protocol (DHCP), Sistanizadeb, however, does not explicitly mention that this encrypted channel is automatically created in response to the DNS request. Moreover, since Sistanizadeb discloses the use of the encrypted channel between a client and a target for allow a client (e.g., a customer or a subscriber) to communication on-line with others, it would have been obvious to one of ordinary skill in the art to recognize that Sistanizadeb would have been obviously disclosed the feature of creation of the encrypted channel while specific circumstance when to created this encrypted channel would have been a matter of choice. And, this specific circumstance would not be a patentably distinguishable.

A shortened statutory period for response to this action is set to expire 3 (three) months and 0 (zero) days from the mail date of this letter. Failure to respond within the period for response will result in **ABANDONMENT** of the application (see 35 U.S.C 133, M.P.E.P 710.02, 710.02(b)).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Krisna Lim whose telephone number is 571-272-3956 The examiner can normally be reached on Monday to Wednesday and Friday from 7:30 AM to 4:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton Burgess, can be reached on 571-272-3949. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

KI

March 20, 2006

KRISNA LIM
PRIMARY EXAMINER

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination
10/259,494	MUNGER ET AL.
Examiner	**Art Unit**
Krisna Lim	2153

x	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim Final	Claim Original	Date 3/20/06	Date	Date	Date	Date	Date	Date	Date	Date
	1	x								
	2	x								
	3	x								
	4	x								
	5	x								
	6	x								
	7	x								
	8	x								
	9	x								
	10	x								
	11	x								
	12	x								
	13	x								
	14	x								
	15	x								
	16	x								
	17	x								
	18	x								
	19	x								
	20	x								
	21	x								
	22	x								
	23	x								
	24	x								
	25	x								
	26	x								
	27	x								
	28	x								
	29	x								
	30	x								
	31	x								
	32	x								
	33	x								
	34	x								
	35									
	36									
	37									
	38									
	39									
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49									
	50									

Claim Final	Claim Original	Date	Date	Date	Date	Date	Date	Date	Date	Date
	51									
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	94									
	95									
	96									
	97									
	98									
	99									
	100									

Claim Final	Claim Original	Date	Date	Date	Date	Date	Date	Date	Date	Date
	101									
	102									
	103									
	104									
	105									
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2001

Application or Docket Number: 000479.00082

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	20	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	20 minus 20=	* 0
INDEPENDENT CLAIMS	7 minus 3 =	* 4
MULTIPLE DEPENDENT CLAIM PRESENT		☐

* If the difference in column 1 is less than zero, enter "0" in column 2

SMALL ENTITY TYPE ☐ OR OTHER THAN SMALL ENTITY

SMALL ENTITY RATE	FEE		OTHER THAN SMALL ENTITY RATE	FEE
BASIC FEE	370.00	OR	BASIC FEE	740.00
X$ 9=		OR	X$18=	
X42=		OR	X84=	336
+140=		OR	+280=	
TOTAL		OR	TOTAL	952

CLAIMS AS AMENDED - PART II

AMENDMENT A

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 34	Minus	** 20	= 14
Independent	*	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY RATE	ADDITIONAL FEE		OTHER THAN SMALL ENTITY RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	Ød
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT B

1/6/06

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 34	Minus	**	=
Independent	* 7	Minus	***	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

AMENDMENT C

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 34	Minus	** 34	=
Independent	* 7	Minus	*** 7	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X42=		OR	X84=	
+140=		OR	+280=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875 (Rev. 8/01) Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

RECEIVED
CENTRAL FAX CENTER
JUL 2 7 2006

Appln. No.: 10/259,494
Amendment dated July 27, 2006
Reply to Office Action of March 28, 2006

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re the Application of (first named inventor):		Atty. Docket No.:	000479.00082
Edmund Colby MUNGER ***et al.***			
Serial No.:	10/259,494	Group Art Unit:	2153
Filed:	September 30, 2002	Examiner:	Krisna Lim
For:	Establishment of a Secure Communication Link Based on a Domain Name Service (DNS) Request (As Amended)	Confirmation No.:	5257

REQUEST FOR RECONSIDERATION

Mail Stop Amendment
U.S. Patent and Trademark Office
Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

I hereby certify that this correspondence is being facsimile transmitted to the Patent and Trademark Office on July 27, 2006, to (571) 273-8300.
Signature: /Ross Dannenberg/
Ross A. Dannenberg, Reg. No. 49,024

Sir:

In response to the non-final Office Action mailed March 28, 2006, Applicants provide the following remarks.

Remarks/Arguments begin on page 2 of this paper.

Applicants hereby petition for any requisite extension of time, and authorize the Office to charge extension of time fees, as well as any other fees associated with this response, to our Deposit Account No. 19-0733.

REMARKS/ARGUMENTS

The Office Action of March 28, 2006 has been carefully reviewed and these remarks are responsive thereto. Reconsideration and allowance of the instant application are respectfully requested in view of the following.

Rejections Under 35 U.S.C. § 103

Claims 1-34 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Pat. No. 5,790,548, hereinafter Sistanizadeb. Applicants traverse.

The Office Action indicates that claims 1-34 are now rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Pat. No. 5,790,548, hereinafter Sistanizadeb, whereas the claims previously were rejected under §102 as being anticipated by Sistanizadeb. The Office Action states:

> While Sistanizadeh discloses the use of an encrypted channel and preferably public/private key encryption to both authenticate the customer and a dynamic host configuration protocol (DHCP), Sistanizadeb does not explicitly mention that this encrypted channel is automatically created in response to the DNS request. Since Sistanizadeb discloses the use of the encrypted channel between a client and a target for allow a client (e.g., a customer or a subscriber) to communication on-line with others, it would have been obvious to one of ordinary skill in the art to recognize that Sistanizadeb would have been obviously disclosed the feature of creation of the encrypted channel while specific circumstance of when to created this encrypted channel would have been a matter of choice. And, this specific circumstance would not be a patentably distinguishable.

Office Action, p. 2.

The above-quoted portion of the Office Action is the ENTIRE argument for the rejection of all pending claims 1-34. Applicants respectfully submit that such a rejection is deficient on its face insofar as it fails to address all the limitations of even one claim, and also fails to specifically address, in detail, each pending claim in the application, as required by 37 C.F.R. § 1.104(b) and MPEP § 7.07. Furthermore, the Office's argument could ostensibly be used to reject claims of an unlimited scope, as every invention involves matters of design choice. However, design choice does not render every invention obvious.

While the Action alludes to rejections in a previous Office Action, the Office Action does not indicate how or why one of skill in the art would be motivated to modify Sistanizadeb to arrive at any of the pending claims, and the Office Action fails to address specific recitations of the claims. For example, with respect to claim 6, the Office Action does not indicate how or why the initiation of an encrypted channel responsive to intercepting a DNS request for a domain name comprising a predetermined domain name extension is taught or suggested by Sistanizadeb. Sistanizadeb, instead, describes DNS lookups of known domain names, not *intercepting* a DNS request for a domain name comprising a predetermined domain name extension, as claimed.

With respect to claim 7, the Office Action does not indicate how or why Sistanizadeb teaches or suggests automatically creating the encrypted channel upon intercepting a DNS request for a domain name comprising a predetermined domain name extension.

With respect to claim 9, the Office Action does not indicate how or why Sistanizadeb teaches or suggests automatically creating the encrypted channel in response to detecting a request for access to a predetermined IP address.

With respect to claim 15 and 20, the Office Action does not indicate how or why Sistanizadeb teaches or suggests memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client.

With respect to claim 18, the Office Action does not indicate how or why Sistanizadeb teaches or suggests, when the client is not authorized to access the secure server, returning a host unknown error message to the client

With respect to claim 21, 23, 25, 27, 29, 31, and 33, the Office Action does not indicate how or why Sistanizadeb teaches or suggests establishing an IP address hopping scheme between the client and the target. Indeed, Sistanizadeb makes no mention whatsoever of any sort of IP hopping scheme. The Office Action cites portions of Sistanizadeb that describe IP address translation from a domain name to an IP address, not an IP address hopping scheme between the client and the target, as claimed.

With respect to claim 22, 24, 26, 28, 30, 32, and 34, the Office Action does not indicate how or why Sistanizadeb teaches or suggests step (ii) avoids sending a true IP address of the target to the client.

Applicants respectfully submit that all claims are allowable over the art of record in this case, and respectfully submit that, should another Office Action be issued, that such Action cannot properly be made final in view of the deficiencies in the Office Action mailed March 28, 2006.

CONCLUSION

All rejections having been addressed, Applicant respectfully submits that the instant application is in condition for allowance, and respectfully solicits prompt notification of the same. However, if for any reason the Examiner believes the application is not in condition for allowance or there are any questions, the Examiner is requested to contact the undersigned at (202) 824-3153.

Respectfully submitted,

BANNER & WITCOFF, LTD.

Dated this 27th day of July, 2006

By: /Ross Dannenberg/

Ross Dannenberg, Registration No. 49,024
1001 G Street, N.W.
Washington, D.C. 20001-4597
Tel: (202) 824-3000
Fax: (202) 824-3001

RAD/mmd

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 08/02/2006

FBRISCOE SALE #00000002 Mailroom Dt: 07/27/2006 190733 10259494
01 FC:1251 120.00 DA

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2006	encrypt$3 adj4 channel	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/16 11:34
L2	122	(establish$3 or creat$3) adj4 encrypt$3 adj4 channel	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/16 11:35
L3	21	l2 and DNS	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/16 11:36
L4	0	l3 and @ad<="11981030"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/16 11:37
L5	0	l3 and @ad<="19981030"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/16 11:37
L6	0	l3 and @ad<="19981030"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/10/16 11:37

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

22907 7590 10/30/2006

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

EXAMINER	
LIM, KRISNA	
ART UNIT	**PAPER NUMBER**
2153	

DATE MAILED: 10/30/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

Office Action Summary	Application No. 10/259,494	Applicant(s) MUNGER ET AL.	
	Examiner Krisna Lim	Art Unit 2153	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 July 2006*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-34* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *11-14,16-19,27,28,31 and 32* is/are allowed.

6)☒ Claim(s) *1-10 and 15* is/are rejected.

7)☒ Claim(s) *21-26, 29-30, 33 and 34* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application

6) ☐ Other: _____.

1. Claims 1-34 are still pending for examination.

2. Claims 1, 7 and 9 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

In claim 1, it is not clearly understood how the encrypted channel is established between the client and the target based on the DNS request.

In claim 7, it is not clearly understood how the encrypted channel is created automatically upon intercepting a DNS request for a domain name comprising a predetermined domain name extension.

In claim 9, it is not clearly understood how the encrypted channel is created automatically in response to detecting a request for access to a predetermined IP address.

3. The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims are rejected under 35 U.S.C. § 103(a) as being unpatentable over Sistanizadeh et al. [U.S. Patent No. 6,101,182].

Sistanizadeh et al. disclose (e.g., see Figs. 1-22) the invention substantially as claimed. Taking claim 1 as an exemplary claim, the reference discloses a method for establishing an encrypted channel between a client and a target computer, comprising the steps of:

a) intercepting a DNS request sent by the client (e.g., see Figs. 6 and 8b).

b) based on the DNS request, establishing the encrypted channel between the client and the target (e.g., see Figs. 6 and 8b, col. 2, lines 43-44, col. 6, lines 62-64, col. 7, lines 33-35, col. 10, lines 62-67).

While Sistanizadeh discloses: a) IP addresses DNS are provided and connected to routers in the network (e.g., see col. 6, lines 62-67); b) the establish session of DNS (e.g., see col. 6, lines 62-67); c) requesting authentication (e.g., see col. 9, lines 65-66), d) encryption and authentication between the server and customer computer (e.g., see col. 10, lines 46-47), Sistanizadeh did not explicitly mention that the encrypted channel is established between the client and the target.

It would have been obvious to one of ordinary skill in the art to recognize that Sistanizadeh's teaching of encryption and authentication between the server and the customer computer would obviously teach the feature of establishing the encrypted channel between the client and the target.

5. As to claim 2, Sistanizadeh further disclosed the steps of: a) determining whether the client is authorized to access the target identified by the DNS request; b) when the client is authorized to access the target, initiating teh encrypted channel; and when the client is not authorized to access the target, sending an error message to the client (e.g., see verification and authentication of Fig. 7, encryption and authentication between the server and customer computer at col. 10, lines 46-47).

6. As to claim 3, Sistanizadeh further disclosed the step of sending encrypted channel parameterto the client (e.g., see the functions of router 330 of Fig. 3, encryption and authentication between the server and customer computer (e.g., see col. 10, lines 46-47).

7. As to claim 4, Sistanizadeh further disclosed a communication protocol independently of an application program (e.g., see col. 3, line 52, to col. 4, line 20).

8. As to claim 5, Sistanizadeh further disclosed a DNS proxy server (DHCP server 334, 348 and 350).

9. As to claim 6, the feature of predetermined domain name extension. It would have been obvious to one of ordinary skill in the art to recognize that such predetermined domain name extention (e.g., .gov, .org, etc.) is well known feature in the art.

10. Claims 7-10, 15 and 20 are rejected for the same rationale as claims 1-6, since they recite substantially identical subject matter. Any differences between the claims do not result in patentably distinct claims and all of the limitations are taught by the above cited references.

11. Objected dependent claims 21-26, 29-30, 33 and 34

12. Claims 11-14, 16-19, 27, 28, 31 and 32 are allowable.

13. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

The references are cited in the Form PTO-892 for the applicant's review.

A shortened statutory period for response to this action is set to expire 3 (three) months and 0 (zero) days from the mail date of this letter. Failure to respond within the period for response will result in **ABANDONMENT** of the application (see 35 U.S.C 133, M.P.E.P 710.02, 710.02(b)).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Krisna Lim whose telephone number is 571-272-3956 The examiner can normally be reached on Monday to Wednesday and Friday from 7:30 AM to 4:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton Burgess, can be reached on 571-272-3949. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kl

October 14, 2006

KRISNA LIM
PRIMARY EXAMINER

Notice of References Cited	Application/Control No. 10/259,494	Applicant(s)/Patent Under Reexamination MUNGER ET AL.	
	Examiner Krisna Lim	Art Unit 2153	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
*	A	US-5,898,830	04-1999	Wesinger et al.	726/15
*	B	US-6,101,182	08-2000	Sistanizadeh et al.	370/352
*	C	US-6,502,135	12-2002	Munger et al.	709/225
*	D	US-2006/0059337	03-2006	Poyhonen et al.	713/165
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes	Application/Control No. 10/259,494	Applicant(s)/Patent under Reexamination MUNGER ET AL.	
	Examiner Krisna Lim	Art Unit 2153	

SEARCHED

Class	Subclass	Date	Examiner

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
EAST TEXT SEARCH	10/16/2006	KL
Inventor Name Search	10/16/2006	KL

U.S. Patent and Trademark Office

Part of Paper No. 20061016

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10259494
	Filing Date	2002-09-30
	First Named Inventor	Edmund Colby Munger
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00082

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code1	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	5870610	A	1999-02-09	Beyda et al.	

If you wish to add additional U.S. Patent citation information please click the Add button.

U.S.PATENT APPLICATION PUBLICATIONS

Examiner Initial*	Cite No	Publication Number	Kind Code1	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button.

FOREIGN PATENT DOCUMENTS

Examiner Initial*	Cite No	Foreign Document Number3	Country Code2 j	Kind Code4	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T5
	1	0 838 930	EP	A	1998-04-29	Compaq Computer Corp.		☐
	2	0 814 589	EP	A	1997-12-29	AT&T Corp.		☐
	3	2 334 181	GB	A	1999-08-11	NEC Technologies; Globalmart Ltd.		☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10259494
	Filing Date	2002-09-30
	First Named Inventor	Edmund Colby Munger
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00082

	4	9827783	WO	A	1998-06-25	Northern Telecom Limited; Antonio, G; Hui, Margare		☐
	5	2 317 792	GB	A	1998-04-01	Secure Computing Corporation		☐
	6							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button

NON-PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T5
	1	Laurie Wells (LANCASTERBIBELMAIL MSN COM); "Subject: Security Icon" USENET NEWSGROUP, 19 October 1998, XP002200606	☐
	2	Davila J et al, "Implementation of Virtual Private Networks at the Transport Layer", Information Security, Second International Workshop, ISW '99. Proceedings (Lecture Springer-Verlag Berlin, Germany, [Online] 1999, pages 85-102, XP002399276, ISBN 3-540-66695-B, retrieved from the Internet: URL: http://www.springerlink.com/content/4uac0tb0heccma89/fulltext.pdf> (Abstract)	☐
	3		☐
	4		☐

If you wish to add additional non-patent literature document citation information please click the Add button

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

EFS Web 2.0

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	10259494
	Filing Date	2002-09-30
	First Named Inventor	Edmund Colby Munger
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00082

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10259494
	Filing Date	2002-09-30
	First Named Inventor	Edmund Colby Munger
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00082

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

☒ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☐ Fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

☐ None

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/Steve Chang/	Date (YYYY-MM-DD)	2006-11-09
Name/Print	Steve S. Chang	Registration Number	42,402

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

EFS Web 2.0

WO9827783

Publication Title:

VIRTUAL PRIVATE NETWORK SERVICE PROVIDER FOR ASYNCHRONOUS TRANSFER MODE NETWORK

Abstract:

Abstract of WO9827783

A virtual private network service provider is used to transfer data over a data network to a final destination, with third-party billing. The method comprises the steps of: prompting the user at a data terminal to select a destination, password, and call type; sending a set-up message to the data network; selecting a virtual private network provider through the data network; the virtual pri 104d vate network provider giving an encryption key to the user, and then prompting the user for a password and a user identification; encrypting the password, and sending the user identification and the encrypted password to the virtual private network provider; the virtual private network provider decrypting the encrypted password, and verifying the password; the virtual private network provider providing an authorization code; and the data terminal transferring the data through the data network to the final destination, using the authorization code.

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04Q 11/04, H04L 12/22

A1

(11) International Publication Number: WO 98/27783

(43) International Publication Date: 25 June 1998 (25.06.98)

(21) International Application Number: PCT/IB97/01563

(22) International Filing Date: 12 December 1997 (12.12.97)

(30) Priority Data:
08/769,649 19 December 1996 (19.12.96) US

(71) Applicant *(for all designated States except US)*: NORTHERN TELECOM LIMITED [CA/CA]; World Trade Center of Montreal, 8th floor, 380 St. Antoine Street West, Montreal, Quebec H2Y 3Y4 (CA).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: TELLO, Antonio, G. [US/US]; 114 Fountain Hills Drive, Garland, TX 75044 (US). HUI, Margaret [US/US]; 9920 Forest Lane #208, Dallas, TX 75243 (US). HOLMES, Kim [US/US]; 5409 Scenic Drive, Rowlett, TX 75088 (US).

(74) Agents: MCCOMBS, David et al.; Haynes and Boone, L.L.P., Suite 3100, 901 Main Street, Dallas, TX 75202-3789 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) **Title:** VIRTUAL PRIVATE NETWORK SERVICE PROVIDER FOR ASYNCHRONOUS TRANSFER MODE NETWORK

(57) Abstract

A virtual private network service provider is used to transfer data over a data network to a final destination, with third-party billing. The method comprises the steps of: prompting the user at a data terminal to select a destination, password, and call type; sending a set-up message to the data network; selecting a virtual private network provider through the data network; the virtual private network provider giving an encryption key to the user, and then prompting the user for a password and a user identification; encrypting the password, and sending the user identification and the encrypted password to the virtual private network provider; the virtual private network provider decrypting the encrypted password, and verifying the password; the virtual private network provider providing an authorization code; and the data terminal transferring the data through the data network to the final destination, using the authorization code.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

VIRTUAL PRIVATE NETWORK SERVICE PROVIDER FOR ASYNCHRONOUS TRANSFER MODE NETWORK

Technical Field

The invention relates generally to asynchronous transfer mode ("ATM") networks and virtual private networks ("VPN"), such as those offered by MCI and Sprint, and, more particularly, to a method of using a VPN to transfer data over a data network, with third-party billing.

Background of the Invention

Telephone service providers offer third-party billing. For example, local and long distance telephone companies offer calling cards for third party billing.

VPNs exist to provide the sense of a private network among a company's locations. The lines/trunks of a VPN are actually shared among several companies, to reduce costs, yet to each company the VPN appears to be that company's own private network. However, a user at a remote data terminal, such as a portable computer in a hotel room, can not immediately charge his company for the access time to a data net, such as the Internet. Instead, his access time is charged to his hotel room, and so he must pay the inflated rates that hotels charge for phone service.

What is needed is a VPN service provider that offers remote access for users belonging to a VPN, user authorizations to prevent delinquent access into the VPN, and convenient third-party billing.

Summary of the Invention

The present invention, accordingly, provides a system and method for using a VPN service provider to transfer data over a data network to a final destination, with third-party billing. The method comprises the steps of: prompting the user at a data terminal to select a destination, password, and call type; selecting a VPN through the data network; giving an encryption key to the user, and then prompting the user for a password and a user identification; verifying the password, and providing an authorization code to the user; and allowing the user to transfer the data through the data network to the final destination, using the authorization code.

In another feature of the invention, the method further comprises negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

In another feature of the invention, the method further comprises encrypting the user's password, and sending the user identification and the encrypted password to the VPN service provider.

In another feature of the invention, the method further comprises a step of sending a set-up message to the data network.

In another feature of the invention, the method further comprises a step of the VPN service provider decrypting the encrypted password.

A technical advantage achieved with the invention is that it shifts or defers costs from an end user to a bulk purchaser of data network services. Another technical advantage achieved with the invention is that it permits end users mobility while attaining a virtual appearance on a corporate intranet.

Brief Description of the Drawings

Fig. 1 is a system block diagram of a VPN service provider of the present invention.

Fig. 2 is a flow chart depicting the method of the present invention, as implemented by application software on a user terminal.

Fig. 3 is the initial screen display of the user interface of the application software.

Figs. 4A and 4B are call flow diagrams, illustrating the preferred sequence of steps of the method of the present invention.

Figs. 5A, 5B, 5C, 5D, 5E, and 5F comprise a flow chart depicting the method of the present invention, as implemented by switching control point software.

Description of the Preferred Embodiment

In Fig. 1, the VPN service provider system of the present invention is designated generally by a reference numeral 10. The VPN service provider system 10 includes a VPN 12. The VPN 12 may be a corporate, government, association, or other organization's telephone/data line `network. The VPN service provider system 10 also includes access lines 13 from the VPN 12 to a data network 14, such as the Internet, or an ATM network. The VPN service provider system 10 also includes access lines 16 from the data network 14 to a long distance phone company 18, such as AT&T, MCI, or Sprint. The VPN service provider system 10 also includes access lines 20 from the data network 14 to a called party 22, such as, for example, American Express reservations service. The VPN service provider system 10 also includes access lines 24 from the data network 14 to a remote user terminal 26, such as a portable computer in a hotel room. The user terminal 26 includes user application software 28, which provides the interface for the user to enter the number to be called, the user identification number, and the user's authorization code. The VPN service provider system 10 also includes VPN service provider software 30, located in a switching control point (SCP) device 32, which, in the preferred embodiment may be physically located anywhere. The SCP 32 connects to the data network 14 via access lines 36. One possible physical location for the SCP 32 is on the premises of a local phone company central switch building 34. However, even when located within the building 34, the SCP 32 connects to the local phone company switches via the data network 14. The local phone company switches connect to the data network 14 via access lines 38.

In an alternate embodiment, the VPN service provider software 30 and the SCP device 32 may be located on the premises of an independent provider of local phone service, or on the premises of an independent VPN service provider.

Referring now to Fig. 2, the application software 28 begins the data transfer process in step 50. In step 52, the user is presented with a screen display.

Referring now to Fig. 3, a screen display 100 displays the following information requests: whether the call is a direct call 102 or a VPN call 104, the number the user desires to call 106, the VPN user ID 108, and the user password 110. The user is also presented with the option to make the call 112, or to quit 114.

Referring back to Fig. 2, in step 54 the user terminal sends to the SCP 32 the information captured through the graphical user interface ("GUI") in step 52 within a user network interface ("UNI") setup message. In step 56 the user terminal 26 waits for a connect message from the SCP 32. In step 58 the user terminal 26 determines if a connection was made. If no connection was made, then in step 60 the user application software 28 displays an error message to the user, and returns to step 50 to begin again the data transfer process.

If a connection was made, then in step 62 the user terminal 26 sends the VPN user ID to the SCP 32. In step 64 the user terminal 26 waits for an encryption key from the SCP 32. In step 66, having received the encryption key from the SCP 32, the user application software 28 encrypts the user's password, and sends it to the SCP 32. In step 68 the user terminal 26 waits for authentication of the user. In step 70 the user application software 28 determines if the SCP 32 authorizes the user to make the call.

If the user is not authorized, then in step 72 the user terminal 26 displays an error message, terminates the connection, blanks the screen display 100, and returns to step 50 to begin again the data transfer process. If the user is authorized, then in step 74 the VPN service provider software 30 sets up the billing, and authorizes it. In step 76 the user terminal 26 sends a "release", meaning to terminate or disconnect the connection, to the SCP 32. In step 78 the user terminal 26 sends a setup message to the number listed by the user as the "number to call", that is, to the final destination. In step 80 the user terminal 26 waits for a connection. In step 82 the user terminal 26 determines if a connection was made.

If a connection to the final destination was not made, then the user application software 28 returns to step 72, in which step the user terminal 26 displays an error message, terminates the connection, blanks the screen display 100, and returns to step 50 to begin again the data transfer process. If a connection to the final destination was made, then in step 84 the user terminal 26 exchanges user data, services, and/or value added or user specific applications with the computer at the address, that is, the telephone number, of the final destination. In step 86 the user selects the option presented to him to release, or terminate, the call. In step 88 the user terminal 26 sends a release message to the final destination. In step 90 the data network 14 sends billing information to the SCP 32. In step 92 the application software 28 ends the data transfer process.

Fig. 4A and Fig. 4B are call flow diagrams, showing the sequence of messages in the method of the preferred embodiment. These diagrams present the same method as the flow chart of Fig. 2. The horizontal arrows represent the messages sent and received. The vertical lines represent the various devices involved in sending and receiving the messages. For example, the top left arrow in Fig. 4A represents a message sent from the user terminal 26, labeled "Macintosh" in Fig.4A, to an interface with a public network. The user terminal 26 can be any brand of a work station computer, a desktop computer, a laptop computer, or even a notebook computer. The interface could be any interface, but in the example of Fig. 4A and Fig. 4B, the interface is imagined to be at a hotel, where a business traveler is using the method of the present invention. Thus, the interface is labeled "Hotel ATM Interface", which is not shown in Fig. 1. The vertical line labeled "Public ATM Network" is the same as the data network 14 in Fig. 1. The vertical line labeled "Moe's VPN Service" represents the VPN service provider software 30 within the SCP 32. The vertical line labeled "Travel ATM Interface" is not shown in Fig. 1, but is located between the called party 22 and the data network 14. The vertical line labeled "Travel Service" is one example of the called party 22 shown in Fig. 1. In the example of Fig. 4A and Fig. 4B, the business traveler is imagined to be using the method of the present invention to contact a travel service to make reservations for his next airline flight. In Figs. 4A and 4B the designation "Ack" represents "acknowledge", and the designation "Cmp" represents "complete".

Referring now to Fig. 5, the VPN service provider software 30 begins the data transfer process in step 300 by waiting for an event. The event it waits for is a setup message on a signaling port of the SCP 32, to be received from the user terminal 26. In step 302, having monitored the signaling ports, and the SCP 32 having received a setup message, the VPN service provider software 30 assigns a call condense block ("CCB") to the setup message, based on a call reference number. The CCB is a software data structure for tracking resources associated with the call. The call reference number is a number, internal to the SCP, for tracking calls. In step 304 the VPN service provider software 30 compiles the connect message. In step 306 the VPN service provider software 30 sends a connect message to the calling address, that is, the hotel room from which the user is calling. In step 308 the VPN service provider software 30 condenses, that is, it remains in a wait state for that call.

Referring now to Fig. 5B, in step 310 the VPN service provider software 30 waits for an event by monitoring the signaling ports of the SCP 32. After the SCP 32 receives a connect acknowledge message from the user terminal 26, then in step 312 the VPN service provider software 30 accesses the CCB, based on the call reference number. In step 314 the VPN service provider software 30 condenses.

Referring now to Fig. 5C, in step 316 the VPN service provider software 30 waits for dialog on a data port of the SCP 32. After the SCP 32 receives a

VPN ID on a data port, the VPN service provider software 30 verifies the VPN ID in step 318. In step 320 the VPN service provider software 30 determines if the VPN ID is valid. If the VPN ID is not valid, then in step 322 the SCP 32 sends a reject message over an assigned switch virtual circuit ("SVC"). The SVC is a channel over the data network 14. In step 324 the VPN service provider software 30 waits for dialog. In step 326, because the VPN ID is valid, the VPN service provider software 30 assigns an encryption key to the user terminal 26, in step 328 sends the encryption key over the assigned SVC to the user terminal 26, and in step 330 waits for dialog.

Referring now to Fig. 5D, in step 332 the VPN service provider software 30 waits for dialog. When the SCP 32 receives the encrypted password from the user terminal 26 at a data port, then in step 334 the VPN service provider software 30 verifies the password, and determines in step 336 if the password is valid. If the password is not valid, then in step 338 the SCP 32 sends a reject message over the assigned SVC to the user terminal, and in step 340 waits for dialog. If the password is valid, then in step 342 the VPN service provider software 30 assigns an authorization token to the user terminal 26, in step 344 sends the token over an assigned SVC to the user terminal 26, and in step 346 waits for dialog.

Referring now to Fig. 5E, in step 348 the VPN service provider software 30 waits for an event. When the VPN service provider software 30 senses that the SCP 32 has received on a signaling port a release message from the user terminal 26, then in step 350 the VPN service provider software 30 accesses the CCB, based on the call reference number of the user terminal 26, in step 352 compiles a release complete message, in step 354 sends a release complete message to the user terminal 26, and in step 356 condenses.

Referring now to Fig. 5F, in step 358 the VPN service provider software 30 waits for an event. When the VPN service provider software 30 senses that the SCP 32 has received on a signaling port a third-party billing setup message from the user terminal 26, then in step 360 the VPN service provider software 30 verifies the token just received from the user terminal 26, to determine, in step 362, if it is the same token that the VPN service provider software 30 sent to the user terminal 26 in step 344. If the token is not valid, then in step 364 the SCP 32 sends a release message to the terminal 26, and in step 366 condenses. If the token is valid, then in step 368 the SCP 32 sends a modified third-party billing setup message to the data network 14, and in step 370 condenses.

Although an illustrative embodiment of the invention has been shown and described, other modifications, changes, and substitutions are intended in the foregoing disclosure. Accordingly, it is appropriate that the appended claims be construed broadly and in a manner consistent with the scope of the invention.

WHAT IS CLAIMED IS:

1. A computerized method of a virtual private network service provider with third party billing, using a virtual private network to transfer data over a data network to a final destination, the method comprising the steps of:

   a. prompting the user at a data terminal to select a destination, password, and call type;
   b. selecting a virtual private network through the data network;
   c. giving an encryption key to the user, and then prompting the user for a password and a user identification;
   d. verifying the password, and providing an authorization code to the user; and
   e. allowing the user to transfer the data through the data network to the final destination, using the authorization code.

2. The method of claim 1, wherein step (d) further comprises negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

3. The method of claim 2, wherein step (c) further comprises encrypting the user's password, and sending the user identification and the encrypted password to the virtual private network service provider.

4. The method of claim 3, further comprising, after step (a), the step of sending a set-up message to the data network.

5. The method of claim 4, further comprising, after step (c), the step of the virtual private network service provider decrypting the encrypted password.

6. An apparatus for providing a datalink connection from a user terminal to a data network and to a virtual private network, with third party billing, comprising:

   a. an interface between the user terminal and the data network;

b. a switching control point device connected to the data network, the switching control point device connected to a computer; and

c. a computer-readable medium encoded with a method of using the virtual private network and the data network, with third party billing, the computer-readable medium accessible by the computer.

7. The apparatus of claim 6, wherein the method comprises negotiating for more bandwidth for the user, and including within an authorization code a grant of additional bandwidth.

8. The apparatus of claim 7, wherein the method further comprises encrypting a user's password, and temporarily storing the user identification and the encrypted password.

9. The apparatus of claim 8, wherein the method further comprises sending a set-up message to the data network.

10. The apparatus of claim 9, wherein the method further comprises decrypting the encrypted password.

11. A computer-readable medium encoded with a method of using a virtual private network, with third party billing, the method comprising the steps of:

a. prompting the user at a data terminal to select a destination, password, and call type;

b. selecting a virtual private network through the data network;

c. giving an encryption key to the user, and then prompting the user for a password and a user identification;

d. verifying the password, and providing an authorization code to the user; and

e. allowing the user to transfer the data through the data network to the final destination, using the authorization code.

12. The computer-readable medium of claim 11 wherein step (d) further comprises negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

13. The computer-readable medium of claim 12 wherein step (c) further comprises encrypting the user's password, and sending the user identification and the encrypted password to the virtual private network service provider.

14. The computer-readable medium of claim 13 further comprising, after step (a), the step of sending a set-up message to the data network.

15. The computer-readable medium of claim 14 further comprising, after step (c), the step of the virtual private network service provider decrypting the encrypted password.

16. An apparatus for providing a datalink connection from a user terminal to a data network and to a virtual private network, with third party billing, comprising:

a. means for prompting a user at the data terminal to select a destination, password, and call type;

b. means for selecting the virtual private network through the data network;

c. means for giving an encryption key to the user, and then prompting the user for a password and a user identification;

d. means for verifying the password, and providing an authorization code to the user; and

e. means for allowing the user to transfer data through the data network to a final destination, using the authorization code.

17. The apparatus of claim 16, further comprising means for negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

18. The apparatus of claim 17, further comprising means for encrypting the user's password, and sending the user identification and the encrypted password to the virtual private network service provider.

19. The apparatus of claim 18, further comprising means for sending a set-up message to the data network.

20. The apparatus of claim 19, further comprising means for decrypting the encrypted password.

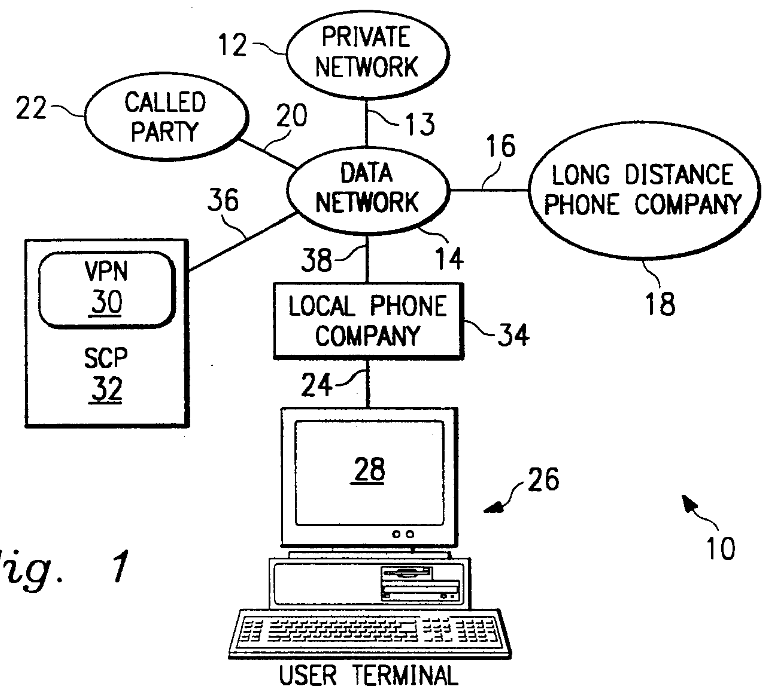

PRIVATE NETWORK
12
CALLED PARTY
22
20
13
DATA NETWORK
16
LONG DISTANCE PHONE COMPANY
36
38
14
18
VPN
30
SCP
32
LOCAL PHONE COMPANY
34
24
28
26
10
USER TERMINAL

Fig. 1

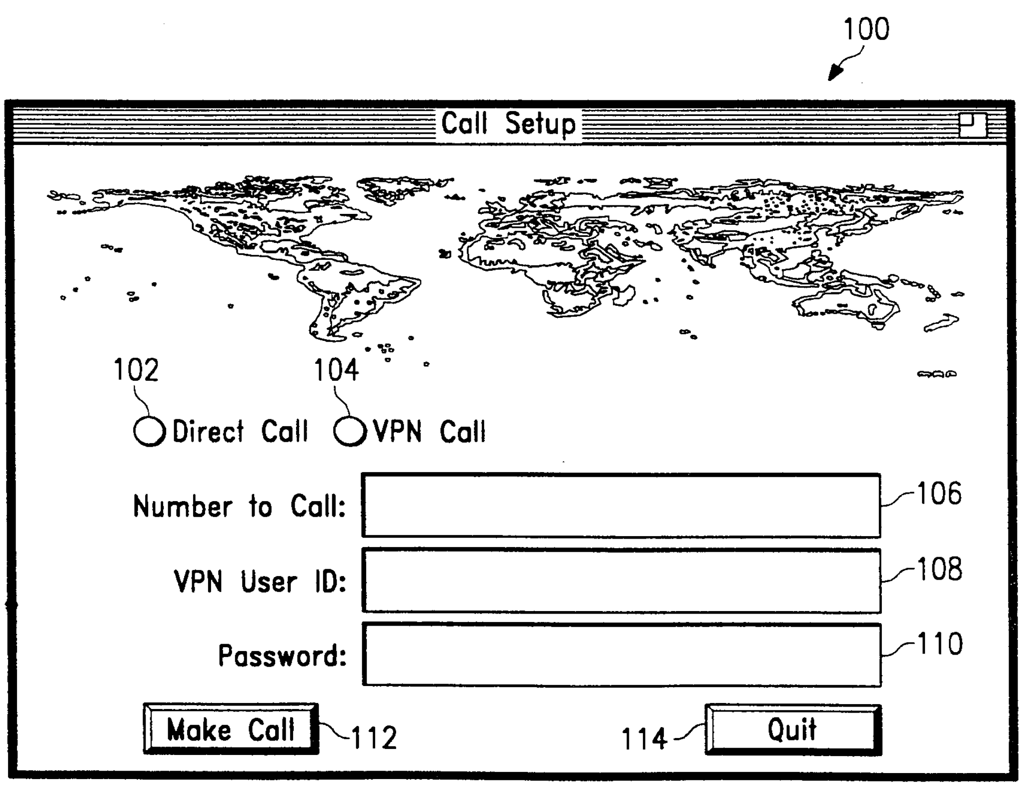

100
Call Setup
102
104
Direct Call
VPN Call
Number to Call:
106
VPN User ID:
108
Password:
110
Make Call
112
114
Quit

Fig. 3

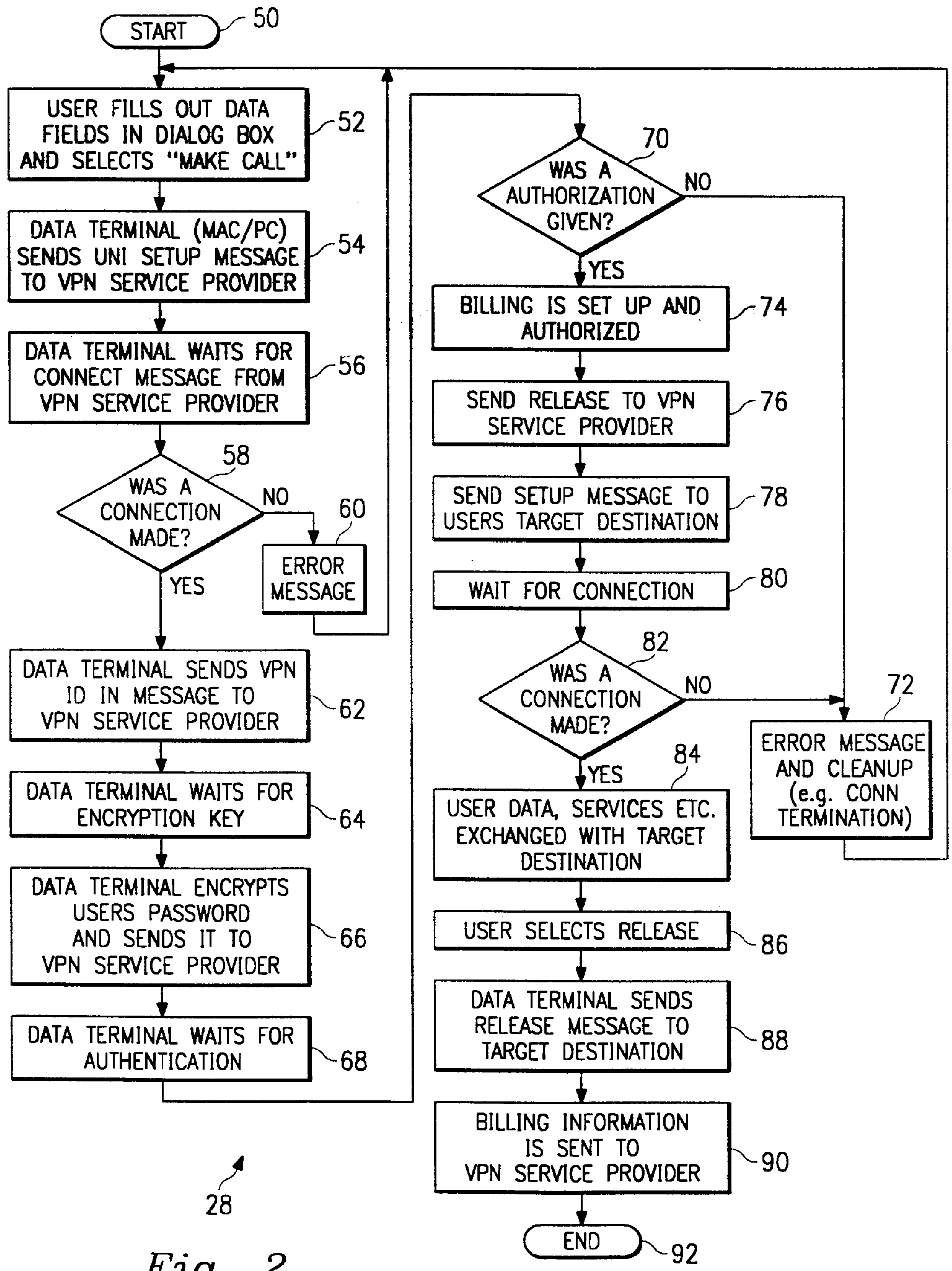
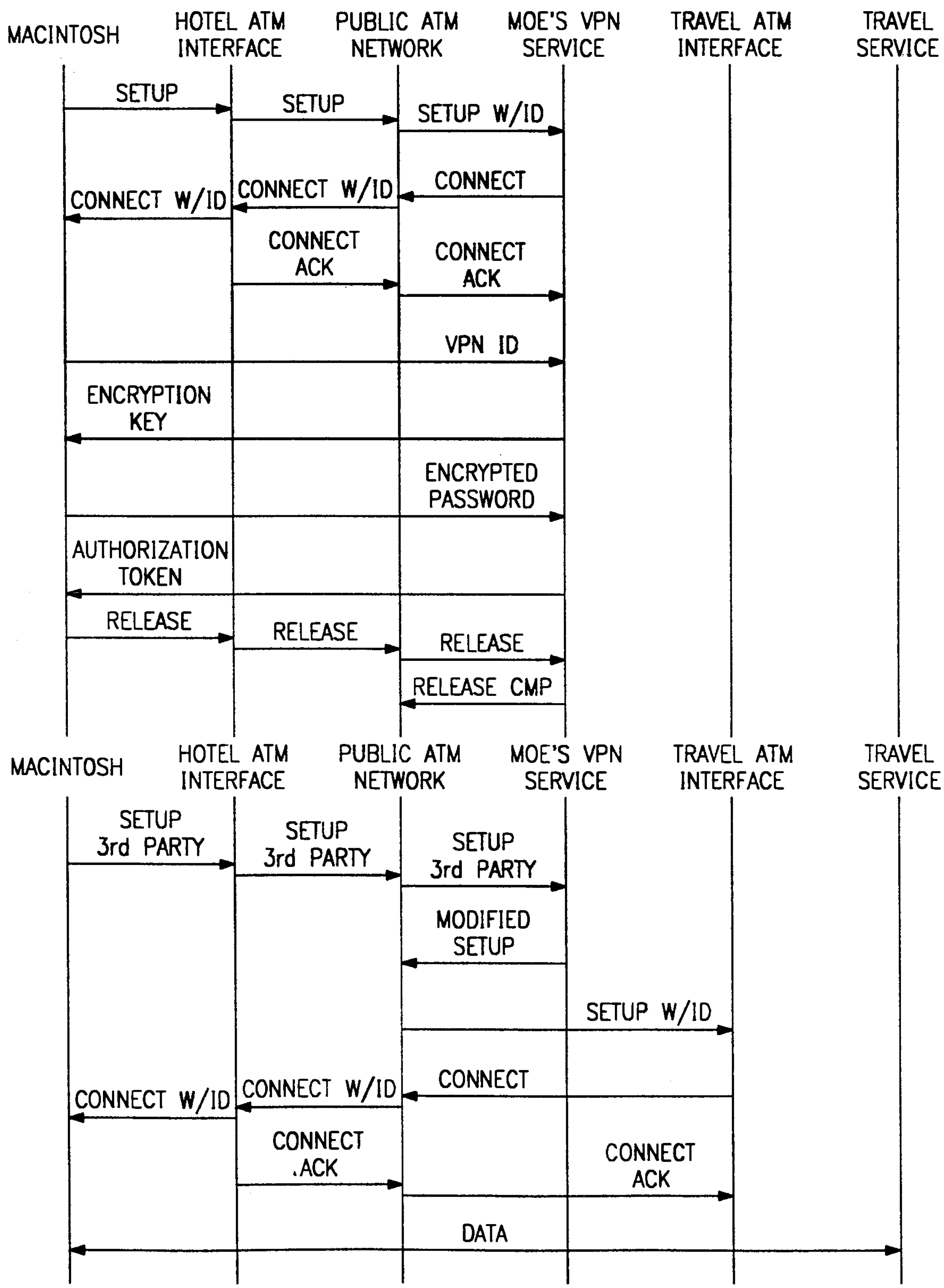

START
50
USER FILLS OUT DATA FIELDS IN DIALOG BOX AND SELECTS "MAKE CALL"
52
DATA TERMINAL (MAC/PC) SENDS UNI SETUP MESSAGE TO VPN SERVICE PROVIDER
54
DATA TERMINAL WAITS FOR CONNECT MESSAGE FROM VPN SERVICE PROVIDER
56
WAS A CONNECTION MADE?
58
NO
YES
ERROR MESSAGE
60
DATA TERMINAL SENDS VPN ID IN MESSAGE TO VPN SERVICE PROVIDER
62
DATA TERMINAL WAITS FOR ENCRYPTION KEY
64
DATA TERMINAL ENCRYPTS USERS PASSWORD AND SENDS IT TO VPN SERVICE PROVIDER
66
DATA TERMINAL WAITS FOR AUTHENTICATION
68
WAS A AUTHORIZATION GIVEN?
70
NO
YES
BILLING IS SET UP AND AUTHORIZED
74
SEND RELEASE TO VPN SERVICE PROVIDER
76
SEND SETUP MESSAGE TO USERS TARGET DESTINATION
78
WAIT FOR CONNECTION
80
WAS A CONNECTION MADE?
82
NO
YES
ERROR MESSAGE AND CLEANUP (e.g. CONN TERMINATION)
72
USER DATA, SERVICES ETC. EXCHANGED WITH TARGET DESTINATION
84
USER SELECTS RELEASE
86
DATA TERMINAL SENDS RELEASE MESSAGE TO TARGET DESTINATION
88
BILLING INFORMATION IS SENT TO VPN SERVICE PROVIDER
90
END
92
28

Fig. 2



MACINTOSH
HOTEL ATM INTERFACE
PUBLIC ATM NETWORK
MOE'S VPN SERVICE
TRAVEL ATM INTERFACE
TRAVEL SERVICE
SETUP
SETUP
SETUP W/ID
CONNECT W/ID
CONNECT W/ID
CONNECT
CONNECT ACK
CONNECT ACK
VPN ID
ENCRYPTION KEY
ENCRYPTED PASSWORD
AUTHORIZATION TOKEN
RELEASE
RELEASE
RELEASE
RELEASE CMP
MACINTOSH
HOTEL ATM INTERFACE
PUBLIC ATM NETWORK
MOE'S VPN SERVICE
TRAVEL ATM INTERFACE
TRAVEL SERVICE
SETUP 3rd PARTY
SETUP 3rd PARTY
SETUP 3rd PARTY
MODIFIED SETUP
SETUP W/ID
CONNECT W/ID
CONNECT W/ID
CONNECT
CONNECT .ACK
CONNECT ACK
DATA

Fig. 4A

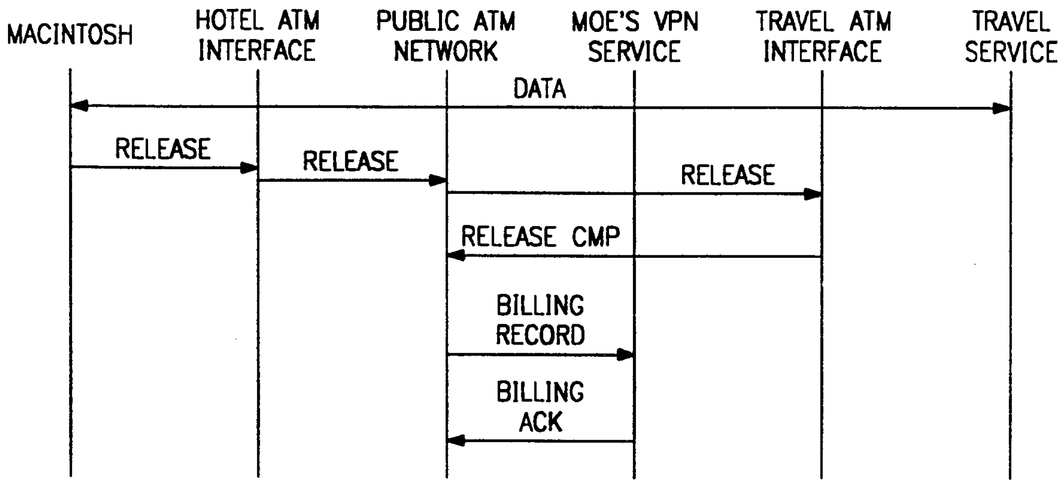


Fig. 4B

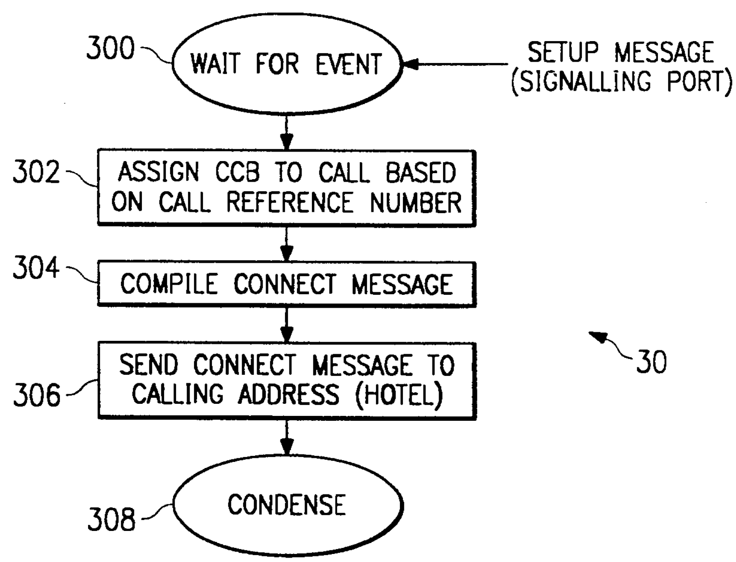


Fig. 5A

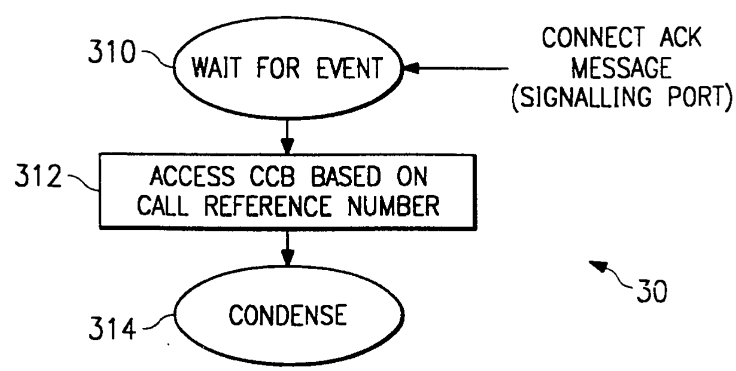


Fig. 5B

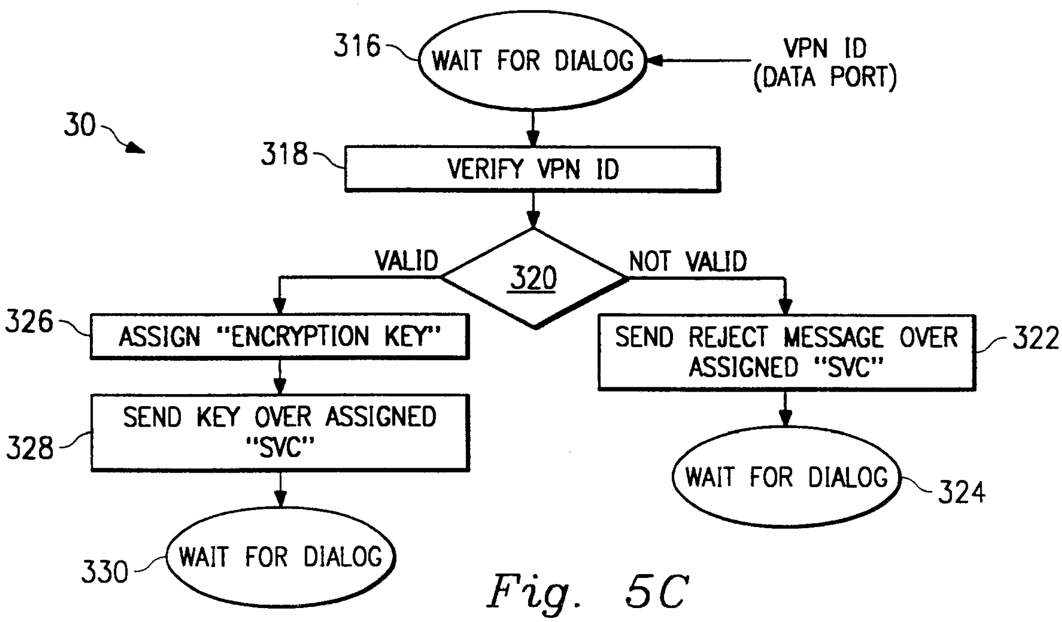

316
WAIT FOR DIALOG
VPN ID
(DATA PORT)
30
318
VERIFY VPN ID
VALID
320
NOT VALID
326
ASSIGN "ENCRYPTION KEY"
SEND REJECT MESSAGE OVER ASSIGNED "SVC"
322
328
SEND KEY OVER ASSIGNED "SVC"
WAIT FOR DIALOG
324
330
WAIT FOR DIALOG

Fig. 5C

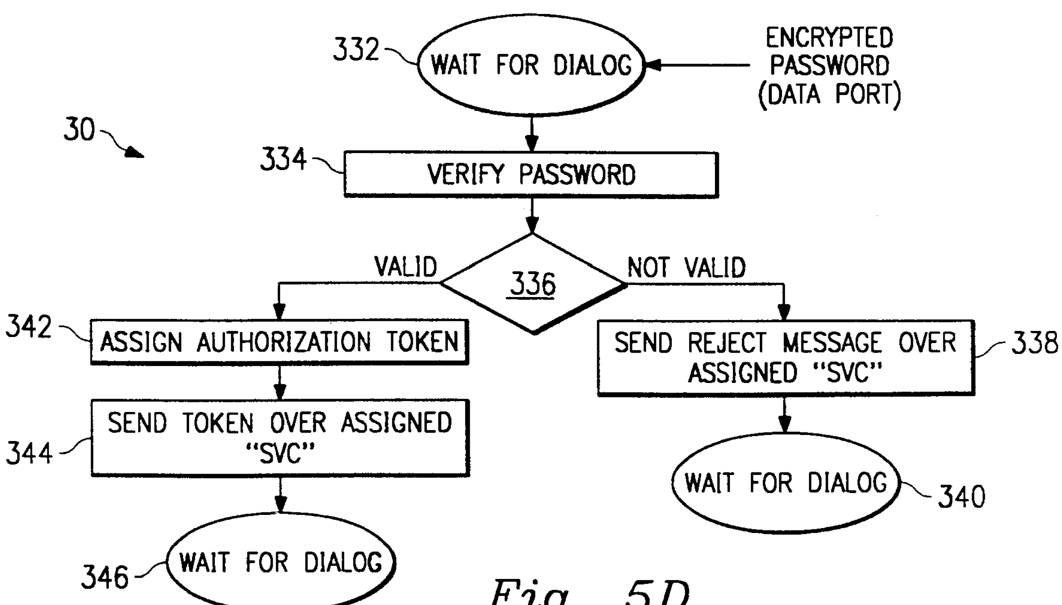

332
WAIT FOR DIALOG
ENCRYPTED
PASSWORD
(DATA PORT)
30
334
VERIFY PASSWORD
VALID
336
NOT VALID
342
ASSIGN AUTHORIZATION TOKEN
SEND REJECT MESSAGE OVER ASSIGNED "SVC"
338
344
SEND TOKEN OVER ASSIGNED "SVC"
WAIT FOR DIALOG
340
346
WAIT FOR DIALOG

Fig. 5D

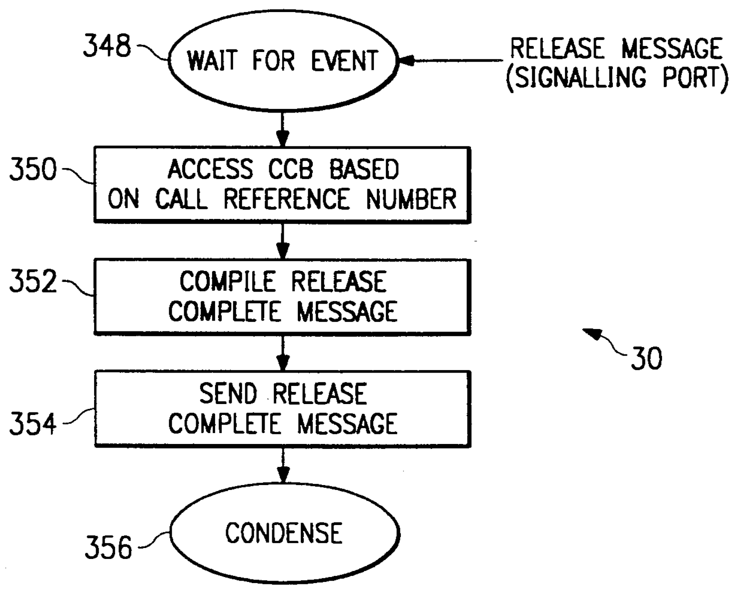

348
WAIT FOR EVENT
RELEASE MESSAGE (SIGNALLING PORT)
350
ACCESS CCB BASED ON CALL REFERENCE NUMBER
352
COMPILE RELEASE COMPLETE MESSAGE
354
SEND RELEASE COMPLETE MESSAGE
356
CONDENSE
30

Fig. 5E

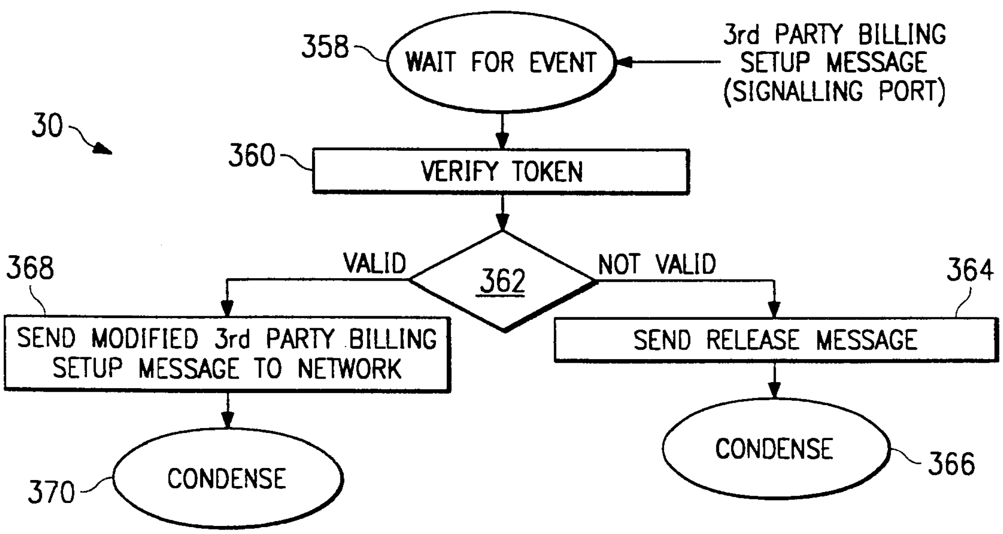

358
WAIT FOR EVENT
3rd PARTY BILLING SETUP MESSAGE (SIGNALLING PORT)
30
360
VERIFY TOKEN
362
VALID
NOT VALID
368
SEND MODIFIED 3rd PARTY BILLING SETUP MESSAGE TO NETWORK
364
SEND RELEASE MESSAGE
370
CONDENSE
CONDENSE
366

Fig. 5F

INTERNATIONAL SEARCH REPORT

Inte   ɔnal Application No
PCT/IB 97/01563

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04Q11/04 H04L12/22

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04Q H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	MUN CHOON CHAN ET AL: "AN ARCHITECTURE FOR BROADBAND VIRTUAL NETWORKS UNDER CUSTOMER CONTROL" NOMS '96 IEEE NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM , vol. 1, 15 April 1996, KYOTO, JP, pages 135-144, XP000641086 see abstract ---	1-20
A	BIC V: "VOICE PERIPHERALS IN THE INTELLIGENT NETWORK" TELECOMMUNICATIONS, vol. 28, no. 6, June 1994, page 29/30, 32, 34 XP000600293 see the whole document ---	1-20
	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

1

Date of the actual completion of the international search	Date of mailing of the international search report
19 March 1998	02/04/1998
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Authorized officer Staessen, B

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inte: ›nal Application No
PCT/IB 97/01563

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 729 256 A (NEDERLAND PTT) 28 August 1996 see abstract figures of pages 136 and 140 ---	1-20
A	CROCETTI P ET AL: "ATM VIRTUAL PRIVATE NETWORKS: ALTERNATIVES AND PERFORMANCES COMPARISONS" SUPERCOMM/ICC '94, 1 May 1994, NEW ORLEANS, US, pages 608-612, XP000438985 see abstract -----	1-20

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

Inte onal Application No
PCT/IB 97/01563

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0729256 A	28-08-96	NL 9500339 A	01-10-96

Form PCT/ISA/210 (patent family annex) (July 1992)

GB2334181

Publication Title:

Over-the-air re-programming of radio transceivers

Abstract:

Abstract of GB2334181

A method of downloading reprogramming data from a network for installation in a mobile station makes use of a dedicated small bandwidth pilot channel. The mobile station obtains from the base station the radio access parameters of a second channel. The second channel is a large bandwidth (bootstrap) channel suitable for fast transfer of data. The bootstrap channel is logically mapped onto a local transmission mode such as DECT or GSM by the mobile station and re-programming data may be downloaded from the base station via the bootstrap channel.

(12) UK Patent Application (19) GB (11) 2 334 181 (13) A

(43) Date of A Publication 11.08.1999

(21) Application No 9802545.5

(22) Date of Filing 06.02.1998

(71) Applicant(s)
NEC Technologies (UK) Ltd
(Incorporated in the United Kingdom)
Castle Farm Campus, Priorslee, TELFORD, Shropshire, TF2 9SA, United Kingdom

(72) Inventor(s)
Charles Marie Herve Noblet

(74) Agent and/or Address for Service
J W White
NEC Technologies (UK) Ltd, Level 3, The Imperium, Imperial Way, READING, Berks, RG2 0TD, United Kingdom

(51) INT CL[6]
H04Q 7/32

(52) UK CL (Edition Q)
H4L LDSC

(56) Documents Cited
US 5613204 A US 5109403 A

(58) Field of Search
UK CL (Edition P) H4L LDSC LDSU LECC LECX
INT CL[6] H04Q 7/32 7/38
Online: WPI

(54) Abstract Title

Over-the-air re-programming of radio transceivers

(57) A method of downloading reprogramming data from a network for installation in a mobile station makes use of a dedicated small bandwidth pilot channel. The mobile station obtains from the base station the radio access parameters of a second channel. The second channel is a large bandwidth (bootstrap) channel suitable for fast transfer of data. The bootstrap channel is logically mapped onto a local transmission mode such as DECT or GSM by the mobile station and re-programming data may be downloaded from the base station via the bootstrap channel.

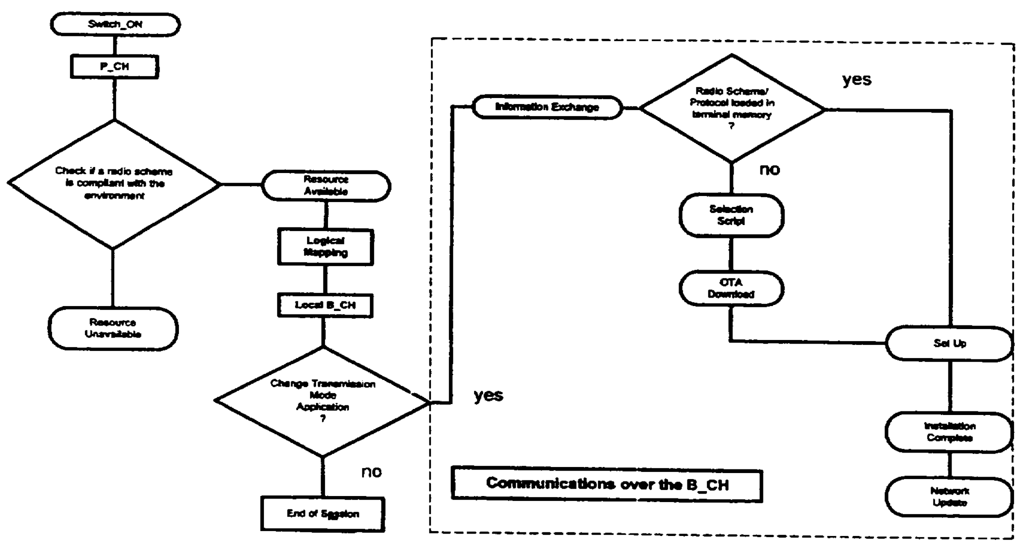


Figure 2

GB 2 334 181 A

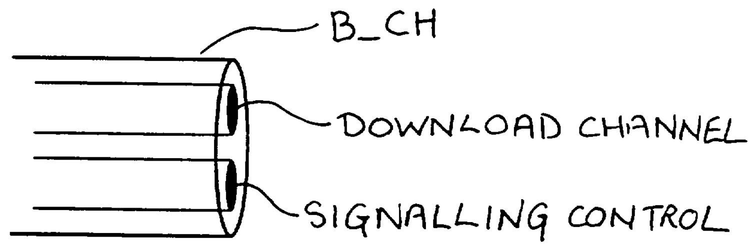
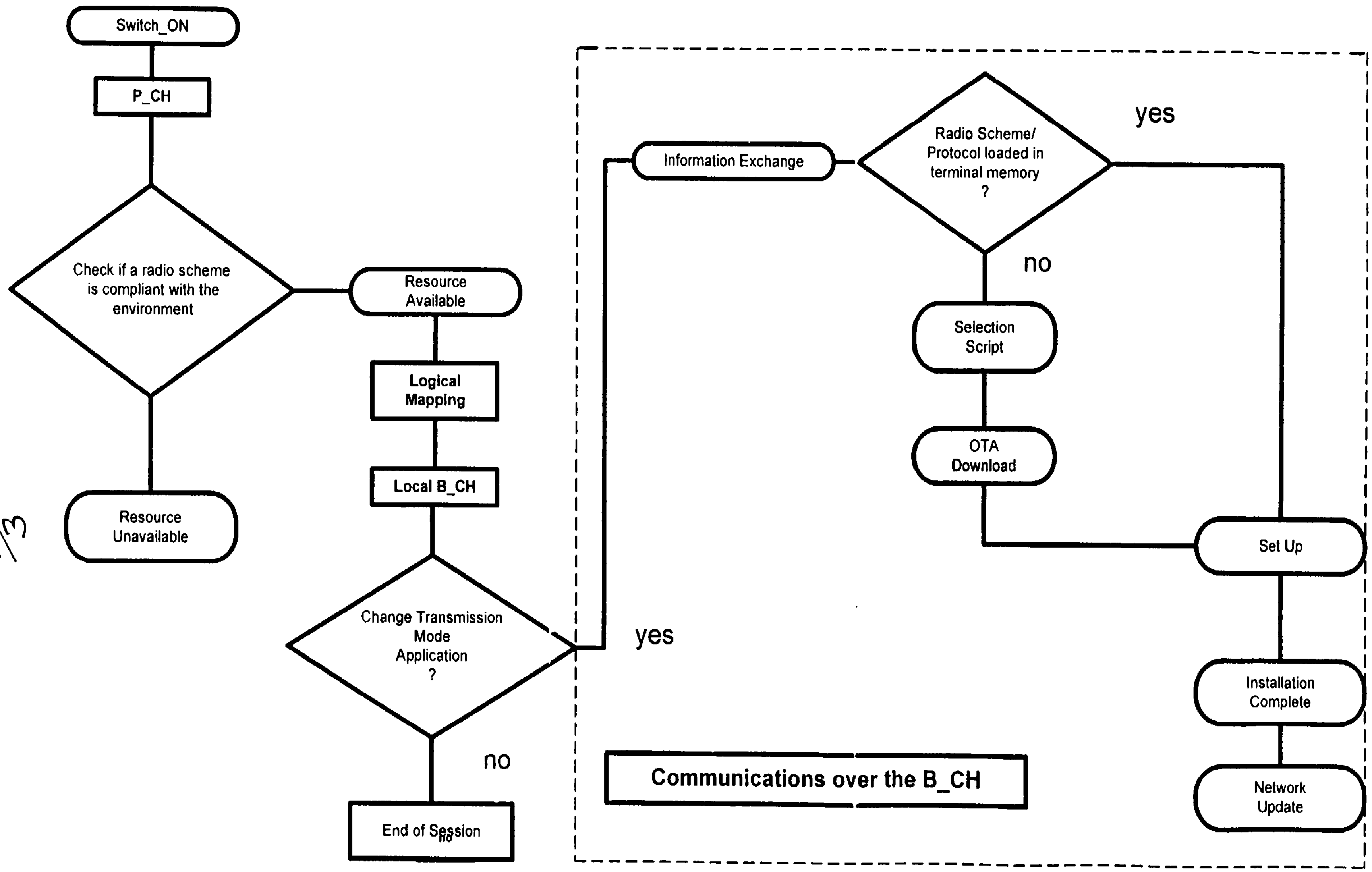


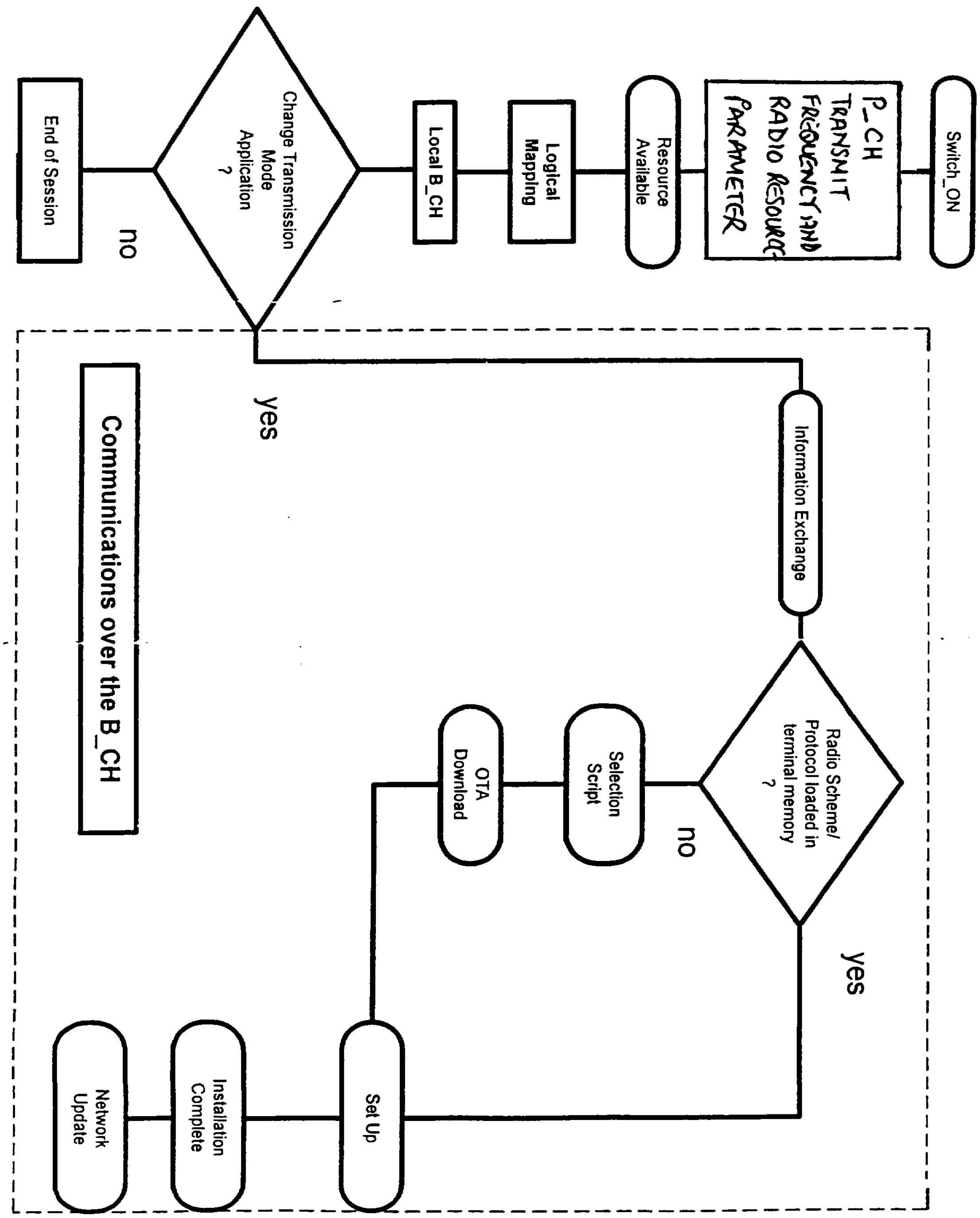
Figure: 1

2/3



Switch_ON
P_CH
Check if a radio scheme is compliant with the environment
Resource Unavailable
Resource Available
Logical Mapping
Local B_CH
Change Transmission Mode Application ?
no
End of Session
yes
Information Exchange
Radio Scheme/ Protocol loaded in terminal memory ?
yes
no
Selection Script
OTA Download
Set Up
Installation Complete
Network Update
Communications over the B_CH

Figure 2



Switch_ON
P_CH TRANSMIT FREQUENCY AND RADIO RESOURCE PARAMETER
Resource Available
Logical Mapping
Local B_CH
Change Transmission Mode Application ?
no
End of Session
yes
Information Exchange
Radio Scheme/ Protocol loaded in terminal memory ?
no
Selection Script
OTA Download
yes
Set Up
Installation Complete
Network Update
Communications over the B_CH

Figure 3

2334181

Over-the-air re-programming of radio transceivers

This invention relates to radio transmitter/receivers and in particular it relates to a method of re-programming radio transmitter/receivers over-the - air.

A radio transmitter/receiver (transceiver) such as a radiotelephone is designed for operation with particular types of networks such as GSM 900 or DCS 1800. Intended use of the radiotelephone with a particular network(s) in a restricted geographical area, however, requires that the telephone be configured so as properly to communicate with the particular network (s). The user of a radiotelephone will usually have a telephone which has been configured for communication with a so called "home network". The home network is the local network usually most used by the subscriber.

The area within which a user of e.g. a GSM radiotelephone may operate, however, is considerable and is not limited to the home network but may be used on many other networks throughout the world. Use of a handset outside the home network is known as "roaming".

When the radiotelephone is to be used in roaming it is often necessary for it to have a configuration different to that for use with the home network. It is possible for re-configuration of radio transmitter/receivers to be effected by means of signals received across the air interface.

It is also convenient for the radio to be re-configurable over the air interface so as to support different types of communication and user applications e.g. addition of address book manager, whether or not it is located in the home network.

Over the air re-programming of radio receivers is well known in the art and reference may be made to US patent 5 381 138 for example. The capability to obtain programming data from a network is particularly useful for a roaming radio transmitter/receiver.

When beginning operation in an area for which the radiotelephone is not configured and it is required to download the data for reconfiguration from one of the available networks, a communication link must first be established with the network of interest. It has been proposed that a pilot channel be established in all areas from which the roaming radiotelephone may obtain the data necessary for reconfiguration.

A pilot channel of this type, however, will require a relatively large bandwidth to allow a sufficiently fast transfer of the data required.

According to the invention there is provided a method of downloading reprogramming data from a network for installation in a radio transmitter/receiver comprising initial communication from a first dedicated channel of relatively small bandwidth broadcasting at least the frequency and radio access parameters of a second channel of relatively large bandwidth from which reprogramming data may be downloaded.

Examples of the invention will now be described in more detail with reference to the accompanying figures in which

figure 1 Illustrates the logical structure of the bootstrap channel

figure 2 Is a flow diagram of a reconfiguration process

figure 3 Is a flow diagram of an alternative reconfiguration process

A roaming radio transmitter/receiver (mobile) is located in a region served by one or more networks and the user wishes to communicate with a network from which he can obtain reprogramming data and subsequently begin communicating with the network in the communication mode selected.

A pilot channel broadcast is maintained in the region and contained in the pilot channel broadcast there is at least sufficient information for the mobile to connect to a second channel which we shall call the bootstrap channel. Conveniently the pilot channel will be broadcast in all regions over a standardised radio interface. Only a small bandwidth is required for the pilot channel because of the small amount of information contained in the broadcast.

The small bandwidth requirement makes the task of standardisation much easier with respect to the pilot channel. The wider bandwidth channels are more conveniently assigned locally for ease of implementation.

The Pilot Channel (P_CH) broadcasts a list of sets of parameters corresponding to networks available in the region. The mobile receives the network transmission through the P_CH. If the existing configuration of the mobile is matched to the available regional radio schemes, then a second channel the bootstrap channel (B_CH) is logically mapped onto the selected transmission mode. The base station and mobile exchange information over this dedicated logical channel.

The Bootstrap channel is logically mapped on top of one of the default modes of the terminal; a mapping of a logical B_CH onto the physical GSM channel for instance may be implemented. Once the mapping has been effected the terminal may download data from the base station. The bootstrap channels provided by each operator may accommodate differing services with regard to the applications available for downloading.

The flow diagram shown at fig 3 depicts a reconfiguration procedure.

When the mobile is switched on, it reads the Pilot Channel broadcast. The mobile must be configured to support the (standardised) radio interface of the Pilot Channel. The Pilot Channel carries local radio parameters (standards supported in the regional environment in which the mobile is located). After processing the received information, the mobile communicates with the base station through the Bootstrap Channel, provided that the mobile has the minimum resources required by its local radio environment. Prior to the change of channel, P_CH to B_CH, a logical mapping of the Bootstrap Channel is performed within the mobile on the selected air interface.

When operation on a local B_CH transmission has been established, the user may wish to change some properties or the performance of his mobile and can request supply of the desired services from the network. If no changes are required then the mobile adopts the default transmission mode in stand-by and releases the allocated B_CH.

If the user requests a change then communication between the base station and mobile is maintained for the exchange, the nature of which will depend on the capabilities of both mobile and network. At least 3 conditions can affect the nature of this information exchange.

Firstly, the mobile may not be able to support the required software. Where the mobile is not able to support the required software, no communication channel is available to the mobile from the existing network resources and use of the mobile within the region will therefore not be possible.

Secondly, the required software may be stored already in the mobile's memory. In this situation there is no need to download a software module but the allocated B_CH connection is maintained for further operations as described.

Thirdly, the software module required to support a different type of communication or user application may need to be downloaded from the base station. Where the download of a software module is required, initially a selection script is downloaded to the mobile followed by downloading and installation of the required software.

When the installation of the required software into the mobile has been completed, the mobile signals to the network the achievement of correct reconfiguration. On receipt of the "correct reconfiguration" signal from the mobile details of the mobile identity and its present configuration are entered on the network database (to license the product for instance) .

With reference to figure 1, the logical structure of the bootstrap channel will include 2 logical sub-channels : a download channel and a signalling control channel (S_CH). The signalling control channel assists in the reduction of errors in transmission so as to allow correct software download.

In the above example, the first channel, the Pilot Channel, is standardised and the mobile must be configured to support the radio interface for the Pilot Channel. The second (bootstrap) channel may be subject to local definition through logical mapping on a local transmission mode e.g. GSM, DECT and the mobile is not initially configured to support the radio interface for the bootstrap channel..

An example of a method of reprogramming providing greater flexibility will now be given. In this example the mobile is configured to support the radio interfaces for both the first, dedicated relatively small bandwidth (Pilot) channel and the second relatively large bandwidth (bootstrap) channel. That is to say that when the mobile is switched on in most and preferably all regions, the network can communicate with the mobile via both pilot and bootstrap channels.

In order for the mobile always to have the appropriate radio interface for the bootstrap channel then this channel would need also to be standardised (in addition to the Pilot Channel). The parameters of the bootstrap channels provided in different regions may have local variations in terms of e.g. allocated frequency, data rate and available user applications.

With reference to figure 3 which is a flow diagram of the reconfiguration process for this example, the mobile when switched on reads the Pilot Channel broadcast. The allocated frequency and radio resource parameters for the bootstrap channel contained in the pilot channel broadcast are processed and any required logical mapping effected. After processing the received information, the mobile communicates with the base station through the Bootstrap Channel.

The condition likely to be experienced in the previous example whereby the mobile is not able to support the required software and no communication channel is available to the mobile from the existing network resources does not apply in this arrangement. The communication via the bootstrap channel allows the request for and supply of the software module necessary to establish communication with the network. The transfer to the bootstrap channel does not depend on the existing configuration of the mobile since the bootstrap channel is standardised in this example and the mobile is equipped to interface, via the pilot channel, with the bootstrap channel.

The services and structure offered by the Bootstrap Channel are common for both of the above examples, however, the requirements on the terminals and networks differ.

The bootstrap channel will provide the following services by means of over -the-air (OTA) reconfiguration :
capability Exchange - the terminal provides some information to the network on its current configuration and capabilities.
module Selection : at this stage the user specifies the software that his terminal requires to download. This operation could be compared to an installation script.
data download : transfer of the data. In some cases software code will have to be downloaded whilst in other cases the software may already be implemented in the mobile. In the latter case, a set-up mechanism would be sufficient to initiate the reconfiguration.

Once the mobile and the base station are synchronised on the bootstrap channel, information exchange can begin.

Claims

1. A method of downloading reprogramming data from a network for installation in a radio transmitter/receiver comprising initial communication from a first dedicated channel of relatively small bandwidth broadcasting at least the frequency and radio access parameters of a second channel of relatively large bandwidth from which reprogramming data may be downloaded.

2. A method of downloading reprogramming data from a network as in claim 1 where first, dedicated relatively small bandwidth channel has a standard radio interface common to many network locations.

3. A method of downloading reprogramming data from a network as in claim 2 where second relatively large bandwidth channel has a standard radio interface common to many network locations.

4. A method of downloading reprogramming data from a network as in claims 1 to 3 where first, dedicated relatively small bandwidth channel broadcasts a list of sets of parameters corresponding to networks available in the region.

5. A method of downloading reprogramming data from a network as in claim 1 where the radio transmitter/receiver is configured to support the radio interfaces for both the first, dedicated relatively small bandwidth channel and the second relatively large bandwidth channel.

The Patent Office

10

Application No: GB 9802545.5
Claims searched: 1 to 5

Examiner: Glyn Hughes
Date of search: 17 August 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): H4L (LDSC, LDSU, LECC, LECX)

Int Cl (Ed.6): H04Q 7/32, 7/38

Other: Online: WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	US 5613204	(HABERMAN ET AL) see in particular column 15 lines 48 to 50	1
X	US 5109403	(SUTPHIN) see abstract	1

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

An Executive Agency of the Department of Trade and Industry

EP0814589

Publication Title:

System and method for automated network reconfiguration

Abstract:

Abstract of EP0814589

A method is disclosed for providing an enhanced level of security for sensitive or proprietary information associated with information transactions in a public network, such as the Internet. In carrying out that method, an on-line information transaction is bifurcated between a generalized information access portion of such a transaction and an exchange of sensitive user information. With such a bifurcation, the generalized information access portion of the transaction, which generally would constitute the more substantial (in terms of network resources) portion of the transaction, would be handled via a non-secure network, usually a public network such as the Internet. The portion of the transaction involving sensitive user information, on the other hand, would be handled by a separate secure connection, such as a private network, or in 10a7 tranetwork. An important characteristic of this bifurcation arrangement is the provision of a means for automated reconfiguration of a user terminal as between accessing the Ageneralized information via the non-secure network and access to the secure communications network for the exchange of sensitive user information. Such an automated reconfiguration will be carried out without the necessity for any action on the part of the user, and indeed will be largely invisible to the user.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 814 589 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
29.12.1997 Bulletin 1997/52

(51) Int. Cl.^{6}: H04L 29/06

(21) Application number: 97109792.8

(22) Date of filing: 16.06.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: 19.06.1996 US 667524

(71) Applicant: AT&T Corp.
New York, NY 10013-2412 (US)

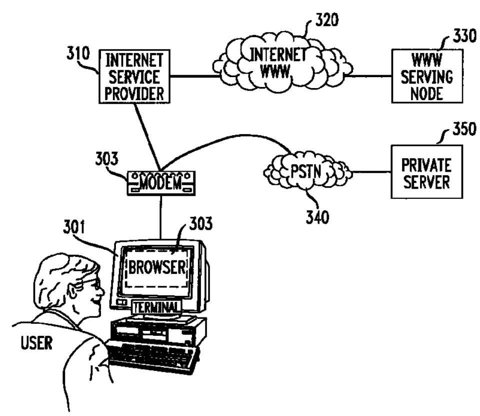
(72) Inventors:
- Harwood, Jonathan P.
  Morganville, N.J. 07751 (US)
- Kimmeth, Thomas
  Gladstone, N.J. 07977 (US)
- Nusbaum, Kurt
  Downers Grove, Illinois 60515 (US)

(74) Representative:
KUHNEN, WACKER & PARTNER
Alois-Steinecker-Strasse 22
85354 Freising (DE)

(54) System and method for automated network reconfiguration

(57) A method is disclosed for providing an enhanced level of security for sensitive or proprietary information associated with information transactions in a public network, such as the Internet. In carrying out that method, an on-line information transaction is bifurcated between a generalized information access portion of such a transaction and an exchange of sensitive user information. With such a bifurcation, the generalized information access portion of the transaction, which generally would constitute the more substantial (in terms of network resources) portion of the transaction, would be handled via a non-secure network, usually a public network such as the Internet. The portion of the transaction involving sensitive user information, on the other hand, would be handled by a separate secure connection, such as a private network, or intranetwork. An important characteristic of this bifurcation arrangement is the provision of a means for automated reconfiguration of a user terminal as between accessing the Ageneralized information via the non-secure network and access to the secure communications network for the exchange of sensitive user information. Such an automated reconfiguration will be carried out without the necessity for any action on the part of the user, and indeed will be largely invisible to the user.

FIG. 3



EP 0 814 589 A2

Printed by Xerox (UK) Business Services
2.15.6/3.4

Description

FIELD OF THE INVENTION

This invention is related to the field of data communications, and more particularly to a method and means for establishing an automatic reconfiguration of a user terminal among alternative tasks.

BACKGROUND OF THE INVENTION

With the increasing popularity of personal computers over the last several years has come a striking growth in transaction-oriented computer-to-computer communications (as opposed to bulk-data transfers among such computers). For convenience herein such transaction-oriented computer-to-computer communications will be described by the shorthand term "information transaction". That growth in the use of computers for such information transactions has unquestionably been fueled by the existence of an international infrastructure for implementing such data communications, known as the Internet. And, driven by the burgeoning demand for such information transaction services, the Internet has itself experienced explosive growth in the amount of traffic handled.

At least partly in response to that demand, a new level of accessibility to various information sources has recently been introduced to the Internet, known as the World Wide Web ("WWW"). The WWW allows a user to access a universe of information which combines text, audio, graphics and animation within a hypermedia document. Links are contained within a WWW document which allow simple and rapid access to related documents. Using a system known as the HyperText Markup Language ("HTML"), pages of information in the WWW contain pointers to other pages, those pointers typically being a key word (commonly known as a hyperlink word). When a user selects one of those key words, a hyperlink is created to another information layer (which may be in the same, or a different information server), where typically additional detail related to that key word will be found.

In order to facilitate implementation of the WWW on the Internet, new software tools have been developed for user terminals, usually known as Web Browsers, which provide a user with a graphical user interface means for accessing information on the Web, and navigating among information layers therein. A commonly used such Web Browser is that provided by Netscape.

The substantial growth in the use of computer networks, and particularly the WWW, for such information transactions, has predictably led to significant commercialization of this communications medium. For example, with the WWW, a user is not only able to access numerous information sources, some public and some commercial, but is also able to access "catalogs" of merchandise, where individual items from such a catalog can be identified and ordered, and is able to carry out a number of banking and other financial transactions. As will be obvious, such commercial transactions will typically involve sensitive and proprietary information, such as credit card numbers and financial information of a user. Thus, with the growth of commercial activity in the Internet, has also come a heightened concern with security.

It is well known that there are persons with a high level of skill in the computer arts, commonly known as "hackers", who have both the ability and the will to intercept communications via the Internet. Such persons are thereby able to gain unauthorized access to various sensitive user information, potentially compromising or misappropriating such information.

The vulnerability of such sensitive user information to misuse when so transmitted via the Internet is a phenomena which has only recently received wide public attention. Unless such security concerns can be quickly addressed and alleviated, the commercial development of this new communications medium may be slowed or even stalled altogether.

SUMMARY OF THE INVENTION

Accordingly, it is an object of the invention to provide an acceptable level of security for sensitive or proprietary information associated with information transactions in a public network, such as the Internet. That object is realized through an arrangement whereby an on-line information transaction is bifurcated between a generalized information access portion of such a transaction and an exchange of sensitive user information. With such a bifurcation, the generalized information access portion of the transaction, which generally would constitute the more substantial (in terms of network resources) portion of the transaction would be handled via a non-secure network, usually a public network such as the Internet. The portion of the transaction involving sensitive user information, on the other hand, would be handled by a separate secure connection, such as a private network, or intranetwork. An important characteristic of this bifurcation arrangement is the provision of a means for automated reconfiguration of a user terminal as between accessing the generalized information via the non-secure network and access to the secure communications network for the exchange of sensitive user information. Such an automated reconfiguration will be carried out without the necessity for any action on the part of the user, and indeed will be largely invisible to the user. In a further embodiment of the invention, a transfer of data is provided from a public to a private network, wherein data selected by a user from a public network site may be arranged and displayed at a user terminal and, subject to further user selection/confirmation activity, thereafter transferred to a private network.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 depicts an illustrative case of information transactions carried out via a public network such as the Internet.

Figure 2 shows the architecture of a browser as would typically be applied for accessing a hypermedia web page.

Figure 3 illustrates the primary elements of the reconfigurable dual-path method of the invention.

Figure 4 depicts in flow chart form the basic jump capability of the methodology of the invention.

Figures 5A & 5B (generally designated collectively herein as "Figure 5") depict in flow chart form the "shopping cart" capability of the methodology of the invention.

Figure 6A & 6B (generally designated collectively herein as "Figure 6") depict in flow chart form the stored configuration capability of the methodology of the invention.

Figure 7A & 7B (generally designated collectively herein as "Figure 7") depict in flow chart form the off-line form capability of the methodology of the invention.

DETAILED DESCRIPTION

For clarity of explanation, the illustrative embodiment of the present invention is presented as comprising individual functional blocks. The functions these blocks represent may be provided through the use of either shared or dedicated hardware, including, but not limited to, hardware capable of executing software.

Figure 1 depicts an illustrative case of information transactions carried out via the Internet. As seen in the figure, an exemplary user obtains access to the Internet by First connecting, via a Terminal **110** having an associated Browser **111**, to an Internet Service Provider **112** selected by the user. That connection between the user and the Internet Service Provider will typically be made via the Public Switched Telephone Network (PSTN) from a modem associated with the user's Terminal to a network node in the Internet maintained by the selected Internet Service Provider.

Once the user has obtained access to the selected Internet Service Provider, an address is provided for connection to another user or other termination site and such a connection is made via the Internet to that destination location. As can be seen from the figure, communication via the Internet may be either user-to-user, as from Terminal **110** to Terminal **130**, or from a user to a node representing an information source accessed via the Internet, such as Public Site **120**.

It will of course be understood that the Internet provides service to a large number of users and includes a large number of such Public Sites, but the illustration provides the essential idea of the communication paths established for such Internet communication. It will also be understood that a number of service classifications are supported by the Internet, with the World Wide Web service, which represents a preferred embodiment for the public network aspect of the method of the invention, being one of the currently most heavily trafficked of such services.

The Web Browser, such as depicted at **111**, can be seen as a software application operating in conjunction with a user terminal (such as Terminal **110**) which provides an interface between such a user terminal and the particular functionality of the WWW information site. The architecture of such a browser is generally described in terms of three main components, as illustrated in Figure 2. At the top level is the Browser **201**, which enables the acquisition of information pages from a WWW server (beginning, in all cases, with the "home page" for that server), for display at a display device associated with the terminal. The Browser also provides the necessary interface for the terminal with the HTML functionality used by the server to provide access to other linked information layers.

The second level of the browser architecture is the TCP/IP Stack **202**, which handles the communications protocols used for connecting the terminal to the WWW server. The bottom level of this architecture is the Dialer **203**, which typically handles the function of providing dialing and setup digits to a modem, as illustrated at **204**, such a modem generally being a part of the terminal. Normally, upon receiving dialing and other setup information from the dialer, the modem would cause a connection to be made via the PSTN to the Internet Service Provider selected for that terminal.

After a connection is established in this manner to the Internet Service Provider, an address would be provided for the WWW information node sought to be contacted, a connection to that node made through the Internet, and the home page for that node caused to be displayed at the terminal's display device. A user would then select a key word in that home page, typically by clicking on the word with a mouse or similar device, and, upon transmission of that selection signal to the WWW server, a hyperlink would be created to the linked information layer and the open page of that layer would be caused to be displayed at the user terminal.

As explained above, serious questions have been raised in respect to the security of communications via the public Internet. (Note, that the discussion herein is focused on the Internet, and particularly the WWW functionality of the Internet, as a preferred embodiment of such public data communication networks generally, but the methodology of the invention will be applicable to any such network.) To address this problem, the methodology of the invention begins with a bifurcation of the information transaction between a user and the selected information transaction provider into a portion related to sensitive or proprietary user information, and other information comprising that transaction. With such a bifurcation, it becomes possible to provide substantial security for that proprietary information by use of an alternative communications path for that separated portion of the transaction via a private network, or intranetwork -- *i.e.*, a connection between a user's terminal and a secure serving node on that private network. It is anticipated that a coordination means will be established in respect to the management of information among the public and private network elements of the bifurcated information transaction.

In its basic form, this methodology may be carried out by the user terminal initiating a call via the Internet to a selected WWW node, and upon establishing connection to that node, proceeding with the desired information transaction up to the point where an exchange of sensitive or proprietary information were required. At that point the user terminal would be instructed by the WWW server to terminate that connection (*i.e.*, hangup) and to place a new call to an identified private network server for the necessary exchange of sensitive information.

However, in order to accomplish such a dual-path transaction, it is necessary that the browser at the user terminal be reconfigured to provide the dialing, authorization (*i.e.*, login and password), and other needed information for accessing the alternative private network, in order to implement the proprietary portion of the transaction. It will also usually be the case that, upon completion of that private-network transaction, the original dialer, stack and browser configurations will need to be restored, in order for the terminal to retain its normal Internet access functionality. Such a reconfiguration and subsequent restoral of the necessary parameters in the browser, stack and dialer is likely to be well beyond the capabilities of the average user.

Accordingly, as a further embodiment of the inventive methodology, an automated browser reconfiguration means is provided which interoperates with the browser. This browser reconfiguration means is described in detail hereafter and will be referred to as the "Bridging Software".

Figure 3 provides an illustration of the primary elements of the reconfigurable dual-path method of the invention. As seen in the figure, a first path comparable to the Internet link shown in Figure 1, between User Terminal **301** and WWW Serving Node **330** (via Browser **302**, Modem **303**, Internet Service Provider **310**, and Internet **320**) is provided. However, an alternative path is now provided from the output of Modem **303** to Private Server **350**. That path is illustrated as being via the PSTN, which is generally regarded as being highly secure, but an alternative dedicated or other more-secure path between the User Terminal **301** and the Private Server **350** could as well be provided. In keeping with the discussion above, Browser **302** shown in Figure 3 would also include the Bridging Software installed as a helper application for implementing the automatic reconfiguration of the Browser.

In the operation of this system, a user would normally make an initial connection to an Internet application, such as the application represented by WWW Serving Node **330**, which, *e.g.*, might be a shopping application, a financial transaction, or the provision of an enrollment form for off-line preparation. After conducting all, or some portion of an information transaction short of an exchange of sensitive or proprietary information, including a capture by the user's terminal of needed information from the public site, a user provides a signal indicative of an end to that portion of that transaction. During the course of the public portion of the information transaction, specially configured files are sent from the WWW serving node to the Bridging Software associated with Browser **302**. Such files contain instructions for the Bridging Software to store information-like products -- e.g., for selected items from a catalog, forms for enrollment, or non-secure portions of a financial transaction, and reconfiguration information for dialing and logging into the private portion of the transaction. The Bridging Software then hangs up the Internet connection, edits the user terminal's browser, stack and dialer files to reconfigure the terminal to connect to the private server. Prior to automatic redialing of the new private site for the user, the Bridging Software may be instructed by the application operating at WWW Server Node **330** to display items chosen for purchase, or to display a form for the end-user to complete off-line before dialing the private application. Upon connecting to the private application and completing the transaction as to the user sensitive information in a private environment, the Bridging Software then restores the end-user software to the dialing and authorization parameters required to dial to the public Internet.

A particularly advantageous application of the automated reconfiguration and information transfer methodology of the Bridging Software is that it adds value to certain WWW servers which do not possess the Common Gateway Interface ("CGI") capability -- *i.e.*, a provision of specialized functions on the server beyond just displaying HTML files, and are accordingly unable to accomplish any transactional processing in respect to items selected by a user. In effect, such a non-CGI server, on its own, can only serve as a "billboard" for the items represented in its database.

However, with the collection and redelivery process of the Bridging Software, a data capture and processing mechanism can be implemented for servers operating in a non-CGI environment -- such servers being incapable of more than the simple delivery of static data packets corresponding to available items. The data set enabled by the Bridging Software is a mechanism for augmenting such limited server capabilities by defining a flexible mechanism for the receipt, display, and delivery of arbitrary data from one site to another.

In such a scenario, the Bridging Software receives a "shopping cart" item list from the host as a data-set defined with a static MIME data packet associated with the Bridging Software. This information comprising the data-set may be updated, displayed to the user in a "read-only" fashion, or presented to the user for order selection.

During the process of interacting with the WWW server, a user may trigger HTML links resulting in additional MIME packets for the Bridging Software being delivered to the client. These packets allow items to be added and/or removed from the specified data set or presented to the user for local confirmation. The user will interact with a pop-up screen provided by the Bridging Software which presents the items available with product information, such as part number, description, unit cost, etc. The user identifies those items which are to be placed into the "shopping cart" and the quantity of items desired. Upon completion of the form, the Bridging Software stores the order in a format suitable for subsequent delivery to the private server site.

An additional feature provided by the methodology of the Bridging Software is an automated mechanism for providing compatibility with user terminals not previously having the Bridging Software included with the terminal's browser. To that end, the Bridging Software located at an accessed public network site initially checks to see if the browser counterpart for that software is loaded at the calling user terminal. If yes, the heretofore described processes of the Bridging Software go forward. If not however, a request is sent through the public host to download the Bridging Software to the calling terminal. After such a download, a helper application loads the Bridging Software to the terminal's browser.

I. Illustrative Embodiments

A variety of browser reconfiguration applications are supported by the automated browser reconfiguration means of the invention. Four essentially diverse capabilities of this invention, which support such applications, are described hereafter as illustrative embodiments of the invention.

A. Basic Jump Capabilities

In this configuration, which is illustrated in flow chart form in Figure 4, an end-user is connected to a chosen WWW serving node (where a desired information product is made available) via a modem and an Internet browser associated with the user's terminal (**Step 401** of Figure 4). After conducting an information transaction with the selected WWW serving node for some interval (determined in relation to the specific application accessed), the user clicks on a hypertext link, or picture, to begin an automated process which will cause that public session to be terminated and a new connection established to an alternate private data network (**Step 402**).

In response to that user action, a data message containing parameter reconfiguration instructions is passed from the WWW server application to the Bridging Software at the user's terminal (**Step 403**). Upon receiving such instructions, the Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (**Step 404**). This reconfiguration is fully automatic and transparent to the user, and includes parameters such as modem dial number, login, password, and TCP/IP addresses. At that point, the Bridging Software causes the modem to disconnect the current data network connection, shutting down the browser, and to then dial the alternate private data network (**Step 405**).

With the establishment of a connection to the private server on the alternate data network, the user interacts with the alternate data network application as appropriate (**Step 406**), and after an interval completes his activity with the alternate data network and provides an indication of such completion (**Step 407**). A data message containing parameter reconfiguration instructions is then passed from the alternate data network application to the Bridging Software (**Step 408**).

At that point, the Bridging Software again edits the user's on-line communications software parameters, reconfiguring them to dial the original public data network, or another preselected network (**Step 409**). As with the first reconfiguration, this configuration is automatic and includes parameters such as modem dial number, login, password, and TCP/IP addresses. The Bridging Software automatically causes the current private data network to be disconnected by the modem (**Step 410**), and if appropriate, causes the original public data network to be redialed (**Step 411**). When such a reconnection to the public data network is established, the end-user would then continue his application in the public data network.

B. "Shopping Cart" Capability

With this configuration, illustrated in flow chart form in Figure 5, a user begins by establishing a connection to a WWW application (assuming for the moment that the application is non-CGI enabled) at a serving node for that application, using the Internet browser and modem associated with the user's terminal (**Step 501** of Figure 5). Upon finding an item in that application to be saved, or remembered for later consideration, or purchase, the user clicks on a hypertext link, or picture, representing that item (**Step 502**). That application then sends a data message to the Bridging Software containing information about the items selected (**Step 503**) and such information is stored by the Bridging Software in the "shopping cart" file in the user's terminal (**Step 504**). Such selection download and storage steps (*i.e.*, steps 502, 503 & 504) are repeated for as many items as the user chooses to select. At any point after the Bridging Software has received the first set of item selection information, the user can instruct the Bridging Software to cause those selected items about which such information has been received to be displayed locally (at the user's terminal), where the user may review or edit (including deletion if desired) the collection of items theretofore selected. The application may also control display characteristics such as color and font for such locally displayed items. Note that in the case of a CGI-enabled application, the application itself will keep track of the items selected by the user and only download the totality of the selected items at the end of the selection process, and accordingly, the described local display option will not be applicable to such a CGI-enabled application.

At the point of completion of his "shopping", the user clicks on a hyper-text link or picture to "check out" (**Step 505**), which will begin a process of causing a jump to an alternate data network for the completion of sensitive portions of the transaction. To that end, a data message containing parameter reconfiguration instructions is passed from the WWW application to the Bridging Software (**Step 506**). It is to be noted that, as a security measure, information such as the new dial number, IP address, home page, configuration data (*e.g.*, login, password, DNS address) may be passed over the public network in encrypted form.

Upon receiving such reconfiguration instructions, the Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (**Step 507**). This reconfiguration is fully automatic and transparent to the user, and includes parameters such as modem dial number, login, password, and TCP/IP addresses. At that point, the Bridging Software causes the modem to disconnect the current data network connection, shutting down the browser, and to then dial the alternate data network (**Step 508**).

The Bridging Software passes the stored "shopping cart" data captured from the WWW application to the alternate network application (**Step 509**), where that data may be displayed for the user, permitting the user to confirm and/or modify the data (**Step 510**). The user interacts with the alternate data network application as appropriate, and after an interval completes his activity with the alternate data network (**Step 511**) and thus, by providing an appropriate completion signal to the application, completing the private portion of the information transaction (**Step 512**). A data message containing parameter reconfiguration instructions is then passed from the alternate data network application to the Bridging Software (**Step 513**).

The Bridging Software, at this point, again edits the user's on-line communications software parameters, reconfiguring them to dial the original (or another pre-defined) data network (**Step 514**). As with the first reconfiguration, this configuration is automatic and includes parameters such as modem dial number, login, password, and TCP/IP addresses. The Bridging Software automatically causes the current private data network to be disconnected by the modem (**Step 515**), and if appropriate, causes the original public data network to be redialed (**Step 516**). When such a reconnection is established to the point in the public data network where the user had left off to handle the secured aspects of his information transaction, the user would then continue his application in the public data network.

C. <u>Stored Configuration Capabilities</u>

For this configuration, depicted in flow chart form in Figure 6, an end-user is connected to a chosen WWW serving node (where a desired information product is made available) via a modem and an Internet browser associated with the user's terminal (**Step 601** of Figure 6). The user selects a hypertext link or picture associated with the WWW application by clicking on such link or picture (**Step 602**). A data message containing parameter reconfiguration instructions and an application icon (related to the selected hypertext link or picture) is passed from the WWW application to the Bridging Software (**Step 603**).

The Bridging Software creates an icon for display at the user's terminal, and saves a Bridging Software configuration file that is associated with that icon (**Step 604**). Such Bridging Software actions are automatic and multiple selections may he captured in this manner. At this point the user may continue the on-line session, or, if all desired selections have been made, a signal is provided from the user that the session should be discontinued (**Step 605**). The Bridging Software then automatically disconnects the current data network connection (**Step 606**).

After disconnecting from the WWW application, and following an interval determined by the user, a new application is selected by the user by clicking on the appropriate new icon displayed at the user's terminal (**Step 607**). The Bridging Software receives the reconfiguration instructions from the file associated with the selected icon (**Step 608**).

The Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (**Step 609**). The Bridging Software then automatically starts the user's Internet browser software and causes the alternate network application to be dialed by the modem associated with that terminal (**Step 610**). Upon establishing a connection to the alternate network, the user interacts with that application and completes the transaction to the user's satisfaction (**Step 611**). After a signal is sent to the alternate network indicating such completion of the user's activity (**Step 612**), a data message containing parameter reconfiguration instructions is passed from the alternate data network application to the Bridging Software (**Step 613**). That Software then causes the user's terminal configuration parameters to be reset (**Step 614**) and the alternate data network to be automatically disconnected (**Step 615**).

D. Off-Line Form Capability

In this configuration, depicted in flow chart form in Figure 7, an end-user is connected to a chosen WWW serving node (where a desired information product is made available) via a modem and an Internet browser associated with the user's terminal (**Step 701** of Figure 7). The user selects a hypertext link or picture associated with an off-line form application -- an exemplary such form being an HTML-based form -- by clicking on such link or picture (**Step 702**). A data message containing parameter reconfiguration instructions for the Bridging Software, the selected off-line-form application, and an optional icon (related to the selected hypertext link or picture) is passed from the WWW application to the Bridging Software (**Step 703**). Note that the selected off-line form may be for either single or multiple use.

In the case of a delayed or multiple use of the selected form, the Bridging Software may create an icon for display at the user's terminal, and will save a Bridging Software configuration file that is associated with that icon (**Step 704**). The form in question is also saved on the user's terminal. Such Bridging Software actions are automatic. At this point the user may continue the on-line session, or, if all desired selections have been made, a signal is provided from the user that the session should be discontinued (**Step 705**). The Bridging Software then automatically disconnects the current data network connection (**Step 706**).

After disconnecting from the WWW application, two cases are to be considered as to the further processing of the selected form: (1) an immediate single use of the form and (2) either a delayed or multiple use of the form. In the first case, the Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network. The Bridging Software then automatically starts the user's Internet browser software which is caused to display the off-line form. The user then completes the off-line form and chooses a "Submit Form" button displayed at his terminal.

In the second case, the Bridging Software will have created an icon for display at the user's terminal and saved a Bridging Software configuration file associated with that icon. Following an interval determined by the user, the off-line-form application is started by the user by clicking on the new form icon displayed at the user's terminal (**Step 707**). The Bridging Software receives the reconfiguration instructions from the file associated with the selected icon (**Step 708**).

The Bridging Software edits the user's on-line communications software parameters, reconfiguring that software to dial the alternate data network (**Step 709**). The Bridging Software then automatically starts the user's Internet browser software which is caused to display the off-line form (**Step 710**). The user then completes the off-line form and chooses a "Submit Form" button displayed at his terminal (**Step 711**).

In either the first or second case, following activation of the "Submit Form" button, the alternate network application is then caused to be dialed by the Bridging Software. Upon establishing a connection to the alternate network, the form data is passed to the alternate network (**Step 712**). The user then interacts with that application and completes the application (**Step 713**). After a signal is sent to the alternate network indicating such completion of the user's activity (**Step 714**),a data message containing parameter reconfiguration instructions is passed from the alternate data network application to the Bridging Software (**Step 715**). That Software then causes the user's terminal configuration parameters to be reset (**Step 716**) and the alternate data network to be automatically disconnected (**Step 717**).

CONCLUSION

A system and method has been described for the automatic switching of an information transaction between two or more alternate networks. This functionality, which incorporates a reconfiguration means designated herein as the Bridging Software, supports the movement of application specific data from one on-line environment to another. Among potential applications of this process for passing data between different environments are: selected items for purchase ("shopping cart"), captured data from forms, and other server captured data such as web pages visited.

The Bridging Software reconfiguration means is intended to work with various Web Browser software implementations, including the Netscape Personal Edition (NPE) Software for Windows 3.1 and 3.11, and which represents a working embodiment for the invention. The Bridging Software installs itself as a helper application within the browser application and utilizes a special MIME type configuration file to pass reconfiguration and "shopping cart" information from the server to the client software.

When an application requires a user to re-connect to a private application, a reconfiguration file is passed to the Bridging Software helper application via a CGI script or simple hyper-text link. The helper application disconnects the current data connection, reconfigures the dial parameters (dial #, login password, DNS address, and home page) and initiates the dial program so the end-user can access the private application.

When the end-user connects to the private application, the Bridging Software reconfiguration means provides the new "private server" application with data collected from the "public server", and the application resumes in a private, secure environment.

The Bridging Software allows both short term and long term storage of dial configurations. Configurations passed to the Bridging Software can be designated as single use configurations and discarded after the application has terminated, or saved and displayed to the end-user as a dial choice by the Bridging Software.

Although the present embodiment of the invention has been described in detail, it should be understood that various changes, alterations and substitutions can be made therein without departing from the spirit and scope of the invention as defined by the appended claims. In particular, it is noted that, while the invention has been primarily described in terms of a preferred embodiment based on an automatic reconfiguration between a public and a private data network, any the methodology of the invention will be equally applicable to any set of alternate networks.

Claims

1. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

   establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;
   receiving information from said serving node in said first data network for effecting a reconfiguration of said communications path for said transaction from said first connection in said first data network to a second connection in a second data network; and
   automatically connecting said terminal device to a serving node in said second data network via said second connection.

2. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

   establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;
   selecting at least one information item from a data base of said information items provided at said serving node in said first data network;
   causing said selected information items to be downloaded to said terminal device via said first connection;
   receiving information from said serving node in said first data network for effecting a reconfiguration of said communications path for said transaction from said first connection in said first data network to a second connection in a second data network; and
   automatically connecting said terminal device to a serving node in said second data network via said second connection.

3. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

   establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;
   identifying at least one data network application from a data base of said data network applications provided at said serving node in said first data network;
   receiving information from said serving node in said first data network for reconfiguring said terminal device for implementation of a communication path via an alternate connection between said terminal device and at least one of said identified data network applications in a second data network; and
   in response to a selection signal from a user, automatically connecting said terminal device to a selected one of said identified data network applications via said alternate connection.

4. A method for managing a transaction via a communications path between a terminal device and a serving node in a data network, said method comprising the steps of:

   establishing an initial communications path via a first connection between said terminal device and a serving node in a first data network;
   selecting an off-line form application from a data base provided at said serving node in said first data network;
   receiving information from said serving node in said first data network for reconfiguring said terminal device for implementation of a communication path via a second connection between said terminal device and said

selected off-line form application in a second data network; and

in response to, a selection signal from a user, automatically connecting said terminal device to said selected off-line form application.

5. The method for managing a transaction of Claim 1 or 2 including the further step of recognizing a signal to reconfigure said communications path from said first connection to said second connection.

6. The method for managing a transaction of Claim 3 wherein said selected data network application is operated at a serving node in said second data network.

7. The method for managing a transaction of Claim 4 wherein said selected off-line form application is operated at a serving node in said second data network.

8. The method for managing a transaction of one of the Claims 1, 2, 6 or 7 wherein said serving nodes in said first and said second data networks are manifested in a common node.

9. The method for managing a transaction of Claim 1 or 2 wherein said step of receiving information includes the further step of effecting said reconfiguration of said communications path.

10. The method for managing a transaction of Claim 1 or 2 wherein said step of automatically connecting includes the step of automatically disconnecting said first connection prior to implementation of said second connection.

11. The method for managing a transaction of Claim 1 or 2 including the further steps of:

automatically disconnecting said second connection in response to a user signal; and

reconfiguring said terminal device to enable, in response to user instruction, an implementation of a connection via an identified data network.

12. The method for managing a transaction of Claim 11 wherein said step of automatically reconfiguring said terminal device includes the step of effecting said implementation of said connection via said identified data network.

13. The method for managing a transaction of Claim 2 wherein said step of causing said selected information items to be downloaded includes the further step of causing said selected information items to be displayed at said terminal device.

14. The method for managing a transaction of Claim 13 wherein said displayed selected items can be edited by a user at said terminal device.

15. The method for managing a transaction of Claim 13 wherein display characteristics for said displayed selected items can be controlled at said terminal device.

16. The method for managing a transaction of Claim 2 wherein said step of automatically connecting includes the step of uploading said selected information items from said terminal device to said service provider via said second connection.

17. The method for managing a transaction of Claim 3 including the further steps of:

automatically disconnecting said alternate connection in response to a user signal; and

reconfiguring said terminal device to enable implementation of a pre-selected connection between said terminal device and an identified data network.

18. The method for managing a transaction of Claim 17 wherein said step of automatically reconfiguring said terminal device includes the further step of effecting said implementation of said pre-selected connection.

19. The method for managing a transaction of Claim 4 including the further step of downloading from said serving node in said first data network to said terminal device of an off-line form related to said off-line form application.

20. The method for managing a transaction of Claim 4 including the further step of uploading said downloaded off-line

form from said terminal device to said selected off-line form application, after processing by a user.

21. The method for managing a transaction of Claim 4 including the further steps of:

automatically disconnecting said connection to said selected off-line form application in response to a user signal; and
reconfiguring said terminal device to enable implementation of a pre-selected connection between said terminal device and an identified data network.

22. The method for managing a transaction of Claim 21 wherein said step of automatically reconfiguring said terminal device includes the further step of effecting said implementation of said pre-selected connection.

23. A method for managing connections between a terminal device and at least one information source/processor wherein at least two of said connections are implemented via separate communications networks, comprising the steps of:

recognizing a signal for connection to an information source/processor via a communications network other than a communications network for which a predetermined connection is configured;
causing said terminal device to implement a connection to said information source/processor via said other communications network; and
upon termination of said information source/processor connection via said other communications network, automatically reconfiguring a connection criteria in said terminal device to enable said terminal device to implement, in response to user instruction, a connection via an alternative one of said communications networks.

24. The method for managing connections of Claim 23 wherein said recognizing step occurs at a point when said terminal device is connected to a given source/processor.

25. The method for managing connections of Claim 23 wherein information items may be selected by a user at said terminal device from said given source/processor, and including the further step of causing said selected information items to be downloaded from said source/processor to said terminal device.

26. The method for managing connections of Claim 25 wherein said step of effecting connection includes the further step of uploading said selected information items from said terminal device to said other information source/processor.

27. The method for managing connections of Claim 26 wherein said selected information items are processed by said user at said terminal device prior to uploading to said other information source/processor.

28. The method for managing connections of Claim 24 including the further step of causing said given source/processor to download to said terminal device configuration data for enabling said step of effecting connection to said other information source/processor.

29. The method for managing connections of Claim 24 including the further step of causing said other source/processor to download to said terminal device configuration data for enabling said step of automatically restoring a prior connection criteria in said terminal device.

30. A method for enhancing security of certain data in an on-line information transaction comprising the steps of:

bifurcating said information transaction into a first portion comprising said certain data and a remaining portion, wherein said remaining portion is carried out via a public on-line communications connection between a terminal device and a public information server;
causing said first portion to be carried out via a secure private on-line communications connection between said terminal device and a private information server; and
automatically reconfiguring network access means in said terminal device to switch between said public connection and said private connection.

FIG. 1

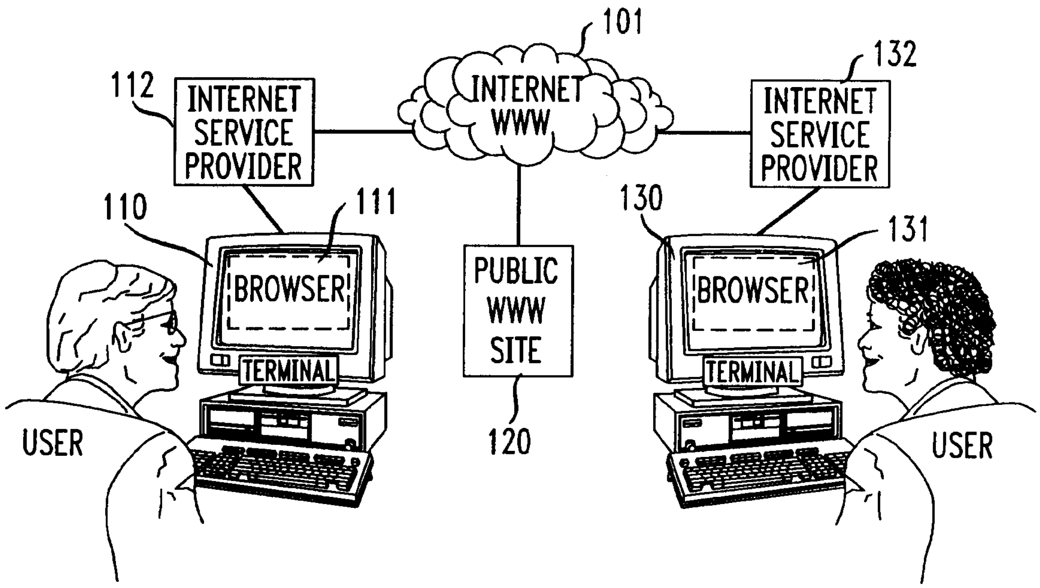

101
INTERNET
WWW
112
INTERNET
SERVICE
PROVIDER
132
INTERNET
SERVICE
PROVIDER
110
111
BROWSER
TERMINAL
PUBLIC
WWW
SITE
120
130
131
BROWSER
TERMINAL
USER
USER

FIG. 2

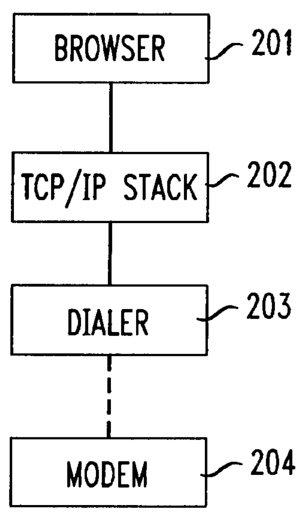

BROWSER
201
TCP/IP STACK
202
DIALER
203
MODEM
204

FIG. 3

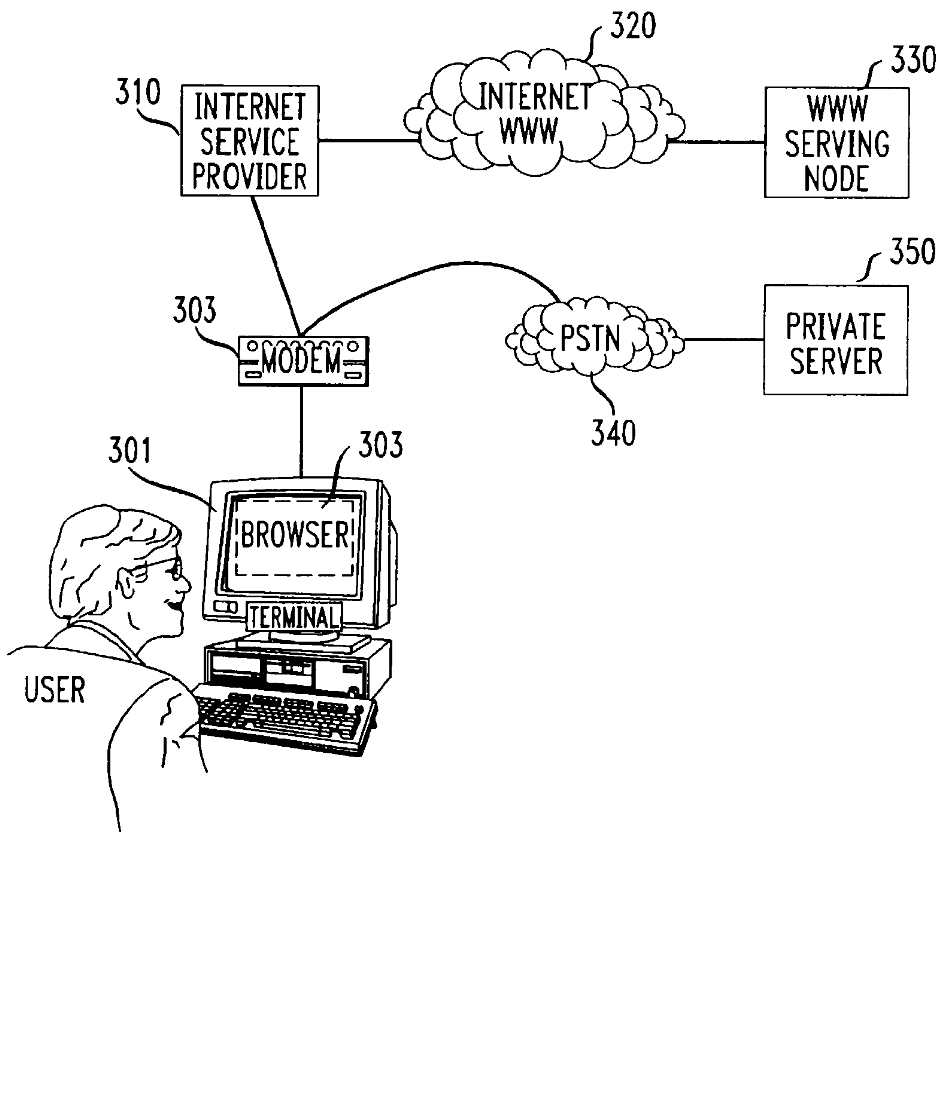

320
330
310
INTERNET SERVICE PROVIDER
INTERNET WWW
WWW SERVING NODE
350
303
PSTN
PRIVATE SERVER
MODEM
340
301
303
BROWSER
TERMINAL
USER

FIG. 4

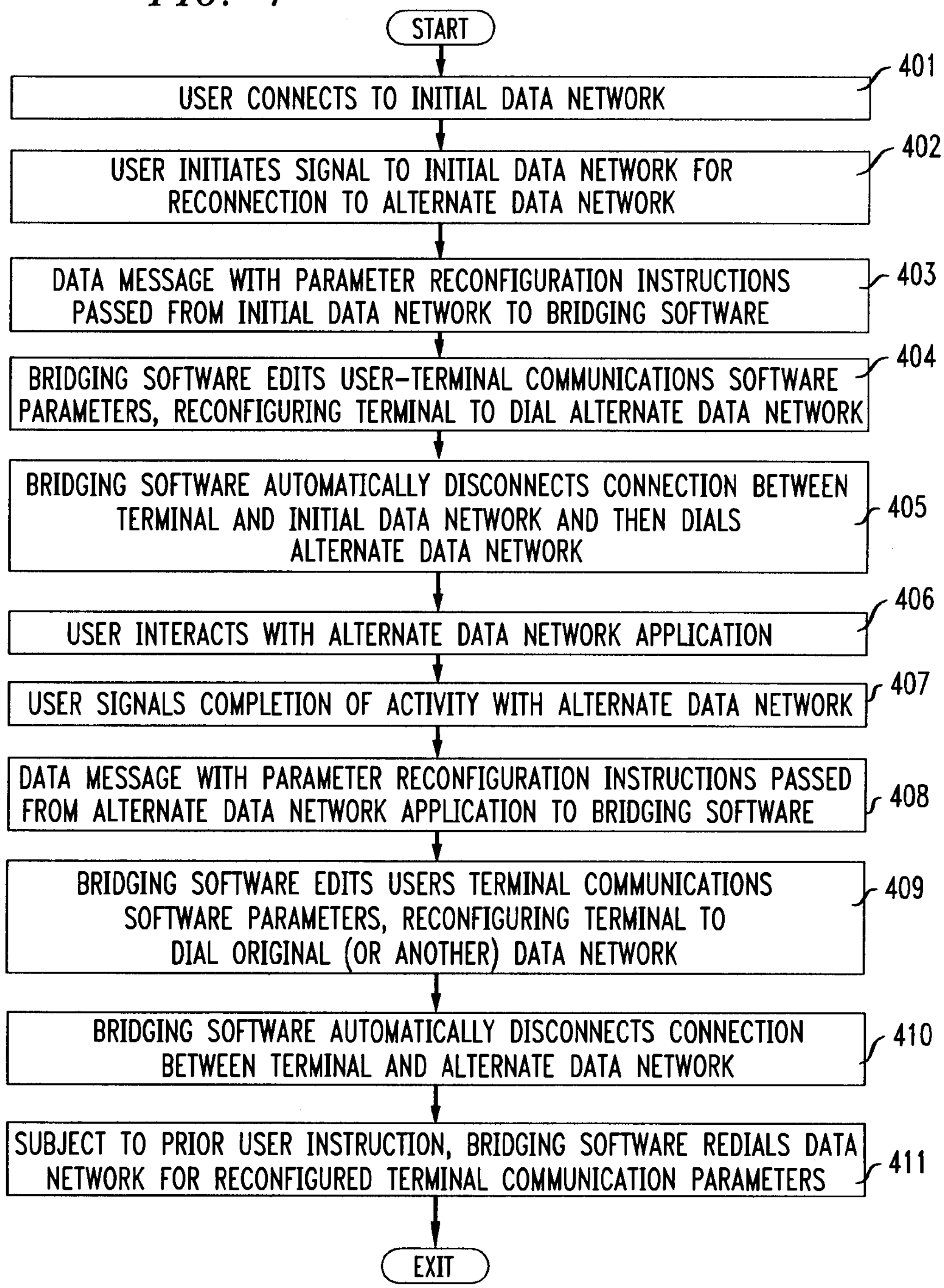

START
USER CONNECTS TO INITIAL DATA NETWORK
401
USER INITIATES SIGNAL TO INITIAL DATA NETWORK FOR RECONNECTION TO ALTERNATE DATA NETWORK
402
DATA MESSAGE WITH PARAMETER RECONFIGURATION INSTRUCTIONS PASSED FROM INITIAL DATA NETWORK TO BRIDGING SOFTWARE
403
BRIDGING SOFTWARE EDITS USER-TERMINAL COMMUNICATIONS SOFTWARE PARAMETERS, RECONFIGURING TERMINAL TO DIAL ALTERNATE DATA NETWORK
404
BRIDGING SOFTWARE AUTOMATICALLY DISCONNECTS CONNECTION BETWEEN TERMINAL AND INITIAL DATA NETWORK AND THEN DIALS ALTERNATE DATA NETWORK
405
USER INTERACTS WITH ALTERNATE DATA NETWORK APPLICATION
406
USER SIGNALS COMPLETION OF ACTIVITY WITH ALTERNATE DATA NETWORK
407
DATA MESSAGE WITH PARAMETER RECONFIGURATION INSTRUCTIONS PASSED FROM ALTERNATE DATA NETWORK APPLICATION TO BRIDGING SOFTWARE
408
BRIDGING SOFTWARE EDITS USERS TERMINAL COMMUNICATIONS SOFTWARE PARAMETERS, RECONFIGURING TERMINAL TO DIAL ORIGINAL (OR ANOTHER) DATA NETWORK
409
BRIDGING SOFTWARE AUTOMATICALLY DISCONNECTS CONNECTION BETWEEN TERMINAL AND ALTERNATE DATA NETWORK
410
SUBJECT TO PRIOR USER INSTRUCTION, BRIDGING SOFTWARE REDIALS DATA NETWORK FOR RECONFIGURED TERMINAL COMMUNICATION PARAMETERS
411
EXIT

FIG. 5A

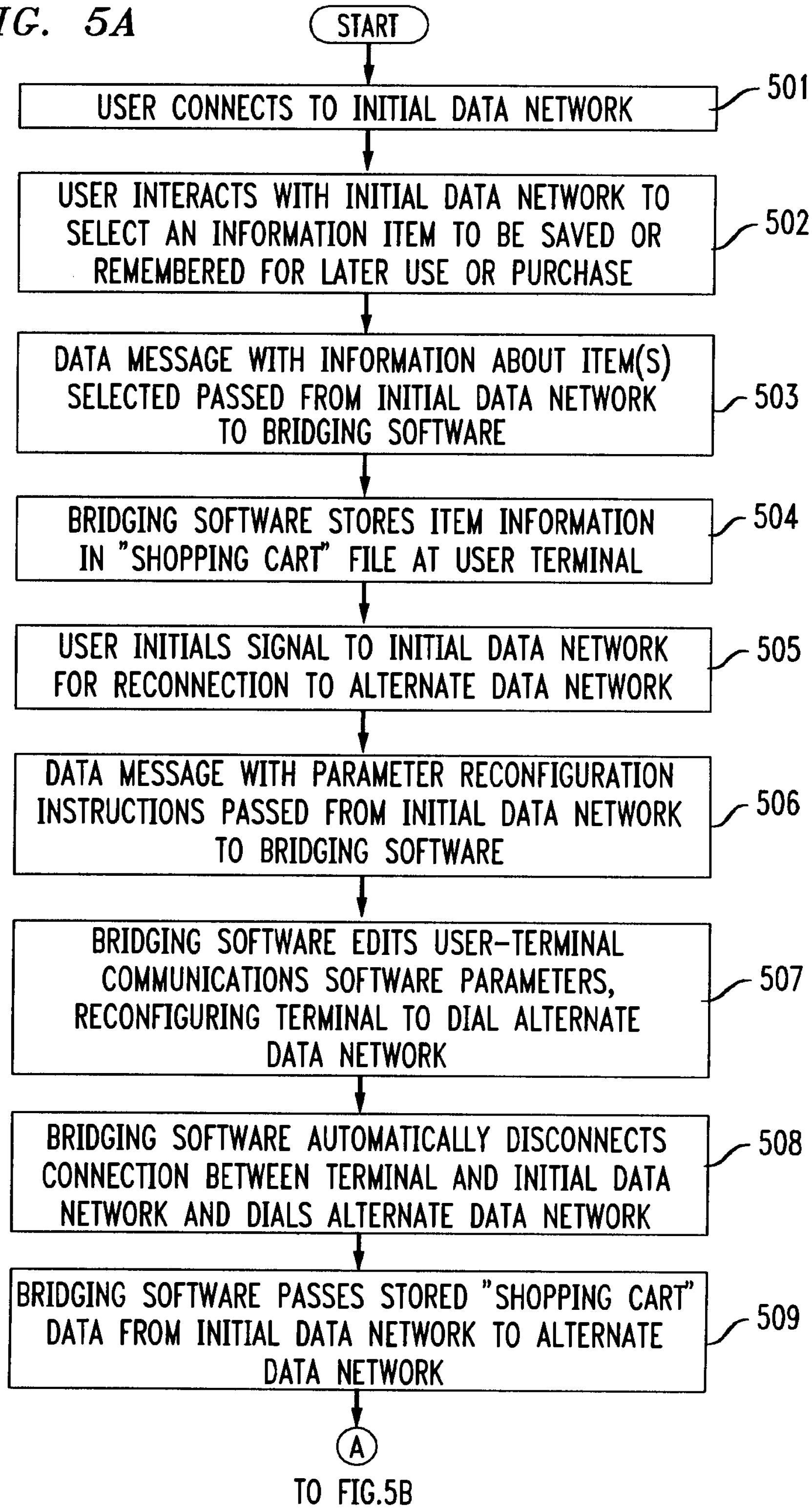

START
USER CONNECTS TO INITIAL DATA NETWORK
501
USER INTERACTS WITH INITIAL DATA NETWORK TO SELECT AN INFORMATION ITEM TO BE SAVED OR REMEMBERED FOR LATER USE OR PURCHASE
502
DATA MESSAGE WITH INFORMATION ABOUT ITEM(S) SELECTED PASSED FROM INITIAL DATA NETWORK TO BRIDGING SOFTWARE
503
BRIDGING SOFTWARE STORES ITEM INFORMATION IN "SHOPPING CART" FILE AT USER TERMINAL
504
USER INITIALS SIGNAL TO INITIAL DATA NETWORK FOR RECONNECTION TO ALTERNATE DATA NETWORK
505
DATA MESSAGE WITH PARAMETER RECONFIGURATION INSTRUCTIONS PASSED FROM INITIAL DATA NETWORK TO BRIDGING SOFTWARE
506
BRIDGING SOFTWARE EDITS USER-TERMINAL COMMUNICATIONS SOFTWARE PARAMETERS, RECONFIGURING TERMINAL TO DIAL ALTERNATE DATA NETWORK
507
BRIDGING SOFTWARE AUTOMATICALLY DISCONNECTS CONNECTION BETWEEN TERMINAL AND INITIAL DATA NETWORK AND DIALS ALTERNATE DATA NETWORK
508
BRIDGING SOFTWARE PASSES STORED "SHOPPING CART" DATA FROM INITIAL DATA NETWORK TO ALTERNATE DATA NETWORK
509
A
TO FIG.5B

FIG. 5B

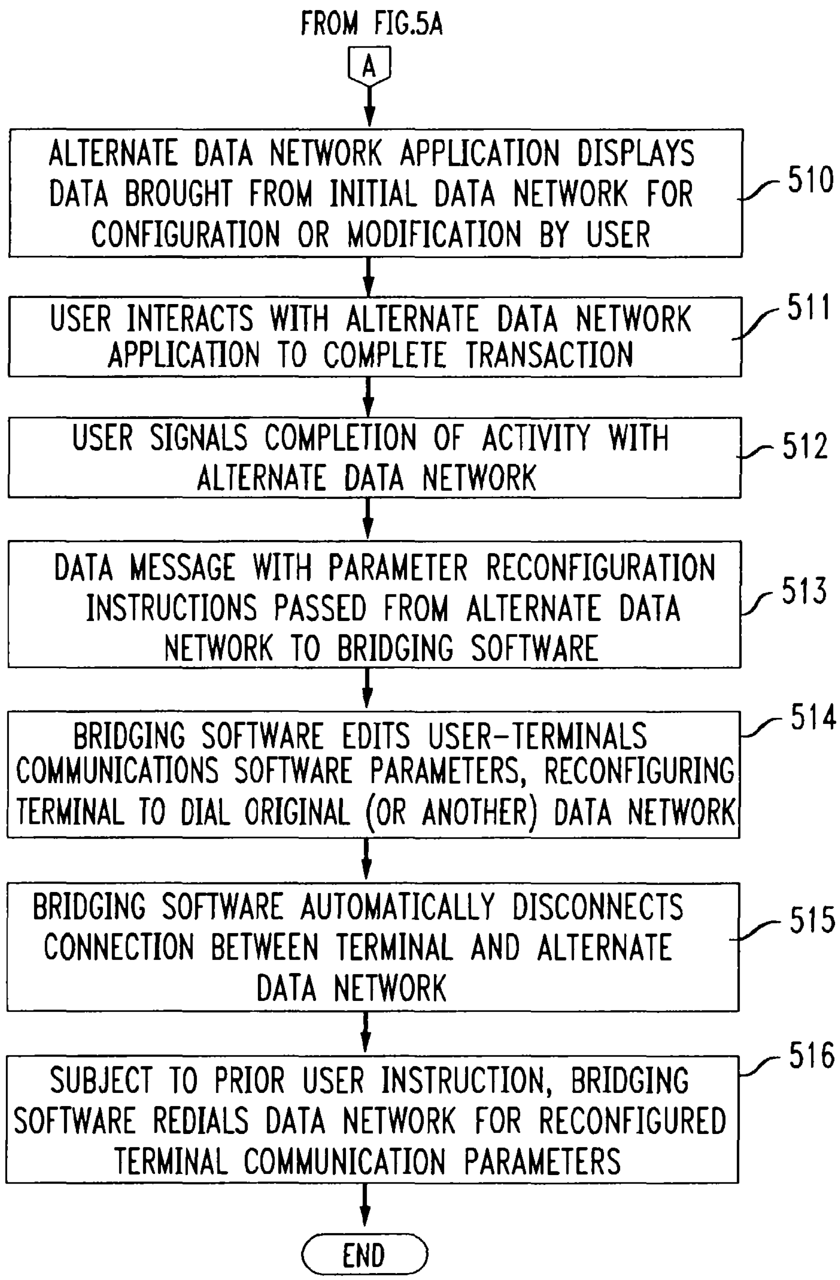

FROM FIG.5A
A
ALTERNATE DATA NETWORK APPLICATION DISPLAYS DATA BROUGHT FROM INITIAL DATA NETWORK FOR CONFIGURATION OR MODIFICATION BY USER
510
USER INTERACTS WITH ALTERNATE DATA NETWORK APPLICATION TO COMPLETE TRANSACTION
511
USER SIGNALS COMPLETION OF ACTIVITY WITH ALTERNATE DATA NETWORK
512
DATA MESSAGE WITH PARAMETER RECONFIGURATION INSTRUCTIONS PASSED FROM ALTERNATE DATA NETWORK TO BRIDGING SOFTWARE
513
BRIDGING SOFTWARE EDITS USER-TERMINALS COMMUNICATIONS SOFTWARE PARAMETERS, RECONFIGURING TERMINAL TO DIAL ORIGINAL (OR ANOTHER) DATA NETWORK
514
BRIDGING SOFTWARE AUTOMATICALLY DISCONNECTS CONNECTION BETWEEN TERMINAL AND ALTERNATE DATA NETWORK
515
SUBJECT TO PRIOR USER INSTRUCTION, BRIDGING SOFTWARE REDIALS DATA NETWORK FOR RECONFIGURED TERMINAL COMMUNICATION PARAMETERS
516
END

FIG. 6A

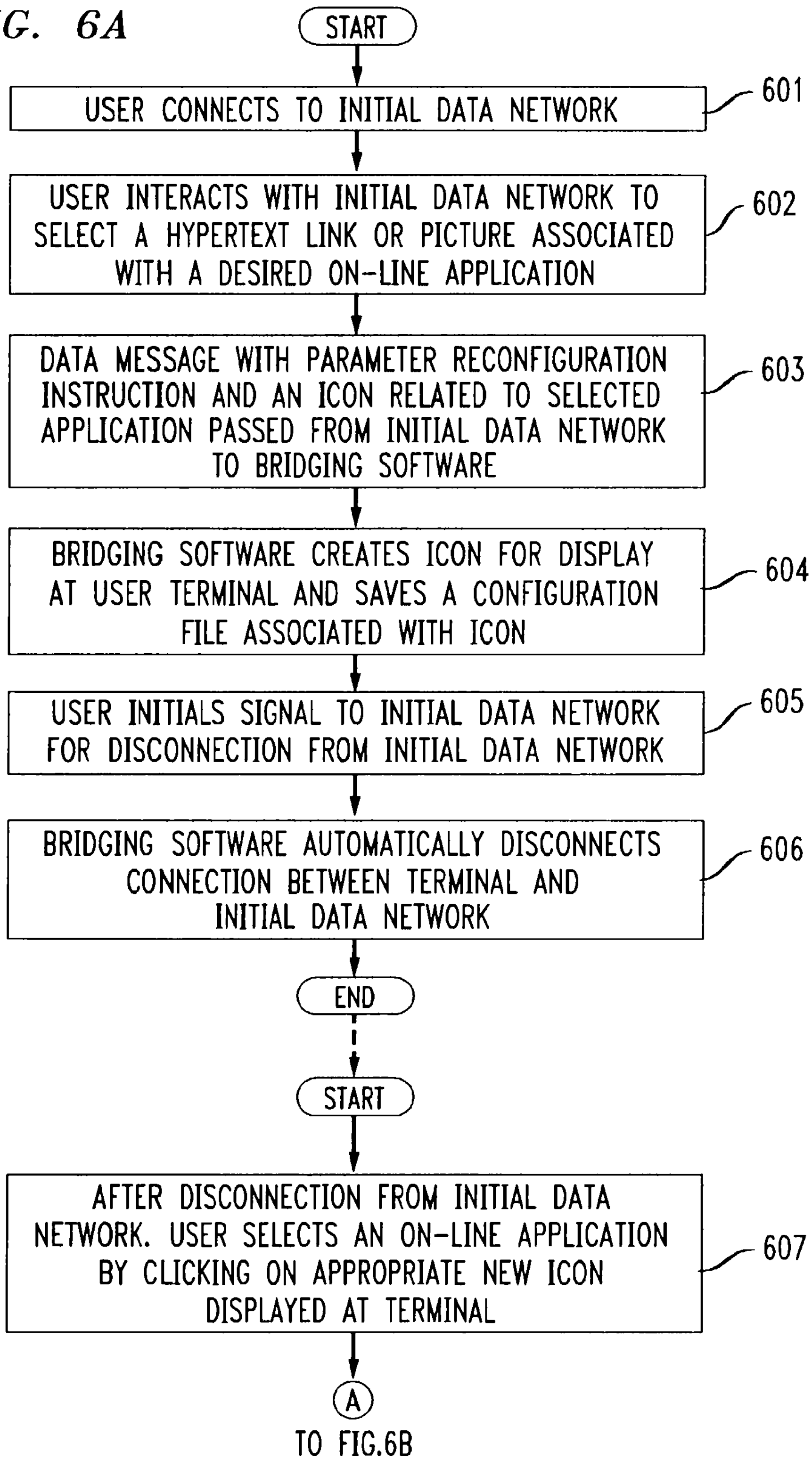

START
USER CONNECTS TO INITIAL DATA NETWORK
601
USER INTERACTS WITH INITIAL DATA NETWORK TO SELECT A HYPERTEXT LINK OR PICTURE ASSOCIATED WITH A DESIRED ON-LINE APPLICATION
602
DATA MESSAGE WITH PARAMETER RECONFIGURATION INSTRUCTION AND AN ICON RELATED TO SELECTED APPLICATION PASSED FROM INITIAL DATA NETWORK TO BRIDGING SOFTWARE
603
BRIDGING SOFTWARE CREATES ICON FOR DISPLAY AT USER TERMINAL AND SAVES A CONFIGURATION FILE ASSOCIATED WITH ICON
604
USER INITIALS SIGNAL TO INITIAL DATA NETWORK FOR DISCONNECTION FROM INITIAL DATA NETWORK
605
BRIDGING SOFTWARE AUTOMATICALLY DISCONNECTS CONNECTION BETWEEN TERMINAL AND INITIAL DATA NETWORK
606
END
START
AFTER DISCONNECTION FROM INITIAL DATA NETWORK. USER SELECTS AN ON-LINE APPLICATION BY CLICKING ON APPROPRIATE NEW ICON DISPLAYED AT TERMINAL
607
A
TO FIG.6B

FIG. 6B

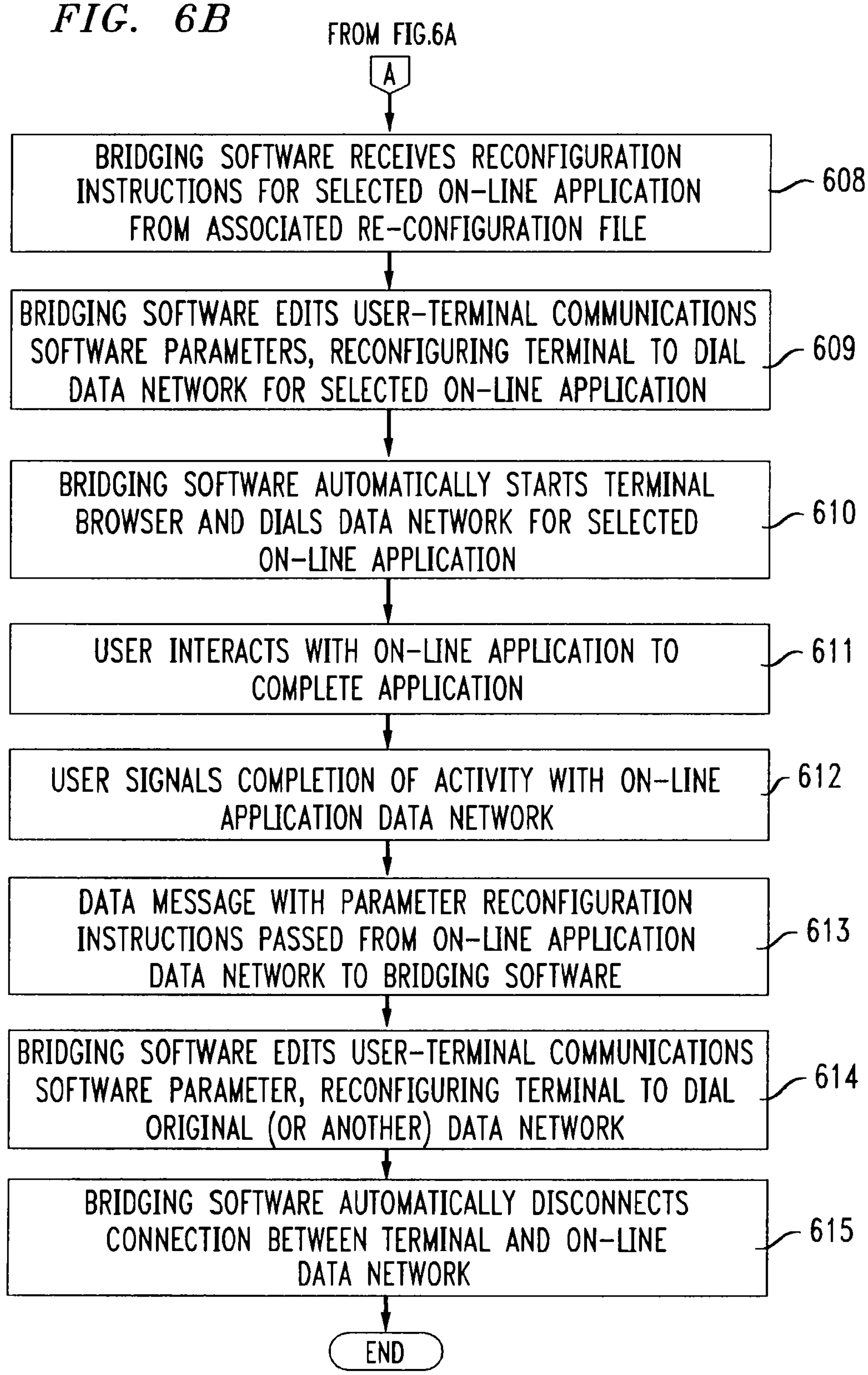

FROM FIG.6A
A
BRIDGING SOFTWARE RECEIVES RECONFIGURATION INSTRUCTIONS FOR SELECTED ON-LINE APPLICATION FROM ASSOCIATED RE-CONFIGURATION FILE
608
BRIDGING SOFTWARE EDITS USER-TERMINAL COMMUNICATIONS SOFTWARE PARAMETERS, RECONFIGURING TERMINAL TO DIAL DATA NETWORK FOR SELECTED ON-LINE APPLICATION
609
BRIDGING SOFTWARE AUTOMATICALLY STARTS TERMINAL BROWSER AND DIALS DATA NETWORK FOR SELECTED ON-LINE APPLICATION
610
USER INTERACTS WITH ON-LINE APPLICATION TO COMPLETE APPLICATION
611
USER SIGNALS COMPLETION OF ACTIVITY WITH ON-LINE APPLICATION DATA NETWORK
612
DATA MESSAGE WITH PARAMETER RECONFIGURATION INSTRUCTIONS PASSED FROM ON-LINE APPLICATION DATA NETWORK TO BRIDGING SOFTWARE
613
BRIDGING SOFTWARE EDITS USER-TERMINAL COMMUNICATIONS SOFTWARE PARAMETER, RECONFIGURING TERMINAL TO DIAL ORIGINAL (OR ANOTHER) DATA NETWORK
614
BRIDGING SOFTWARE AUTOMATICALLY DISCONNECTS CONNECTION BETWEEN TERMINAL AND ON-LINE DATA NETWORK
615
END

FIG. 7A

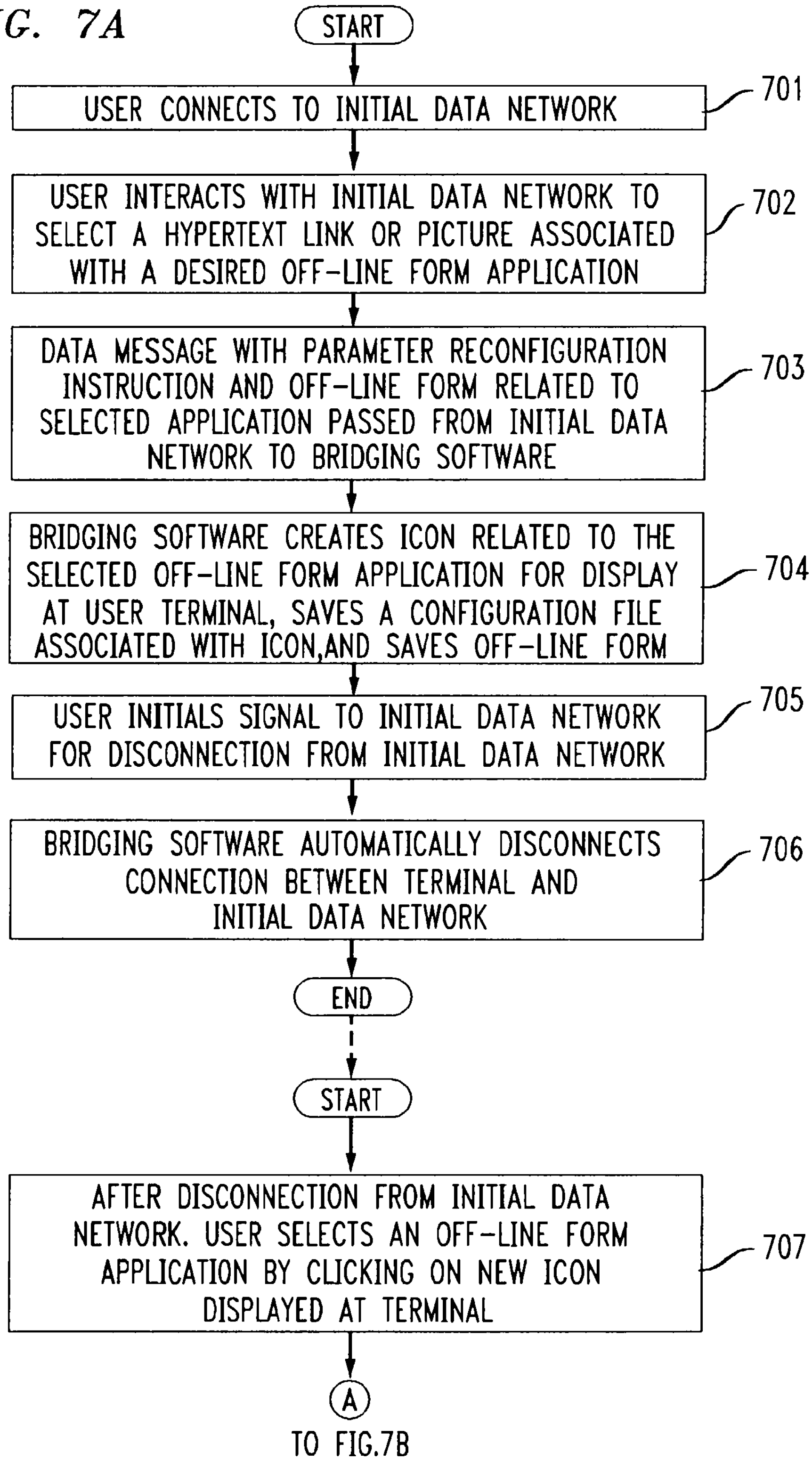
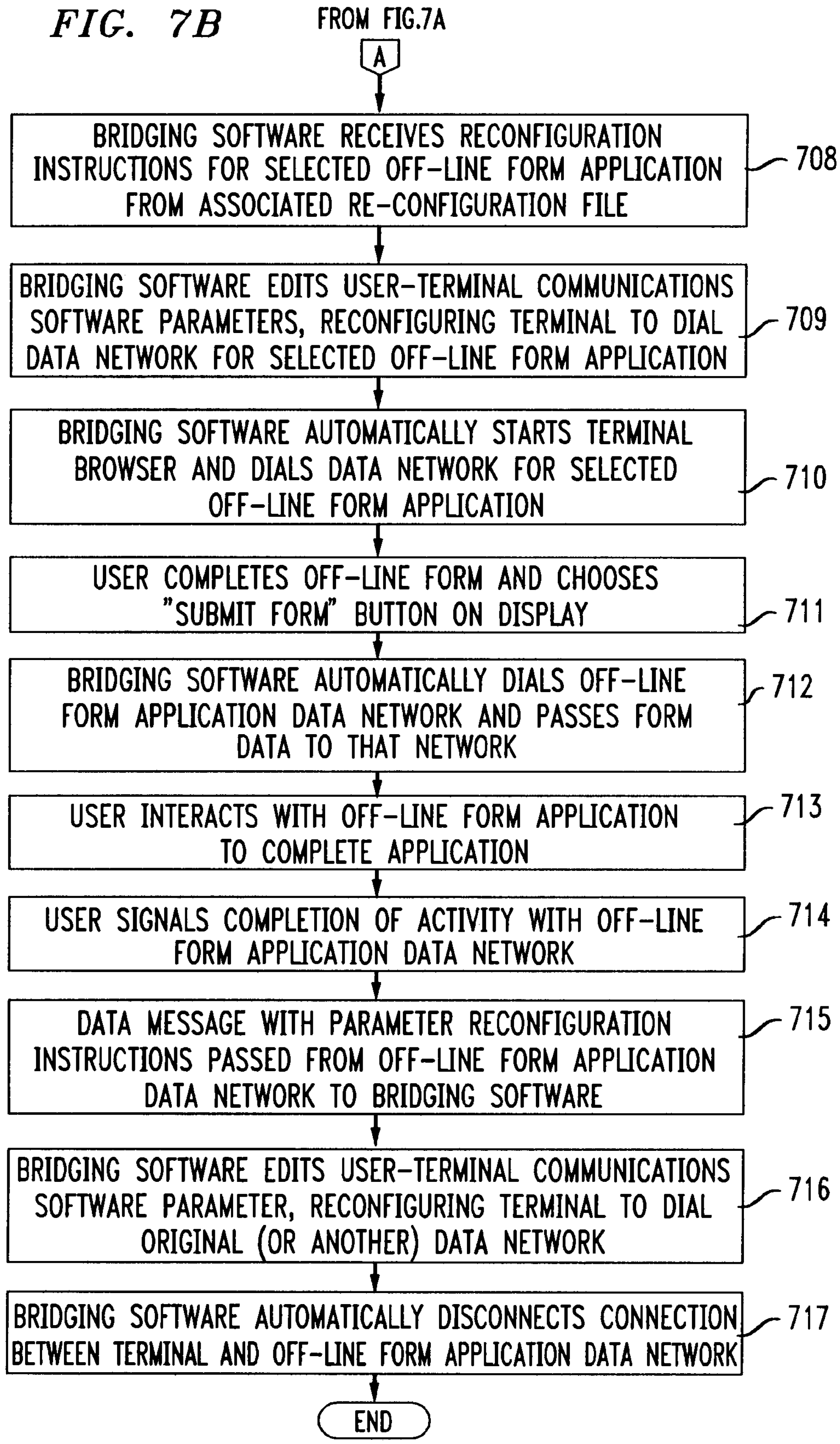

START
USER CONNECTS TO INITIAL DATA NETWORK
701
USER INTERACTS WITH INITIAL DATA NETWORK TO SELECT A HYPERTEXT LINK OR PICTURE ASSOCIATED WITH A DESIRED OFF-LINE FORM APPLICATION
702
DATA MESSAGE WITH PARAMETER RECONFIGURATION INSTRUCTION AND OFF-LINE FORM RELATED TO SELECTED APPLICATION PASSED FROM INITIAL DATA NETWORK TO BRIDGING SOFTWARE
703
BRIDGING SOFTWARE CREATES ICON RELATED TO THE SELECTED OFF-LINE FORM APPLICATION FOR DISPLAY AT USER TERMINAL, SAVES A CONFIGURATION FILE ASSOCIATED WITH ICON,AND SAVES OFF-LINE FORM
704
USER INITIALS SIGNAL TO INITIAL DATA NETWORK FOR DISCONNECTION FROM INITIAL DATA NETWORK
705
BRIDGING SOFTWARE AUTOMATICALLY DISCONNECTS CONNECTION BETWEEN TERMINAL AND INITIAL DATA NETWORK
706
END
START
AFTER DISCONNECTION FROM INITIAL DATA NETWORK. USER SELECTS AN OFF-LINE FORM APPLICATION BY CLICKING ON NEW ICON DISPLAYED AT TERMINAL
707
A
TO FIG.7B

FIG. 7B


FROM FIG.7A
A
BRIDGING SOFTWARE RECEIVES RECONFIGURATION INSTRUCTIONS FOR SELECTED OFF-LINE FORM APPLICATION FROM ASSOCIATED RE-CONFIGURATION FILE
708
BRIDGING SOFTWARE EDITS USER-TERMINAL COMMUNICATIONS SOFTWARE PARAMETERS, RECONFIGURING TERMINAL TO DIAL DATA NETWORK FOR SELECTED OFF-LINE FORM APPLICATION
709
BRIDGING SOFTWARE AUTOMATICALLY STARTS TERMINAL BROWSER AND DIALS DATA NETWORK FOR SELECTED OFF-LINE FORM APPLICATION
710
USER COMPLETES OFF-LINE FORM AND CHOOSES "SUBMIT FORM" BUTTON ON DISPLAY
711
BRIDGING SOFTWARE AUTOMATICALLY DIALS OFF-LINE FORM APPLICATION DATA NETWORK AND PASSES FORM DATA TO THAT NETWORK
712
USER INTERACTS WITH OFF-LINE FORM APPLICATION TO COMPLETE APPLICATION
713
USER SIGNALS COMPLETION OF ACTIVITY WITH OFF-LINE FORM APPLICATION DATA NETWORK
714
DATA MESSAGE WITH PARAMETER RECONFIGURATION INSTRUCTIONS PASSED FROM OFF-LINE FORM APPLICATION DATA NETWORK TO BRIDGING SOFTWARE
715
BRIDGING SOFTWARE EDITS USER-TERMINAL COMMUNICATIONS SOFTWARE PARAMETER, RECONFIGURING TERMINAL TO DIAL ORIGINAL (OR ANOTHER) DATA NETWORK
716
BRIDGING SOFTWARE AUTOMATICALLY DISCONNECTS CONNECTION BETWEEN TERMINAL AND OFF-LINE FORM APPLICATION DATA NETWORK
717
END

EP0838930

Publication Title:

Pseudo network adapter for frame capture, encapsulation and encryption

Abstract:

Abstract of EP0838930

A new pseudo network adapter provides an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network, and includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. A transmit path in the system processes data packets from the local communications protocol stack for transmission through the pseudo network adapter. An encryption engine encrypts the data packets and an encapsulation engine encapsulates the encrypted data packets into tunnel data frames. The network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for de ed1 livery to a user.

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 838 930 A2

(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
29.04.1998 Bulletin 1998/18

(51) Int. Cl.[6]: **H04L 29/06**

(21) Application number: **97118556.6**

(22) Date of filing: **24.10.1997**

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV RO SI

(30) Priority: **25.10.1996 US 738155**

(71) Applicant:
DIGITAL EQUIPMENT CORPORATION
Maynard, Massachusetts 01754 (US)

(72) Inventors:
- **Alden, Kenneth F.**
Boylston, Massachusetts 01505 (US)
- **Lichtenberg, Mitchell P.**
Sunnyvale, CA 94087 (US)
- **Wobber, Edward P.**
Menlo Park, California 94025 (US)

(74) Representative: **Betten & Resch**
Reichenbachstrasse 19
80469 München (DE)

(54) **Pseudo network adapter for frame capture, encapsulation and encryption**

(57) A new pseudo network adapter provides an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network, and includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. A transmit path in the system processes data packets from the local communications protocol stack for transmission through the pseudo network adapter. An encryption engine encrypts the data packets and an encapsulation engine encapsulates the encrypted data packets into tunnel data frames. The network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for delivery to a user.

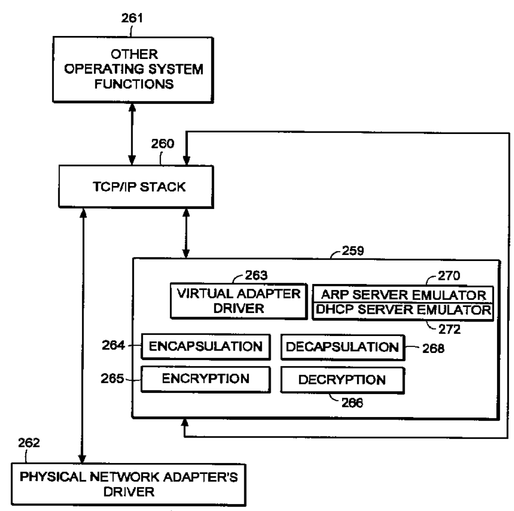


FIG. 15

EP 0 838 930 A2

Printed by Xerox (UK) Business Services
2.16.1/3.4

Description

FIELD OF THE INVENTION

The invention relates generally to establishing secure virtual private networks. The invention relates specifically to a pseudo network adapter for capturing, encapsulating and encrypting messages or frames.

BACKGROUND

In data communications it is often required that secure communications be provided between users of network stations (also referred to as "network nodes") at different physical locations. Secure communications must potentially extend over public networks as well as through secure private networks. Secure private networks are protected by "firewalls", which separate the private network from a public network. Firewalls ordinarily provide some combination of packet filtering, circuit gateway, and application gateway technology, insulating the private network from unwanted communications with the public network.

One approach to providing secure communications is to form a virtual private network. In a virtual private network, secure communications are provided by encapsulating and encrypting messages. Encapsulated messaging in general is referred to as "tunneling". Tunnels using encryption may provide protected communications between users separated by a public network, or among a subset of users of a private network.

Encryption may for example be performed using an encryption algorithm using one or more encryption "keys". When an encryption key is used, the value of the key determines how the data is encrypted and decrypted. When a public-key encryption system is used, a key pair is associated with each communicating entity. The key pair consists of an encryption key and a decryption key. The two keys are formed such that it is unfeasible to generate one key from the other. Each entity makes its encryption key public, while keeping its decryption key secret. When sending a message to node A, for example, the transmitting entity uses the public key of node A to encrypt the message, and then the message can only be decrypted by node A using node A's private key.

In a symmetric key encryption system a single key is used as the basis for both encryption and decryption. An encryption key in a symmetric key encryption system is sometimes referred to as a "shared" key. For example, a pair of communicating nodes A and B could communicate securely as follows: a first shared key is used to encrypt data sent from node A to node B, while a second shared key is to be used to encrypt data sent from node B to node A. In such a system, the two shared keys must be known by both node A and node B. More examples of encryption algorithms and keyed encryption are disclosed in many textbooks, for example "Applied Cryptography - Protocols, Algorithms, and Source Code in C", by Bruce Schneier, published by John Wiley and Sons, New York, New York, copyright 1994.

Information regarding what encryption key or keys are to be used, and how they are to be used to encrypt data for a given secure communications session is referred to as "key exchange material". Key exchange material may for example determine what keys are used and a time duration for which each key is valid. Key exchange material for a pair of communicating stations must be known by both stations before encrypted data can be exchanged in a secure communications session. How key exchange material is made known to the communicating stations for a given secure communications session is referred to as "session key establishment".

A tunnel may be implemented using a virtual or "pseudo" network adapter that appears to the communications protocol stack as a physical device and which provides a virtual private network. A pseudo network adapter must have the capability to receive packets from the communications protocol stack, and to pass received packets back through the protocol stack either to a user or to be transmitted.

A tunnel endpoint is the point at which any encryption/decryption and encapsulation/decapsulation provided by a tunnel is performed. In existing systems, the tunnel end points are pre-determined network layer addresses. The source network layer address in a received message is used to determine the "credentials" of an entity requesting establishment of a tunnel connection. For example, a tunnel server uses the source network layer address to determine whether a requested tunnel connection is authorized. The source network layer address is also used to determine which cryptographic key or keys to use to decrypt received messages.

Existing tunneling technology is typically performed by encapsulating encrypted network layer packets (also referred to as "frames") at the network layer. Such systems provide "network layer within network layer" encapsulation of encrypted messages. Tunnels in existing systems are typically between firewall nodes which have statically allocated IP addresses. In such existing systems, the statically allocated IP address of the firewall is the address of a tunnel end point within the firewall. Existing systems fail to provide a tunnel which can perform authorization based for an entity which must dynamically allocate its network layer address. This is especially problematic for a user wishing to establish a tunnel in a mobile computing environment, and who requests a dynamically allocated IP address from an Internet Service Provider (ISP).

Because existing virtual private networks are based on network layer within network layer encapsulation, they are generally only capable of providing connectionless datagram type services. Because datagram type services do not guarantee delivery of packets, existing tunnels can only easily employ encryption methods over the data contained within each transmitted packet. Encryption based on the contents of multiple packets is desirable, such as cipher block chaining or stream ciphering over multiple packets. For example, encrypted data would advantageously be formed based not only on the contents of the present packet data being encrypted, but also based on some attribute of the connection or session history between the communicating stations. Examples of encryption algorithms and keyed encryption are disclosed in many textbooks, for example "Applied Cryptography - Protocols, Algorithms, and Source Code in C", by Bruce Schneier, published by John Wiley and Sons, New York, New York, copyright 1994.

Thus there is required a new pseudo network adapter providing a virtual private network having a dynamically determined end point to support a user in a mobile computing environment. The new pseudo network adapter should appear to the communications protocol stack of the node as an interface to an actual physical device. The new pseudo network adapter should support guaranteed, in-order delivery of frames over a tunnel to conveniently support cipher block chaining mode or stream cipher encryption over multiple packets.

SUMMARY OF THE INVENTION

A new pseudo network adapter is disclosed providing a virtual private network. The new system includes an interface for capturing packets from a local communications protocol stack for transmission on the virtual private network. The interface appears to the local communications stack as a network adapter device driver for a network adapter.

The invention, in its broad form, includes a pseudo network adapter as recited in claim 1, providing a virtual network and a method therefor as recited in claim 9.

The system as described hereinafter further includes a Dynamic Host Configuration Protocol (DHCP) server emulator, and an Address Resolution Protocol (ARP) server emulator. The new system indicates to the local communications protocol stack that nodes on a remote private network are reachable through a gateway that is in turn reachable through the pseudo network adapter. The new pseudo network adapter includes a transmit path for processing data packets from the local communications protocol stack for transmission through the pseudo network adapter. The transmit path includes an encryption engine for encrypting the data packets and an encapsulation engine for encapsulating the encrypted data packets into tunnel data frames. The pseudo network adapter passes the tunnel data frames back to the local communications protocol stack for transmission to a physical network adapter on a remote server node.

Preferably, as described hereinafter, the pseudo network adapter includes a digest value in a digest field in each of the tunnel data frames. A keyed hash function is a hash function which takes data and a shared cryptographic key as inputs, and outputs a digital signature referred to as a digest. The value of the digest field is equal to an output of a keyed hash function applied to data consisting of the data packet encapsulated within the tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to the remote server node. In another aspect of the system, the pseudo network adapter processes an Ethernet header in each one of the captured data packets, including removing the Ethernet header.

The new pseudo network adapter further includes an interface into a transport layer of the local communications protocol stack for capturing received data packets from the remote server node, and a receive path for processing received data packets captured from the transport layer of the local communications protocol stack. The receive path includes a decapsulation engine, and a decryption engine, and passes the decrypted, decapsulated data packets back to the local communications protocol stack for delivery to a user.

Thus there is disclosed a new pseudo network adapter providing a virtual private network having dynamically determined end points to support users in a mobile computing environment. The new pseudo network adapter provides a system for capturing a fully formed frame prior to transmission. The new pseudo network adapter appears to the communications protocol stack of the station as an interface to an actual physical device. The new pseudo network adapter further includes encryption capabilities to conveniently provide secure communications between tunnel end points using stream mode encryption or cipher block chaining over multiple packets.

BRIEF DESCRIPTION OF THE DRAWINGS

A more detailed understanding of the invention may be had from the following description of a preferred embodiment, given by way of example and to be understood in conjunction with the accompanying drawing in which:

- Fig. 1 is a block diagram showing the Open Systems Interconnection (OSI) reference model;
- Fig. 2 is a block diagram showing the TCP/IP internet protocol suite;
- Fig. 3 is a block diagram showing an examplary embodiment of a tunnel connection across a public network between two tunnel servers;
- Fig. 4 is a flow chart showing an examplary embodiment of steps performed to establish a tunnel con-

nection;

- Fig. 5 is a flow chart showing an examplary embodiment of steps performed to perform session key management for a tunnel connection;
- Fig. 6 is a block diagram showing an examplary embodiment of a relay frame;
- Fig. 7 is a block diagram showing an examplary embodiment of a connection request frame;
- Fig. 8 is a block diagram showing an examplary embodiment of a connection response frame;
- Fig. 9 is a block diagram showing an examplary embodiment of a data frame;
- Fig. 10 is a block diagram showing an examplary embodiment of a close connection frame;
- Fig. 11 is a state diagram showing an examplary embodiment of a state machine forming a tunnel connection in a network node initiating a tunnel connection;
- Fig. 12 is a state diagram showing an examplary embodiment of a state machine forming a tunnel connection in a server computer;
- Fig. 13 is a state diagram showing an examplary embodiment of a state machine forming a tunnel connection in a relay node;
- Fig. 14 is a block diagram showing an examplary embodiment of a tunnel connection between a client computer (tunnel client) and a server computer (tunnel server);
- Fig. 15 is a block diagram showing an examplary embodiment of a pseudo network adapter;
- Fig. 16 is a block diagram showing an examplary embodiment of a pseudo network adapter;
- Fig. 17 is a flow chart showing steps performed by an examplary embodiment of a pseudo network adapter during packet transmission;
- Fig. 18 is a flow chart showing steps performed by an examplary embodiment of a pseudo network adapter during packet receipt;
- Fig. 19 is a data flow diagram showing data flow in an examplary embodiment of a pseudo network adapter during packet transmission;
- Fig. 20 is a data flow diagram showing data flow in an examplary embodiment of a pseudo network adapter during packet receipt;
- Fig. 21 is a diagram showing the movement of encrypted and unencrypted data in an examplary embodiment of a system including a pseudo network adapter;
- Fig. 22 is a diagram showing the movement of encrypted and unencrypted data in an examplary embodiment of a system including a pseudo network adapter; and
- Fig. 23 is a flow chart showing steps initialization of an examplary embodiment of a system including a pseudo network adapter.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Now with reference to Fig. 1 there is described for purposes of explanation, communications based on the Open Systems Interconnection (OSI) reference model. In Fig. 1 there is shown communications 12 between a first protocol stack 10 and a second protocol stack 14. The first protocol stack 10 and second protocol stack 14 are implementations of the seven protocol layers (Application layer, Presentation layer, Session layer, Transport layer, Network layer, Data link layer, and Physical layer) of the OSI reference model. A protocol stack implementation is typically in some combination of software and hardware. Descriptions of the specific services provided by each protocol layer in the OSI reference model are found in many text books, for example "Computer Networks", Second Edition, by Andrew S. Tannenbaum, published by Prentice-Hall, Englewood Cliffs, New Jersey, copyright 1988.

As shown in Fig. 1, data 11 to be transmitted from a sending process 13 to a receiving process 15 is passed down through the protocol stack 10 of the sending process to the physical layer 9 for transmission on the data path 7 to the receiving process 15. As the data 11 is passed down through the protocol stack 10, each protocol layer prepends a header (and possibly also appends a trailer) portion to convey information used by that protocol layer. For example, the data link layer 16 of the sending process wraps the information received from the network layer 17 in a data link header 18 and a data link layer trailer 20 before the message is passed to the physical layer 9 for transmission on the actual transmission path 7.

Fig. 2 shows the TCP/IP protocol stack. Some protocol layers in the TCP/IP protocol stack correspond with layers in the OSI protocol stack shown in Fig. 1. The detailed services and header formats of each layer in the TCP/IP protocol stack are described in many texts, for example "Internetworking with TCP/IP, Vol. 1: Principles, Protocols, and Architecture", Second Edition, by Douglas E. Comer, published by Prentice-Hall, Englewood Cliffs, New Jersey, copyright 1991. The Transport Control Protocol (TCP) 22 corresponds to the Transport layer in the OSI reference model. The TCP protocol 22 provides a connection-oriented, end to end transport service with guaranteed, in-sequence packet delivery. In this way the TCP protocol 22 provides a reliable, transport layer connection.

The IP protocol 26 corresponds to the Network layer of the OSI reference model. The IP protocol 26 provides no guarantee of packet delivery to the upper layers. The hardware link level and access protocols 32 correspond to the Data link and Physical layers of the OSI reference model.

The Address Resolution Protocol (ARP) 28 is used to map IP layer addresses (referred to as "IP addresses") to addresses used by the hardware link level and access protocols 32 (referred to as "physical addresses" or "MAC addresses"). The ARP protocol layer in each network station typically contains a table of mappings between IP addresses and physical addresses (referred to as the "ARP cache"). When a mapping between an IP address and the corresponding physical address is not known, the ARP protocol 28 issues a broadcast packet (an "ARP request" packet) on the local network. The ARP request indicates an IP address for which a physical address is being requested. The ARP protocols 28 in each station connected to the local network examine the ARP request, and if a station recognizes the IP address indicated by the ARP request, it issues a response (an "ARP response" or "ARP reply" packet) to the requesting station indicating the responder's physical address. The requesting ARP protocol reports the received physical address to the local IP layer which then uses it to send datagrams directly to the responding station. As an alternative to having each station respond only for its own IP address, an ARP server may be used to respond for a set of IP addresses it stores internally, thus potentially eliminating the requirement of a broadcast request. In that case, the ARP request can be sent directly to the ARP server for physical addresses corresponding to any IP address mappings stored within the ARP server.

At system start up, each station on a network must determine an IP address for each of its network interfaces before it can communicate using TCP/IP. For example, a station may need to contact a server to dynamically obtain an IP address for one or more of its network interfaces. The station may use what is referred to as the Dynamic Host Configuration Protocol (DHCP) to issue a request for an IP address to a DHCP server. For example, a DHCP module broadcasts a DHCP request packet at system start up requesting allocation of an IP address for an indicated network interface. Upon receiving the DHCP request packet, the DHCP server allocates an IP address to the requesting station for use with the indicated network interface. The requesting station then stores the IP address in the response from the server as the IP address to associate with that network interface when communicating using TCP/IP.

Fig. 3 shows an example configuration of network nodes for which the presently disclosed system is applicable. In the example of Fig. 3, the tunnel server A is an initiator of the tunnel connection. As shown in Fig. 3, the term "tunnel relay" node is used to refer to a station which forwards data packets between transport layer connections (for example TCP connections).

For example, in the present system a tunnel relay may be dynamically configured to forward packets between transport layer connection 1 and transport layer connection 2. The tunnel relay replaces the header information of packets received over transport layer connection 1 with header information indicating transport layer connection 2. The tunnel relay can then forward the packet to a firewall, which may be conveniently programmed to pass packets received over transport layer connection 2 into a private network on the other side of the firewall. In the present system, the tunnel relay dynamically forms transport layer connections when a tunnel connection is established. Accordingly the tunnel relay is capable of performing dynamic load balancing or providing redundant service for fault tolerance over one or more tunnel servers at the time the tunnel connection is established.

Fig. 3 shows a Tunnel Server A 46 in a private network N1 48, physically connected with a first Firewall 50. The first Firewall 50 separates the private network N1 48 from a public network 52, for example the Internet. The first Firewall 50 is for example physically connected with a Tunnel Relay B 54, which in turn is virtually connected through the public network 52 with a Tunnel Relay C. The connection between Tunnel Relay B and Tunnel Relay C may for example span multiple intervening forwarding nodes such as routers or gateways through the public network 52.

The Tunnel Relay C is physically connected with a second Firewall 58, which separates the public network 52 from a private network N2 60. The second Firewall 58 is physically connected with a Tunnel Server D 62 on the private network N2 60. During operation of the elements shown in Fig. 3, the Tunnel Server D 62 provides routing of IP packets between the tunnel connection with Tunnel Server A 46 and other stations on the private network N2 60. In this way the Tunnel Server D 62 acts as a router between the tunnel connection and the private network N2 60.

During operation of the elements shown in Fig. 3, the present system establishes a tunnel connection between the private network N1 48 and the private network N2 60. The embodiment of Fig. 3 thus eliminates the need for a dedicated physical cable or line to provide secure communications between the private network 48 and the private network 60. The tunnel connection between Tunnel Server A 46 and Tunnel Server D 62 is composed of reliable, pair-wise transport layer connections between Tunnel Server A 46 (node "A"), Tunnel Relay B 54 (node "B"), Tunnel Relay C 56 (node "C"), and Tunnel Server D 62 (node "D"). For example, such pair-wise connections may be individual transport layer connections between each node A and node B, node B and node C, and node C and node D. In an alternative embodiment, as will be described below, a tunnel connection may alternatively be formed between a standalone PC in a public network and a tunnel server within a private network.

Fig. 4 and Fig. 5 show an example embodiment of steps performed during establishment of the tunnel connection between Tunnel Server A 46 (node "A") and Tunnel Server D 62 (node "D") as shown in Fig. 3. Prior to the steps shown in Fig. 4, node A selects a tunnel path to reach node D. The tunnel path includes the tunnel end points and any intervening tunnel relays. The tunnel path is for example predetermined by a system administrator for node A. Each tunnel relay along the tunnel path is capable of finding a next node in the tunnel path, for example based on a provided next node name (or "next node arc"), using a predetermined naming convention and service, for example the Domain Name System (DNS) of the TCP/IP protocol suite.

During the steps shown in Fig. 4, each of the nodes A, B and C perform the following steps:

- resolve the node name of the next node in the tunnel path, for example as found in a tunnel relay frame;

- establish a reliable transport layer (TCP) connection to the next node in the tunnel path;

- forward the tunnel relay frame down the newly formed reliable transport layer connection to the next node in the tunnel path.

As shown for example in Fig. 4, at step 70 node A establishes a reliable transport layer connection with node B. At step 72 node A identifies the next downstream node to node B by sending node B a tunnel relay frame over the reliable transport layer connection between node A and node B. The tunnel relay frame contains a string buffer describing all the nodes along the tunnel path (see below description of an example tunnel relay frame format). At step 74, responsive to the tunnel relay frame from node A, node B searches the string buffer in the relay frame to determine if the string buffer includes node B's node name. If node B finds its node name in the string buffer, it looks at the next node name in the string buffer to find the node name of the next node in the tunnel path.

Node B establishes a reliable transport layer connection with the next node in the tunnel path, for example node C. Node B further forms an association between the reliable transport layer connection between Node A and Node B, over which the relay frame was received, and the newly formed reliable transport layer connection between Node B and Node C, and as a result forwards subsequent packets received over the reliable transport layer connection with Node A onto the reliable transport layer connection with Node C, and vice versa. At step 76 node B forwards the tunnel relay frame on the newly formed reliable transport layer connection to node C.

At step 78, responsive to the relay frame forwarded from node B, node C determines that the next node in the tunnel path is the last node in the tunnel path, and accordingly is a tunnel server. Node C may actively determine whether alternative tunnel servers are available to form the tunnel connection. Node C may select one of the alternative available tunnel servers to form the tunnel connection in order to provide load balancing or fault tolerance. As a result node C may form a transport layer connections with one of several available tunnel servers, for example a tunnel server that is relatively underutilized at the time the tunnel connection is established. In the example embodiment, node C establishes a reliable transport layer connection with the next node along the tunnel path, in this case node D.

Node C further forms an association between the reliable transport layer connection between Node B and Node C, over which the relay frame was received, and the newly formed reliable transport layer connection between Node C and Node D, and as a result forwards subsequent packets received over the reliable transport layer connection with Node B to the reliable transport layer connection with Node D, and vice versa. At step 80 node C forwards the relay frame to node D on the newly formed reliable transport layer connection.

Fig. 5 shows an example of tunnel end point authentication and sharing of key exchange material provided by the present system. The present system supports passing authentication data and key exchange material through the reliable transport layer connections previously established on the tunnel path. The following are provided by use of a key exchange/authentication REQUEST frame and a key exchange/authentication RESPONSE frame:

a) mutual authentication of both endpoints of the tunnel connection;

b) establishment of shared session encryption keys and key lifetimes for encrypting/authenticating subsequent data sent through the tunnel connection;

d) agreement on a shared set of cryptographic transforms to be applied to subsequent data; and

e) exchange of any other connection-specific data between the tunnel endpoints, for example strength and type of cipher to be used, any compression of the data to be used, etc. This data can also be used by clients of this protocol to qualify the nature of the authenticated connection.

At step 90 a key exchange/authentication request frame is forwarded over the reliable transport layer connections formed along the tunnel path from node A to node D. At step 92, a key exchange/authentication response frame is forwarded from node D back to node A through the reliable transport layer connections. The attributes exchanged using the steps shown in Fig. 5 may be used for the lifetime of the tunnel connection. In an alternative embodiment the steps shown in Fig. 5 are repeated as needed for the tunnel end points to exchange sufficient key exchange material to agree upon a set of session parameters for use during the tunnel connection such as cryptographic keys, key durations, and choice of encryption/decryption algorithms.

Further in the disclosed system, the names used for authentication and access control with regard to node A and node D need not be the network layer address or physical address of the nodes. For example, in an alternative embodiment where the initiating node sending the tunnel relay frame is a stand-alone PC located within a public network, the user's name may be used for authentication and/or access control purposes. This provides a significant improvement over existing systems which base authorization on predetermined IP addresses.

Fig. 6 shows the format of an example embodiment of a tunnel relay frame. The tunnel frame formats shown in Figs. 6, 7, 8 and 9 are encapsulated within the data portion of a transport layer (TCP) frame when transmitted. Alternatively, another equivalent, connection-oriented transport layer protocol having guaranteed, in-sequence frame delivery may be used. The example TCP frame format, including TCP header fields, is conventional and not shown.

The field 100 contains a length of the frame. The field 102 contains a type of the frame, for example a type of RELAY. The field 104 contains a tunnel protocol version number. The field 106 contains an index into a string buffer field 112 at which a name of the originating node is located, for example a DNS host name of the node initially issuing the relay frame (node A in Fig. 3). The fields following the origin index field 106 contain indexes into the string buffer 112 at which names of nodes along the tunnel path are located. For example each index may be the offset of a DNS host name within the string buffer 112. In this way the field 108 contains the index of the name of the first node in the tunnel path, for example node B (Fig. 3). The field 110 contains the index of the name of the second node in the tunnel path, etc. The field 112 contains a string of node names of nodes in the tunnel path.

During operation of the present system, the initiating node, for example node A as shown in Fig. 3, transmits a tunnel relay frame such as the tunnel relay frame shown in Fig. 6. Node A sends the tunnel relay frame to the first station along the tunnel path, for example node B (Fig. 3), over a previously established reliable transport layer connection. Node B searches the string buffer in the tunnel relay frame to find its node name, for example its DNS host name. Node B finds its node name in the string buffer indexed by path index 0, and then uses the contents of path index 1 110 to determine the location within the string buffer 112 of the node name of the next node along the tunnel path. Node B uses this node name to establish a reliable transport layer connection with the next node along the tunnel path. Node B then forwards the relay frame to the next node. This process continues until the end node of the tunnel route, for example tunnel server D 62 (Fig. 3) is reached.

Fig. 7 shows the format of an example embodiment of a key exchange/key authentication request frame. The field 120 contains a length of the frame. The field 122 contains a type of the frame, for example a type of REQUEST indicating a key exchange/key authentication request frame. The field 124 contains a tunnel protocol version number. The field 126 contains an offset of the name of the entity initiating the tunnel connection, for example the name of a user on the node originally issuing the request frame. This name and key exchange material in the request frame are used by the receiving tunnel end point to authenticate the key exchange/authentication REQUEST. The name of the entity initiating the tunnel connection is also use to authorize any subsequent tunnel connection, based on predetermined security policies of the system. The field 128 contains an offset into the frame of the node name of the destination node, for example the end node of the tunnel shown as node D 62 in Fig. 3.

The field 130 contains an offset into the frame at which key exchange data as is stored, for example within the string buffer field 138. The key exchange data for example includes key exchange material used to determine a shared set of encryption parameters for the life of the tunnel connection such as cryptographic keys and any validity times associated with those keys. The key exchange data, as well as the field 132, further include information regarding any shared set of cryptographic transforms to be used and any other connection-specific parameters, such as strength and type of cipher to be used, type of compression of the data to be used, etc. The field 134 contains flags, for example indicating further information about the frame. The field 136 contains client data used in the tunnel end points to configure the local routing tables so that packets for nodes reachable through the virtual private network are sent through the pseudo network adapters. In an example embodiment, the string buffer 138 is encrypted using a public encryption key of the receiving tunnel end point.

During operation of the present system, one of the end nodes of the tunnel sends a key exchange/authentication REQUEST frame as shown in Fig. 7 to the other end node of the tunnel in order to perform key exchange and authentication as described in step 90 of Fig. 5.

Fig. 8 shows the format of an example embodiment of a key exchange/key authentication response frame, referred to as a connection RESPONSE frame. The field 150 contains a length of the frame. The field 152 contains a type of the frame, for example a type of connection RESPONSE indicating a key exchange/key authentication request frame. The field 154 contains a tunnel protocol version number.

The field 156 contains an offset into the frame at which key exchange data as is stored, for example within the string buffer field 163. The key exchange data for example includes key exchange material to be used for encryption/decryption over the life of the tunnel connection and any validity times associated with that key exchange material. The key exchange data, as well as the field 158, further includes information regarding any shared set of cryptographic transforms to be applied to subsequent data and any other connection-specific parameters, such as strength and type of cipher to be used, any compression of the data to be used, etc. The field 160 contains flags, for example indicating other information about the frame. The client data field 162 contains data used by the pseudo network adapters in the tunnel end points to configure the local routing tables so that packets for nodes in the virtual private network are sent through the pseudo network adapters. The string buffer includes key exchange material. The string buffer is for example encrypted using a public encryption key of the receiving tunnel end point, in the this case the initiator of the tunnel connection.

During operation of the present system, one of the end nodes of the tunnel sends a key exchange/authentication RESPONSE frame as shown in Fig. 7 to the other end node of the tunnel in order to perform key exchange and authentication as described in step 92 of Fig. 5.

Fig. 9 shows the format of an example embodiment of an tunnel data frame used to communicate through a tunnel connection. Fig. 9 shows how an IP datagram may be encapsulated within a tunnel frame by the present system for secure communications through a virtual private network. The field 170 contains a length of the frame. The field 172 contains a type of the frame, for example a type of DATA indicating a tunnel data frame. The field 174 contains a tunnel protocol version number.

The fields 176, 178 and 182 contain information regarding the encapsulated datagram. The field 180 contains flags indicating information regarding the frame. The field 184 contains a value indicating the length of the optional padding 189 at the end of the frame. The frame format allows for optional padding in the event that the amount of data in the frame needs to be padded to an even block boundary for the purpose of being encrypted using a block cipher. The field 186 contains a value indicating the length of the digest field 187.

The data frame format includes a digital signature generated by the transmitting tunnel end point referred to as a "digest". The value of the digest ensures data integrity, for example by detecting invalid frames and replays of previously transmitted valid frames. The digest is the output of a conventional keyed cryptographic hash function applied to both the encapsulated datagram 190 and a monotonically increasing sequence number. The resulting hash output is passed as the value of the digest field 187. The sequence number is not included in the data frame. In the example embodiment, the sequence number is a counter maintained by the transmitter (for example node A in Fig. 3) of all data frames sent to the receiving node (for example node D in Fig. 3) since establishment of the tunnel connection.

In order to determine if the data frame is invalid or a duplicate, the receiving node decrypts the encapsulated datagram 190, and applies the keyed cryptographic hash function (agreed to by the tunnel end nodes during the steps shown in Fig. 5) to both the decrypted encapsulated datagram and the value of a counter indicating the number of data frames received from the transmitter since establishment of the tunnel connection. For example the keyed hash function is applied to the datagram concatenated to the counter value. If the resulting hash output matches the value of the digest field 187, then the encapsulated datagram 190 was received correctly and is not a duplicate. If the hash output does not match the value of the digest field 187, then the integrity check fails, and the tunnel connection is closed. The field 188 contains an encrypted network layer datagram, for example an encrypted IP datagram.

The encapsulated datagram may be encrypted using various encryption techniques. An example embodiment of the present system advantageously encrypts the datagram 190 using either a stream cipher or cipher block chaining encryption over all data transmitted during the life of the tunnel connection. This is enabled by the reliable nature of the transport layer connections within the tunnel connection. The specific type of encryption and any connection specific symmetric encryption keys used is determined using the steps shown in Fig. 5. The fields in the tunnel data frame other than the encapsulated datagram 188 are referred to as the tunnel data frame header fields.

Fig. 10 is a block diagram showing an example embodiment of a "close connection" frame. The field 190 contains the length of the frame. The field 191 contains a frame type, for example having a value equal to CLOSE. Field 192 contains a value equal to the current protocol version number of the tunnel protocol. The field 193 contains a status code indicating the reason the tunnel connection is being closed.

During operation of the present system, when end point of a tunnel connection determines that the tunnel connection should be closed, a close connection frame as shown in Fig. 10 is transmitted to the other end point of the tunnel connection. When a close connection close frame is received, the receiver closes the tunnel connection and no further data will be transmitted or received through the tunnel connection.

Fig. 11 is a state diagram showing an example embodiment of forming a tunnel connection in a node initiating a tunnel connection. In Fig. 11, Fig. 12, and Fig. 13, states are indicated by ovals and actions or events are indicated by rectangles. For example the tunnel server node A as shown in Fig. 3 may act as a tunnel connection initiator when establishing a tunnel connection with the tunnel server node D. Similarly the client system 247 in Fig. 14 may act as a tunnel connection initiator when establishing a tunnel connection with the tunnel server. The tunnel initiator begins in an idle state 194. Responsive to an input from a user indicating that a tunnel connection should be established, the tunnel initiator transitions from the idle state 194 to a TCP Open state 195. In the TCP Open state 195, the tunnel initiator establishes a reliable transport layer connection with a first node along the tunnel path. For example, the tunnel initiator opens a socket interface associated with a TCP connection to the first node along the tunnel path. In Fig. 3 node A opens a socket interface associated with a TCP connection with node B.

Following establishment of the reliable transport layer connection in the TCP Open state 195, the tunnel initiator enters a Send Relay state 197. In the Send Relay state 197, the tunnel initiator transmits a relay frame at 198 over the reliable transport layer connection. Following transmission of the relay frame, the tunnel initiator enters the connect state 199. If during transmission of the relay frame there is a transmission error, the tunnel initiator enters the Network Error state 215 followed by the Dying state 208. In the Dying state 208, the tunnel initiator disconnects the reliable transport layer connection formed in the TCP Open state 195, for example by disconnecting a TCP connection with Node B. Following the disconnection at 209, the tunnel initiator enters the Dead state 210. The tunnel initiator subsequently transitions back to the Idle state 194 at a point in time predetermined by system security configuration parameters.

In the Connect state 199, the tunnel initiator sends a key exchange/authentication REQUEST frame at 200 to the tunnel server. Following transmission of the key exchange/authentication REQUEST frame 200, the tunnel initiator enters the Response Wait state 201. The tunnel initiator remains in the Response Wait state 201 until it receives a key exchange/authentication RESPONSE frame 202 from the tunnel server. After the key exchange/authentication RESPONSE frame is received at 202, the tunnel initiator enters the Authorized state 203, in which it may send or receive tunnel data frames. Upon receipt of a CLOSE connection frame at 216 in the Authorized state 203, the tunnel initiator transitions to the Dying state 208.

Upon expiration of a session encryption key at 211, the tunnel initiator enters the Reconnect state 212, and sends a CLOSE connection frame at 213 and disconnects the TCP connection with the first node along the tunnel path at 214. Subsequently the tunnel initiator enters the TCP Open state 195.

If during the authorized state 203, a local user issues an End Session command at 204, or there is a detection of an authentication or cryptography error in a received data frame at 205, the tunnel initiator enters the Close state 206. During the Close state 206 the tunnel initiator sends a CLOSE connection frame at 207 to the tunnel server. The tunnel initiator then enters the Dying state at 208.

Figure 12 is a state diagram showing the states within an example embodiment of a tunnel server, for example node D in Fig. 3 or tunnel server 253 in Fig. 14. The tunnel server begins in an Accept Wait state 217. In the Accept Wait state 217, the tunnel server receives a request for a reliable transport layer connection, for example a TCP connection reguest 218 from the last node in the tunnel path prior to the tunnel server, for example Node C in Fig. 3. In response to a TCP connection request 218 the tunnel server accepts the request and establishes a socket interface associated with the resulting TCP connection with Node C.

Upon establishment of the TCP connection with the last node in the tunnel path prior to the tunnel server, the tunnel server enters the Receive Relay state 219. In the Receive Relay state 219, the tunnel server waits to receive a relay frame at 220, at which time the tunnel server enters the Connect Wait state 221. If there is some sort of network error 234 during receipt of the relay frame at 219, the tunnel server enters the Dying state 230. During the Dying state 230 the tunnel server disconnects at 231 the transport layer connection with the last node in the tunnel path prior to the tunnel server. After disconnecting the connection, the tunnel server enters the Dead state 232.

In the Connect Wait state 221, the tunnel server waits for receipt of a key exchange/authentication REQUEST frame at 222. Following receipt of the key exchange/authentication REQUEST frame at 222, the tunnel server determines whether the requested tunnel connection is authorized at step 223. The determination of whether the tunnel connection is authorized is based on a name of the tunnel initiator, and the key exchange material within the key exchange/authentication REQUEST frame.

If the requested tunnel connection is authorized the tunnel server sends a key exchange/authentication RESPONSE frame at 224 back to the tunnel initiator. If the requested tunnel connection is not authorized, the tunnel server enters the Close state 228, in which it sends a close connection frame at 229 to the tunnel client. Following transmission of the CLOSE connection frame at 229, the tunnel server enters the Dying state 230.

If the requested tunnel connection is determined to be authorized at step 223, the tunnel server enters the Authorized state 225. In the Authorized state, the tunnel server transmits and receives tunnel data frames between itself and the tunnel initiator. If during the Authorized state 225, the tunnel server receives a CLOSE connection frame at 233, the tunnel server transitions to the Dying state 230. If during the authorized state 225, the tunnel server receives an end session command from a user at 226, then the tunnel server transitions to the Close state 228, and transmits a close connection frame at 229 to the tunnel initiator. If the tunnel server in the Authorized state 225 detects an integrity failure in a received packet, the tunnel server transitions to the Close state 228. In the close state 228 the tunnel server sends a CLOSE connection frame at 229 and subsequently enters the Dying state 230.

Fig. 13 is a state diagram showing an example embodiment of a state machine within a tunnel relay node. The tunnel relay node begins in an Accept Wait state 235. When a request is received to form a reliable transport layer connection at 236, a reliable transport layer connection is accepted with the requesting node. For example, a TCP connection is accepted between the relay node and the preceding node in the tunnel path.

The relay node then transitions to the Receive Relay state 237. During the Receive Relay state 237, the relay node receives a relay frame at 238. Following receipt of the relay frame at 238, the relay node determines what forwarding address should be used to forward frames received from the TCP connection established responsive to the TCP connect event 236. If the next node in the tunnel path is a tunnel server, the forwarding address may be selected at 239 so as to choose an underutilized tunnel server from a group of available tunnel servers or to choose an operational server where others are not operational.

Following determination of the forwarding address or addresses in step 239, the relay node enters the Forward Connect state 240. In the Forward Connect state 240, the relay node establishes a reliable transport layer connection with the node or nodes indicated by the forwarding address or addresses determined in step 239.

Following establishment of the new connection at event 241, the tunnel relay enters the Forward state 242. During the Forward state 242, the relay node forwards all frames between the connection established at 236 and those connections established at 241. Upon detection of a network error or receipt of a frame indicating a closure of the tunnel connection at 243, the tunnel relay enters the Dying state 244. Following the Dying state 244, the relay node disconnects any connections established at event 241. The relay node then enters the Dead state 246.

Fig. 14 shows an example embodiment of a virtual private network 249 formed by a pseudo network adapter 248 and a tunnel connection between a tunnel client 247 and a tunnel server 253 across a public network 251. The tunnel server 253 and tunnel client 247 are for example network stations including a CPU or microprocessor, memory, and various I/O devices. The tunnel server 253 is shown physically connected to a private LAN 256 including a Network Node 1 257 and a Network Node 2 258, through a physical network adapter 254. The tunnel server 253 is further shown physically connected with a firewall 252 which separates the private LAN 256 from the public network 251. The firewall 252 is physically connected with the public network 251. The tunnel server 253 is further shown including a pseudo network adapter 255. The client system 247 is shown including a physical network adapter 250 physically connected to the public network 251.

During operation of the elements shown in Fig. 14, nodes within the virtual private network 249 appear to the tunnel client 247 as if they were physically connected to the client system through the pseudo network adapter 248. Data transmissions between the tunnel client and any nodes that appear to be within the virtual private network are passed through the pseudo network adapter 248. Data transmissions between the tunnel client 247 and the tunnel server 253 are physically accomplished using a tunnel connection between the tunnel client 247 and the tunnel server 253.

Fig. 15 shows elements in an example embodiment of a pseudo network adapter such as the pseudo network adapter 248 in Fig. 14. In an example embodiment the elements shown in Fig. 15 are implemented as software executing on the tunnel client 247 as shown in Fig. 14. In Fig. 15 there is shown a pseudo network adapter 259 including a virtual adapter driver interface 263, an encapsulation engine 264, an encryption engine 265, a decapsulation engine 268, and a decryption engine 266. Further shown in the pseudo network adapter 259 are an ARP server emulator 270 and a Dynamic Host Configuration Protocol (DHCP) server emulator.

The pseudo network adapter 259 is shown interfaced to a TCP/IP protocol stack 260, through the virtual adapter driver interface 260. The TCP/IP protocol stack 260 is shown interfaced to other services in an operating system 261, as well as a physical network adapter's driver 262. The physical network adapter's driver 262 is for example a device driver which controls the operation of a physical network adapter such as physical network adapter 250 as shown in Fig. 14.

During operation of the elements shown in Fig. 15, the pseudo network adapter 259 registers with the network layer in the TCP/IP stack 260 that it is able to reach the IP addresses of nodes within the virtual private network 249 as shown in Fig. 14. For example, the pseudo network adapter on the client system registers that it can reach the pseudo network adapter on the server. Subsequently, a message from the tunnel client addressed to a node reachable through the virtual private network will be passed by the TCP/IP stack to the pseudo network adapter 259. The pseudo network adapter 259 then encrypts the message, and encapsulates the message into a tunnel data frame. The pseudo network adapter 259 then passes the tunnel data frame back to the TCP/IP protocol stack 260 to be sent through to the physical network adapter in the tunnel server. The tunnel server passes the received data frame to the pseudo network adapter in the server, which de-encapsulates and decrypts the message.

Fig. 16 shows a more detailed example embodiment of a pseudo network adapter 280. The pseudo network adapter 280 includes a virtual network adapter driver interface 288. The transmit path 290 includes an encryption engine 292, and an encapsulation engine 294. The encapsulation engine 294 is interfaced with a TCP/IP transmit interface 312 within a TCP/IP protocol stack, for example a socket interface associated with the first relay node in the tunnel path, or with the remote tunnel end point if the tunnel path includes no relays.

In the example embodiment of Fig. 16, the pseudo network adapter 280 appears to the TCP/IP protocol stack 282 as an Ethernet adapter. Accordingly, ethernet packets 286 for a destination addresses understood by the TCP/IP protocol stack to be reachable through the virtual private network are passed from the TCP/IP protocol stack 282 to the virtual network adapter interface 288 and through the transmit path 290. Similarly, ethernet packets 284 received through the pseudo network adapter 280 are passed from the receive path 296 to the virtual network adapter interface 288 and on to the TCP/IP protocol stack 282.

Further shown in the pseudo network adapter 280 of Fig. 16 is a receive path 296 having a decryption engine 298 interfaced to the virtual network adapter interface 288 and a decapsulation engine 300. The decapsulation engine 300 in turn is interfaced to a TCP/IP receive function 314 in the TCP/IP protocol stack 282, for example a socket interface associated with the first relay in the tunnel path, or with the remote tunnel end point if the tunnel path includes no relays. The pseudo network adapter 280 further includes an ARP server emulator 304 and a DHCP server emulator 306. ARP and DHCP request packets 302 are passed to the ARP server emulator 304 and DHCP server emulator 306 respectively. When a received packet is passed from the receive path 296 to the TCP/IP stack 282, a receive event must be indicated to the TCP/IP stack 282, for example through an interface such the Network Device Interface Specification (NDIS), defined by Microsoft™ Corporation.

Also in Fig. 16 is shown is an operating system 310 coupled with the TCP/IP protocol stack 282. The TCP/IP protocol stack 282 is generally considered to be a component part of the operating system. The operating system 310 in Fig. 16 is accordingly the remaining operating system functions and procedures outside the TCP/IP protocol stack 282. A physical network adapter 308 is further shown operated by the TCP/IP protocol stack 282.

During operation of the elements shown in Fig. 16, a user passes data for transmission to the TCP/IP protocol stack 282, and indicates the IP address of the node to which the message is to be transmitted, for example through a socket interface to the TCP layer. The TCP/IP protocol stack 282 then determines whether the destination node is reachable through the virtual private network. If the message is for a node that is reachable through the virtual private network, the TCP/IP protocol stack 282 an ethernet packet 286 corresponding to the message to the pseudo network adapter 280. The pseudo network adapter 280 then passes the ethernet packet 286 through the transmit path, in which the ethernet packet is encrypted and encapsulated into a tunnel data frame. The tunnel data frame is passed back into the TCP/IP protocol stack 282 through the TCP/IP transmit function 312 to be transmitted to the tunnel server through the tunnel connection. In an example embodiment, a digest value is calculated for the tunnel data frame before encryption within the transmit path within the pseudo network adapter.

Further during operation of the elements shown in Fig. 16, when the TCP/IP protocol stack 282 receives a packet from the remote endpoint of the TCP/IP tunnel connection, for example the tunnel server, the packet is passed to the pseudo network adapter 280 responsive to a TCP receive event. The pseudo network adapter 280 then decapsulates the packet by removing the tunnel header. The pseudo network adapter further decrypts the decapsulated data and passes it back to the TCP/IP protocol stack 282. The data passed from the pseudo network adapter 280 appears to the TCP/IP protocol stack 282 as an ethernet packet received from an actual physical device, and is the data it contains is passed on to the appropriate user by the TCP/IP protocol stack 282 based on information in the ethernet packet header provided by the pseudo network adapter.

Fig. 17 is a flow chart showing steps performed by an example embodiment of a pseudo network adapter during packet transmission, such as in the transmit path 290 of Fig. 14. The TCP/IP protocol stack determines that the destination node of a packet to be transmitted is reachable through the virtual LAN based on the destination IP address of the packet and a network layer routing table. At step 320 the packet is passed to the pseudo network adapter from the TCP/IP protocol stack. As a result, a send routine in the pseudo adapter is triggered for example in the virtual network adapter interface 288 of Fig. 16.

At step 322 the pseudo network adapter send routine processes the Ethernet header of the packet provided by the TCP/IP stack, and removes it. At step 324, the send routine determines whether the packet is an ARP request packet. If the packet is an ARP request packet for an IP address of a node on the virtual LAN, such as the pseudo network adapter of the tunnel server, then step 324 is followed by step 326. Otherwise, step 324 is followed by step 330.

At step 326, the ARP server emulator in the pseudo network adapter generates an ARP reply packet. For example, if the ARP request were for a physical address corresponding to the IP address of the pseudo network adapter on the tunnel server, the ARP reply would indicate a predetermined, reserved physical address to be associated with that IP address. At step 328 the pseudo network adapter passes the ARP response to the virtual network adapter interface. The virtual network adapter interface then indicates a received packet to the TCP/IP protocol stack, for example using an NDIS interface. The TCP/IP protocol stack then processes the ARP response as if it had been received over an actual physical network.

At step 330 the send routine determines whether the packet is a DHCP request packet requesting an IP address for the pseudo network adapter. If so, then step 330 is followed by step 332. Otherwise, step 330 is followed by step 334.

At step 334, the DHCP server emulator in the pseudo network adapter generates a DHCP response. The format of DHCP is generally described in the DHCP RFC. At step 328 the pseudo network adapter passes the DHCP response to the virtual network adapter interface, for example indicating an IP address received from the tunnel server in the client data field of the key exchange/authentication RESPONSE frame. The virtual network adapter interface then indicates a received packet to the TCP/IP protocol stack. The TCP/IP protocol stack then processes the DHCP response as if it had been received over an actual physical network.

At step 334 the pseudo network adapter encrypts the message using an encryption engine such that only the receiver is capable of decrypting and reading the message. At step 336 the pseudo network adapter encapsulates the encrypted message into a tunnel data frame. At step 338 the pseudo network adapter transmits the tunnel data frame through the tunnel connection using the TCP/IP protocol stack.

Fig. 18 is a flow chart showing steps performed by an example embodiment of a pseudo network adapter during packet receipt, such as in the receive path 296 of Fig. 14.

At step 350, the pseudo network adapter is notified that a packet has been received over the tunnel connection. At step 352 the pseudo network adapter decapsulates the received message by removing the header fields of the tunnel data frame. At step 354 the pseudo network adapter decrypts the decapsulated datagram from the tunnel data frame. At step 356, in an example embodiment, the pseudo network adapter forms an Ethernet packet from the decapsulated message. At step 358 the pseudo network adapter indicates that an Ethernet packet has been received to the TCP/IP protocol stack through the virtual network adapter interface. This causes the TCP/IP protocol stack to behave as if it had received an Ethernet packet from an actual Ethernet adapter.

Fig. 19 shows the data flow within the transmit path in an example embodiment of a pseudo network adapter. At step 1 370, an application submits data to be transmitted to the TCP protocol layer 372 within the TCP/IP protocol stack. The application uses a conventional socket interface to the TCP protocol layer 372 to pass the data, and indicates the destination IP address the data is to be transmitted to. The TCP protocol layer 372 then passes the data to the IP protocol layer 374 within the TCP/IP protocol stack. At step 2 376, the TCP/IP protocol stack refers to the routing table 378 to determine which network interface should be used to reach the destination IP address.

Because in the example the destination IP address is of a node reachable through the virtual private network, the IP layer 374 determines from the routing table 378 that the destination IP address is reachable through pseudo network adapter. Accordingly at step 3 380 the TCP/IP protocol stack passes a packet containing the data to the pseudo network adapter 382.

At step 4 384, the pseudo network adapter 382 encrypts the data packets and encapsulates them into tunnel data frames.

The pseudo network adapter 382 then passes the tunnel data frames packets back to the TCP protocol layer 372 within the TCP/IP protocol stack through a conventional socket interface to the tunnel connection with the first node in the tunnel path.

The TCP protocol layer 372 then forms a TCP layer packet for each tunnel data frame, having the tunnel data frame as its data. The TCP frames are passed to the IP layer 374. At step 5 386 the routing table 378 is again searched, and this time the destination IP address is the IP address associated with the physical network adapter on the tunnel server, and accordingly is determined to be reachable over the physical network adapter 390. Accordingly at step 6 388 the device driver 390 for the physical network adapter is called to pass the packets to the physical network adapter. At step 7 392 the physical network adapter transmits the data onto the physical network 394.

Fig. 20 is a data flow diagram showing data flow in an example embodiment of packet receipt involving a pseudo network adapter. At step 1 410 data arrives over the physical network 412 and is received by the physical network adapter and passed to the physical network driver 414. The physical network driver 414 passes the data at step 2 418 through the IP layer 420 and TCP layer 422 to the pseudo network adapter 426 at step 3 424, for example through a conventional socket interface. At step 4 428 the pseudo network adapter 426 decrypts and decapsulates the received data and passes it back to the IP layer of the TCP/IP protocol stack, for example through the TDI (Transport Layer Dependent Interface API) of the TCP/IP stack. The data is then passed through the TCP/IP protocol stack and to the user associated with the destination IP address in the decapsulated datagrams at step 5 430.

Fig. 21 shows data flow in an example embodiment of packet transmission involving a pseudo network adapter. Fig. 21 shows an example embodiment for use on a Microsoft™ Windows 95™ PC platform. In Fig. 21 a user application 450 passes unencrypted data to an interface into the TCP layer of the TCP/IP protocol, for example the WinSock API 452. The user indicates a destination IP address associated with a node reachable through a virtual private network accessible through the pseudo network adapter.

The TCP layer 454 passes the data to the IP layer 456, which in turn passes the data to the Network Device Interface Specification Media Access Control (NDIS MAC) interface 458. The pseudo network adapter 459 has previously registered with the routing layer (IP) that it is able to reach a gateway address associated with the destination IP address for the user data. Accordingly the IP layer uses the NDIS MAC layer interface to invoke the virtual device driver interface 460 to the pseudo network adapter 459. The pseudo network adapter 459 includes a virtual device driver interface 460, an ARP server emulator 462, and a DHCP server emulator 464.

In the example embodiment of Fig. 19, the pseudo network adapter 459 passes the data to a tunnel application program 466. The tunnel application program 466 encrypts the IP packet received from the IP layer and encapsulates it into a tunnel data frame. The tunnel application then passes the tunnel data frame including the encrypted data to the WinSock interface 452, indicating a destination IP address of the remote tunnel end point. The tunnel data frame is then passed through the TCP layer 454, IP layer 456, NDIS MAC layer interface 458, and physical layer 468, and transmitted on the network 470. Since the resulting packets do not contain a destination IP address which the pseudo network adapter has registered to convey, these packets will not be diverted to the pseudo network adapter.

Fig. 22 is a data flow diagram showing data flow in an example embodiment of packet transmission involving a pseudo network adapter. The embodiment shown in Fig. 22 is for use on a UNIX platform. In Fig. 20 a user application 472 passes unencrypted data to a socket interface to the TCP/IP protocol stack in the UNIX socket layer 474, indicating a destination IP address of a node reachable through the virtual private network.

The UNIX socket layer 474 passes the data through the TCP layer 476 and the IP layer 478. The pseudo network adapter 480 has previously registered with the routing layer (IP) that it is able to reach a gateway associated with the destination IP address for the user data. Accordingly the IP layer 478 invokes the virtual device driver interface 482 to the pseudo network adapter 480. The IP layer 478 passes the data to the pseudo network adapter 480. The pseudo network adapter 480 includes a virtual device driver interface 482, and a DHCP server emulator 484.

In the example embodiment of Fig. 22, the pseudo network adapter 480 passes IP datagrams to be transmitted to a UNIX Daemon 486 associated with the tunnel connection. The UNIX Daemon 486 encrypts the IP packet(s) received from the IP layer 478 and encapsulates them into tunnel data frames. The UNIX Daemon 486 then passes the tunnel data frames to the UNIX socket layer 474, through a socket associated with the tunnel connection. The tunnel data frames are then processed by the TCP layer 476, IP layer 478, data link layer 488, and physical layer 490 to be transmitted on the network 492. Since the resulting packets are not addressed to an IP address which the pseudo network adapter 480 has registered to convey, the packets will not be diverted to the pseudo network adapter 480.

Fig. 23 is a flow chart showing steps to initialize a example embodiment of a virtual private network. The steps shown in Fig. 23 are performed for example in the tunnel client 247 as shown in Fig. 14. At step 500 a tunnel application program executing in the tunnel client sends a tunnel relay frame to the tunnel server. At step 502 the tunnel application program sends a tunnel key exchange/authentication REQUEST frame to the tunnel server. The tunnel application in the tunnel server ignores the contents of the client data field in the tunnel key exchange/authentication REQUEST frame. The tunnel application in the tunnel server fills in the client data field in the tunnel key exchange/authentication RESPONSE frame with Dynamic Host Configuration Protocol (DHCP) information, for example including the following information in standard DHCP format:

1) IP Address for tunnel client Pseudo Network Adapter
2) IP Address for tunnel server Pseudo Network Adapter
3) Routes to nodes on the private network physically connected to the tunnel server which are to be reachable over the tunnel connection.

At step 504 the tunnel application receives a tunnel key exchange/authentication RESPONSE frame from the tunnel server. The client data field 508 in the tunnel connection response is made available to the pseudo network adapter in the tunnel client. The tunnel application in the tunnel client tells the TCP/IP stack that the pseudo network adapter in the tunnel client is active. The pseudo network adapter in the tunnel client is active and ready to be initialized at step 510.

The tunnel client system is configured such that it must obtain an IP address for the tunnel client pseudo network adapter dynamically. Therefore the TCP/IP stack in the tunnel client broadcasts a DHCP request packet through the pseudo network adapter. Accordingly, at step 512 the pseudo network adapter in the client receives a conventional DHCP request packet from the TCP/IP stack requesting a dynamically allocated IP address to associate with the pseudo network adapter. The pseudo network adapter passes the DHCP request packet to the DHCP server emulator within the pseudo network adapter, which forms a DHCP response based on the client data 508 received from the tunnel application. The DHCP response includes the IP address for the client pseudo adapter provided by the tunnel server in the client data. At step 514 the pseudo network adapter passes the DHCP response to the TCP/IP stack.

At step 520, the tunnel application modifies the routing tables within the tunnel client TCP/IP stack to indicate that the routes to the nodes attached to the private network to which the tunnel server is attached all are reachable only through the pseudo network adapter in the tunnel server. The IP address of the pseudo network adapter in the tunnel server provided in the client data is in this way specified as a gateway to the nodes on the private network to which the tunnel server is attached. In this way those remote nodes are viewed by the TCP/IP stack as being reachable via the virtual private network through the client pseudo network adapter.

At step 516 the pseudo network adapter in the tunnel client receives an ARP request for a physical address associated with the IP address of the pseudo network adapter in the tunnel server. The pseudo network adapter passes the ARP request to the ARP server emulator, which forms an ARP reply indicating a reserved physical address to be associated with the IP address of the pseudo network adapter in the tunnel server. At step 518 the pseudo network adapter passes the ARP response to the TCP/IP stack in the tunnel client. In response to the ARP response, the TCP/IP stack determines that packets addressed to any node on the virtual private network must be initially transmitted through the pseudo network adapter.

In an example embodiment the present system reserves two physical addresses to be associated with the pseudo network adapter in the client and the pseudo network adapter in the server respectively. These reserved physical addresses are used in responses to ARP requests passed through the pseudo network adapter for physical addresses corresponding to the IP addresses for the pseudo network adapter in the client and the pseudo network adapter in the server respectively. The reserved physical addresses should have a high likelihood of not being used in any actual network interface.

While the invention has been described with reference to specific example embodiments, the description is not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as other embodiments of the invention, will be apparent to persons skilled in the art upon reference to this description. Specifically, while various embodiments have been described using the TCP/IP protocol stack, the invention may advantageously be applied where other communications protocols are used. Also, while various flow charts have shown steps performed in an example order, various implementations may use altered orders of step in order to apply the invention. And further, while certain specific software and/or hardware platforms have been used in the description, the invention may be applied on other platforms with similar advantage. It is therefore contemplated that the appended claims will cover any such modifications or embodiments which fall within the scope of the invention.

Claims

1. A pseudo network adapter providing a virtual private network, comprising:

   an interface for capturing packets from a local communications protocol stack for transmission on said virtual private network, said interface appearing to said local communications protocol stack as a network adapter device driver for a network adapter connected to said virtual private network;

   a first server emulator, providing a first reply packet responsive to a first request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said first request packet requesting a network layer address for said pseudo network adapter, said first reply indicating a network layer address for said pseudo network adapter; and

   a second server emulator, providing a second reply packet responsive to an second request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said second request packet requesting a physical address corresponding to a network layer address of a second pseudo network adapter, said second pseudo network adapter located on a remote server node, said second reply indicating a predetermined, reserved physical address.

2. The pseudo network adapter of claim 1, further comprising a means for indicating to said local communications protocol stack that said predetermined, reserved physical address is reachable through said pseudo network adapter, wherein said means for indicating modifies a data structure in said local communications protocol stack indicating which nodes or networks are reachable through each network interface of the local system.

3. The pseudo network adapter of claim 1, further comprising a means for indicating to said local communications protocol stack that one or more nodes on a remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node.

4. The pseudo network adapter of claim 1, further comprising:

a transmit path for processing data packets captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network;
an encryption engine, within said transmit path, for encrypting said data packets;
an encapsulation engine, within said transmit path, for encapsulating said encrypted data packets into tunnel data frames; and
a means for passing said tunnel data frames back to said local communications protocol stack for transmission to a physical network adapter on said remote server node.

5. The pseudo network adapter of claim 4, wherein said transmit path further includes means for storing a digest value in a digest field in each of said tunnel data frames, said digest value equal to an output of a keyed hash function applied to said data packet encapsulated within said tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to said remote server node.

6. The pseudo network adapter of claim 4, wherein said transmit path further includes means for processing an Ethernet header in each one of said captured data packets, said processing of said Ethernet header including removing said Ethernet header.

7. The pseudo network adapter of claim 1, further comprising:

an interface into a transport layer of said local communications protocol stack for capturing received data packets from said remote server node.

8. The pseudo network adapter of claim 7, further comprising:

a receive path for processing received data packets captured by said interface into said transport layer of said local communications protocol stack for capturing received data packets from said remote server node;
an decapsulation engine, within said receive path, for decapsulating said received data packets by removing a tunnel frame header;
an decryption engine, within said receive path, for decrypting said received data packets; and
a means for passing said received data packets back to said local communications protocol stack for delivery to a user.

9. A method for providing a pseudo network adapter for a virtual private network, comprising the steps of:

capturing packets from a local communications protocol stack for transmission on said virtual private network, said capturing through an interface appearing to said local communications stack as a network adapter device driver for a network adapter connected to said virtual private network;
issuing a first reply packet responsive to a first request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said first request packet requesting a network layer address for said pseudo network adapter, said first reply indicating a network layer address for said pseudo network adapter; and
issuing a second reply packet responsive to a second request packet captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network, said second request packet requesting a physical address corresponding to a network layer address of a second pseudo network adapter, said second pseudo network adapter located on a remote server node, said ARP Reply indicating a predetermined, reserved physical address.

10. The method of claim 9, further comprising indicating to said local communications protocol stack that said predetermined, reserved physical address is reachable through said pseudo network adapter, wherein said step of indicating to said local communications protocol stack modifies a data structure in said local communications protocol stack indicating which nodes or networks are reachable through each network interface of the local system.

11. The method of claim 9, further comprising indicating to said local communications protocol stack that one or more nodes on a remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node, wherein said step of indicating to said local communications protocol stack that one or more nodes on said remote private network connected to said remote server node are reachable through a gateway node equal to said second pseudo network adapter on said remote server node modifies a network layer routing table in said local communications protocol stack.

12. The method of claim 9, further comprising:

processing data packets captured by said interface for capturing packets from said local communications protocol stack for transmission on said virtual private network in a transmit data path;

encrypting said data packets in an encryption engine, within said transmit path;

encapsulating said encrypted data packets into tunnel data frames by an encapsulation engine, within said transmit path; and

passing said tunnel data frames back to said local communications protocol stack for transmission to a physical network adapter on said remote server node, wherein said transmit path further includes storing a digest value in a digest field in each of said tunnel data frames, said digest value equal to an output of a keyed hash function applied to said data packet encapsulated within said tunnel data frame concatenated with a counter value equal to a total number of tunnel data frames previously transmitted to said remote server node.

13. The method of claim 12, wherein said transmit path further includes processing an Ethernet header in each one of said captured data packets, said processing of said Ethernet header including removing said Ethernet header.

14. The method of claim 9, further comprising capturing received data packets from said remote server node through an interface into a transport layer of said local communications protocol stack, further comprising:

processing received data packets captured by said interface into said transport layer of said local communications protocol stack for capturing received data packets from said remote server node in a receive path;

decapsulating said received data packets by removing a tunnel frame header in an decapsulation engine, within said receive path;

decrypting said received data packets in a decryption engine within said receive path; and

passing said received data frames packets back to said local communications protocol stack for delivery to a user.

15. The method of claim 9, wherein said network layer address for said pseudo network adapter and said predetermined, reserved physical address is communicated to said pseudo network adapter from said remote server node as client data in a connection response frame.

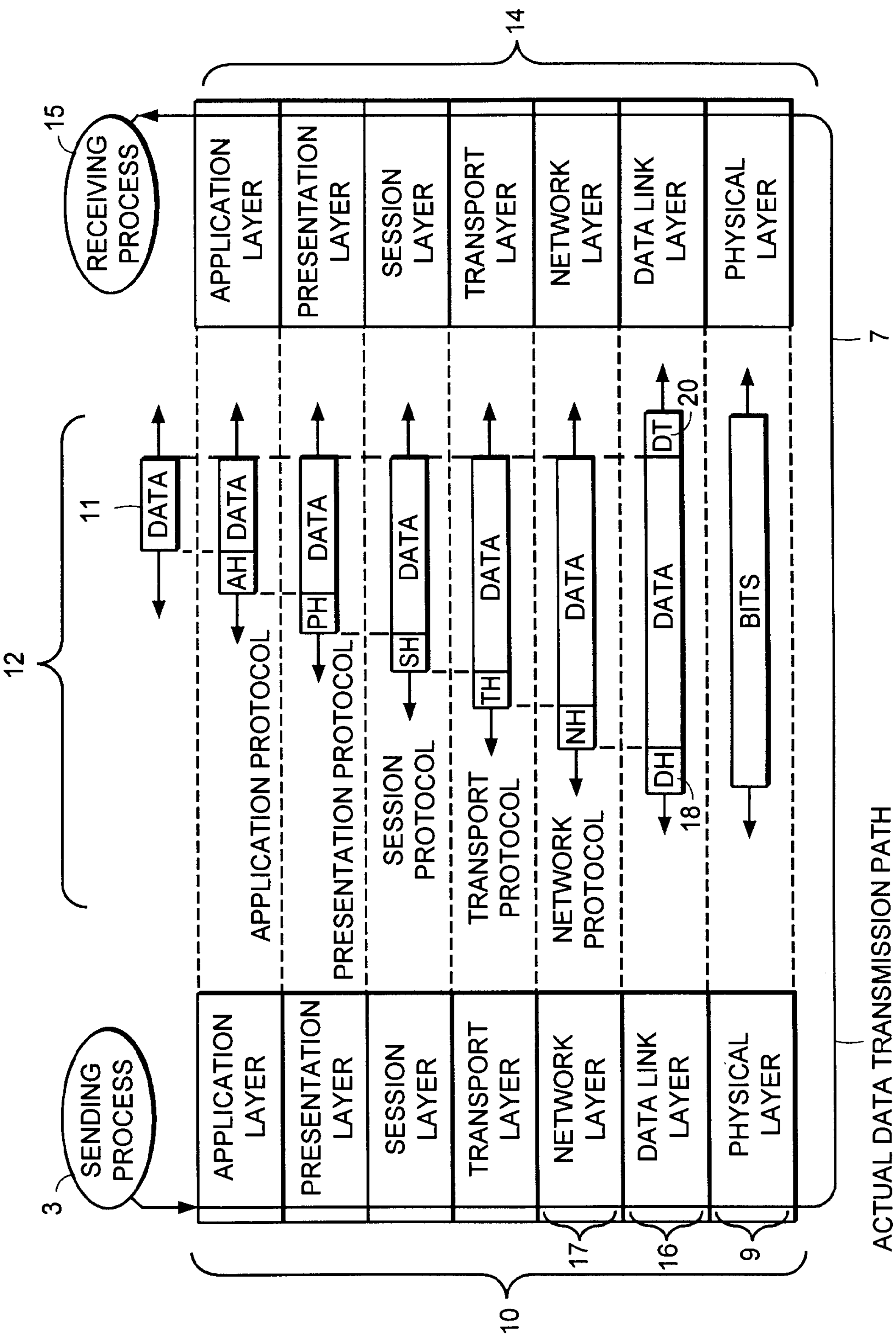

SENDING PROCESS
RECEIVING PROCESS
APPLICATION LAYER
PRESENTATION LAYER
SESSION LAYER
TRANSPORT LAYER
NETWORK LAYER
DATA LINK LAYER
PHYSICAL LAYER
APPLICATION PROTOCOL
PRESENTATION PROTOCOL
SESSION PROTOCOL
TRANSPORT PROTOCOL
NETWORK PROTOCOL
DATA
AH
PH
SH
TH
NH
DH
DT
BITS
ACTUAL DATA TRANSMISSION PATH
3
7
9
10
11
12
14
15
16
17
18
20

FIG. 1

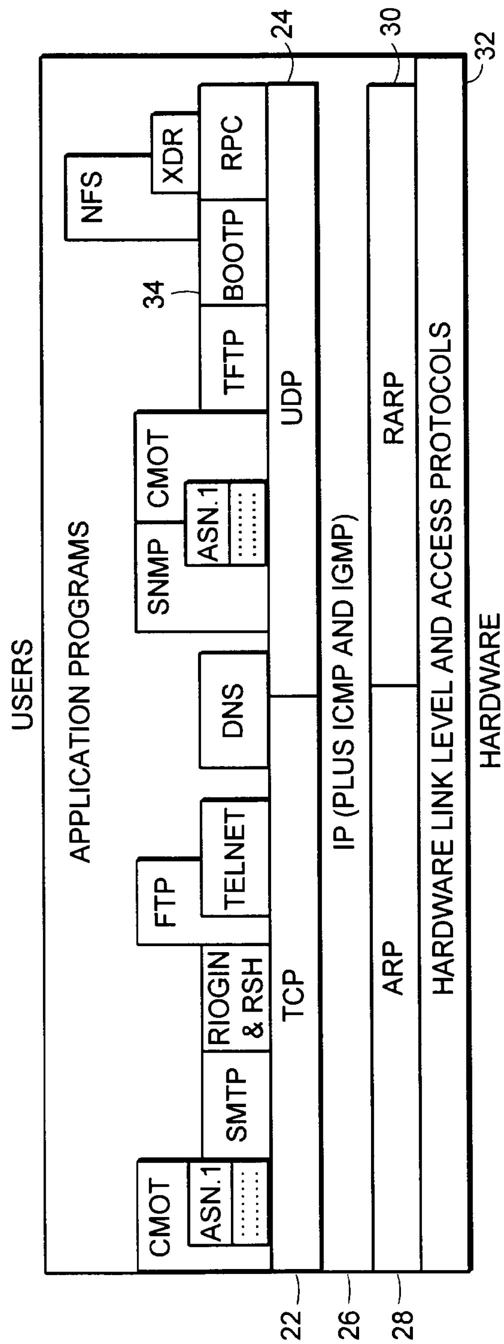

USERS
APPLICATION PROGRAMS
CMOT
ASN.1
SMTP
RIOGIN & RSH
FTP
TELNET
DNS
SNMP
ASN.1
CMOT
TFTP
BOOTP
NFS
XDR
RPC
TCP
UDP
IP (PLUS ICMP AND IGMP)
ARP
RARP
HARDWARE LINK LEVEL AND ACCESS PROTOCOLS
HARDWARE
22
26
28
24
30
32
34

FIG. 2

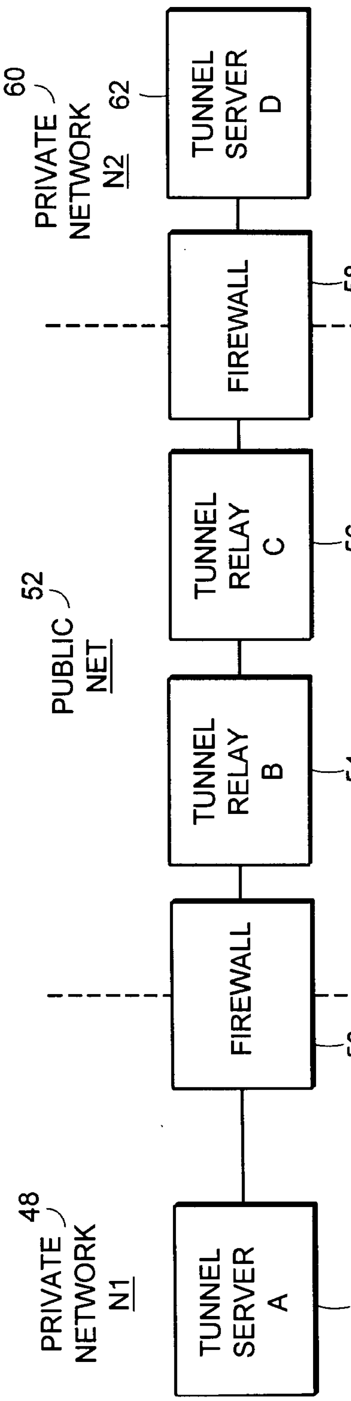

PRIVATE NETWORK N1
48
TUNNEL SERVER A
46
FIREWALL
50
PUBLIC NET
52
TUNNEL RELAY B
54
TUNNEL RELAY C
56
FIREWALL
58
PRIVATE NETWORK N2
60
62
TUNNEL SERVER D

FIG. 3

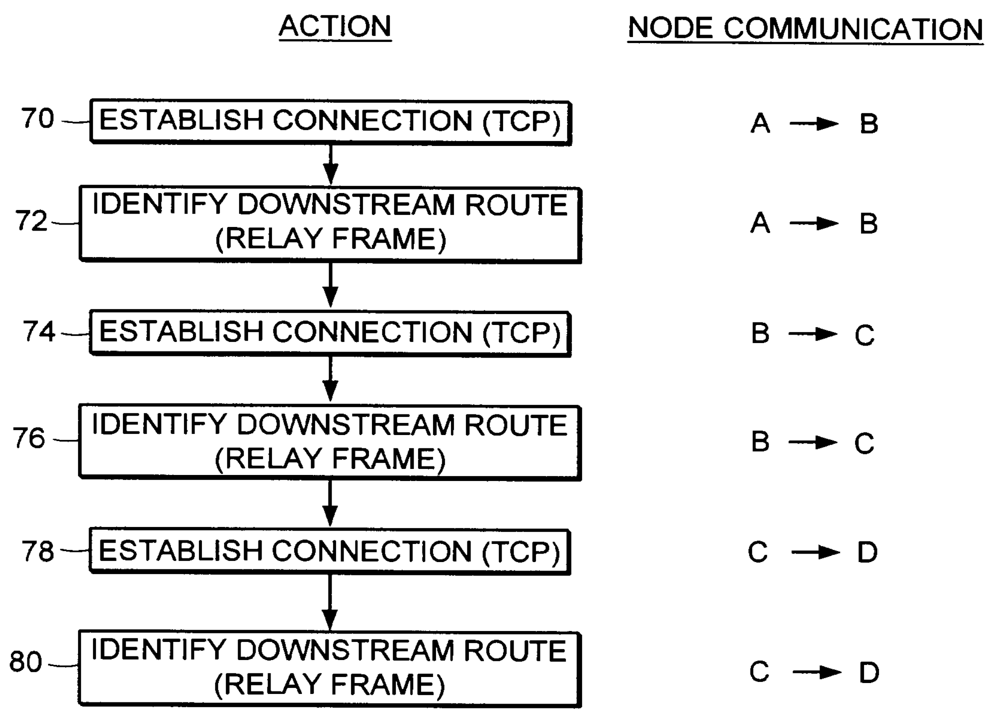

ACTION
NODE COMMUNICATION
70
ESTABLISH CONNECTION (TCP)
A → B
72
IDENTIFY DOWNSTREAM ROUTE (RELAY FRAME)
A → B
74
ESTABLISH CONNECTION (TCP)
B → C
76
IDENTIFY DOWNSTREAM ROUTE (RELAY FRAME)
B → C
78
ESTABLISH CONNECTION (TCP)
C → D
80
IDENTIFY DOWNSTREAM ROUTE (RELAY FRAME)
C → D

FIG. 4

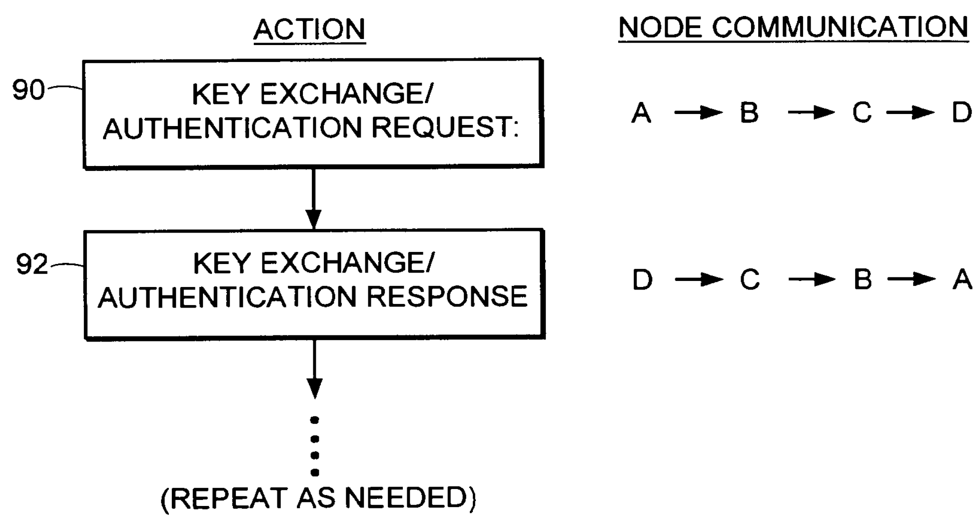

ACTION
NODE COMMUNICATION
90
KEY EXCHANGE/ AUTHENTICATION REQUEST:
A → B → C → D
92
KEY EXCHANGE/ AUTHENTICATION RESPONSE
D → C → B → A
(REPEAT AS NEEDED)

FIG. 5

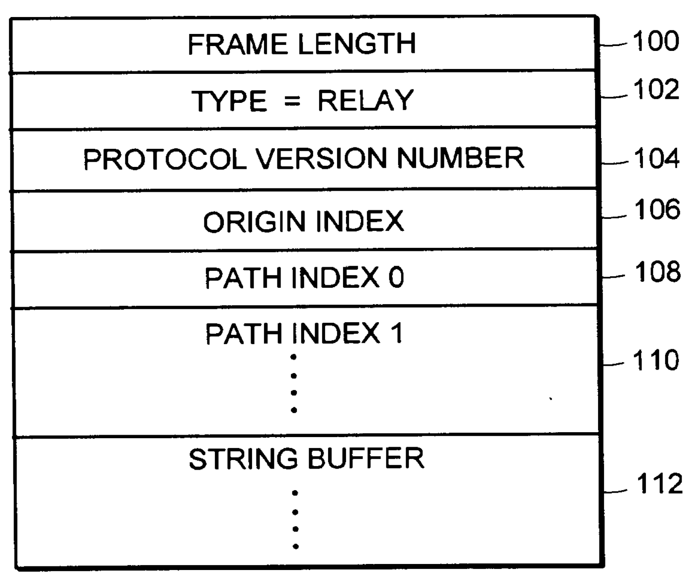

FRAME LENGTH
100
TYPE = RELAY
102
PROTOCOL VERSION NUMBER
104
ORIGIN INDEX
106
PATH INDEX 0
108
PATH INDEX 1
110
STRING BUFFER
112

FIG. 6

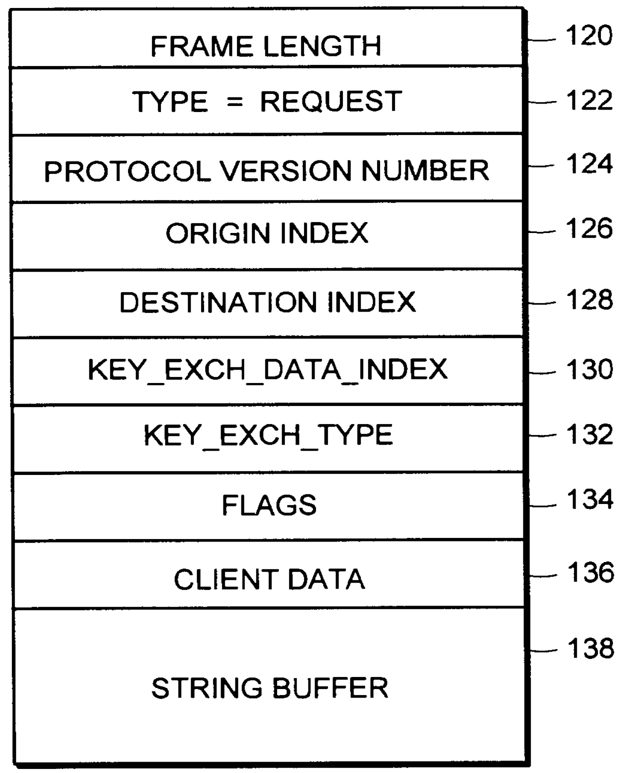

FRAME LENGTH
120
TYPE = REQUEST
122
PROTOCOL VERSION NUMBER
124
ORIGIN INDEX
126
DESTINATION INDEX
128
KEY_EXCH_DATA_INDEX
130
KEY_EXCH_TYPE
132
FLAGS
134
CLIENT DATA
136
STRING BUFFER
138

FIG. 7

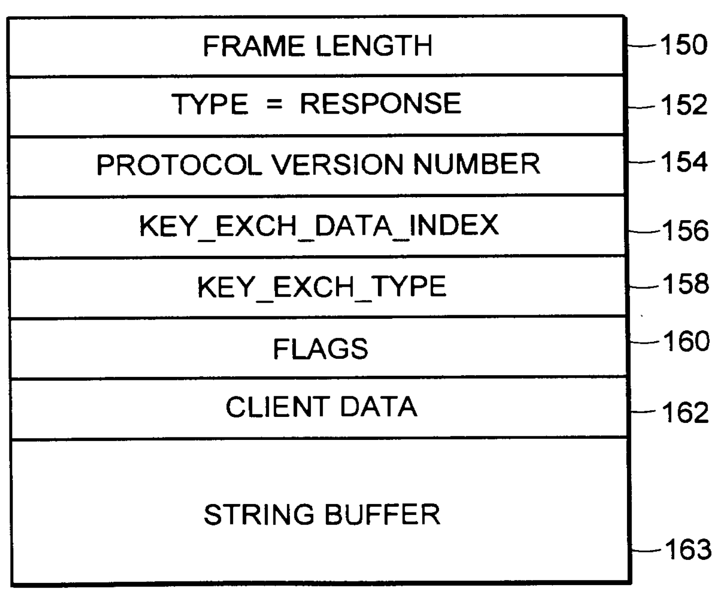

FRAME LENGTH
150
TYPE = RESPONSE
152
PROTOCOL VERSION NUMBER
154
KEY_EXCH_DATA_INDEX
156
KEY_EXCH_TYPE
158
FLAGS
160
CLIENT DATA
162
STRING BUFFER
163

FIG. 8

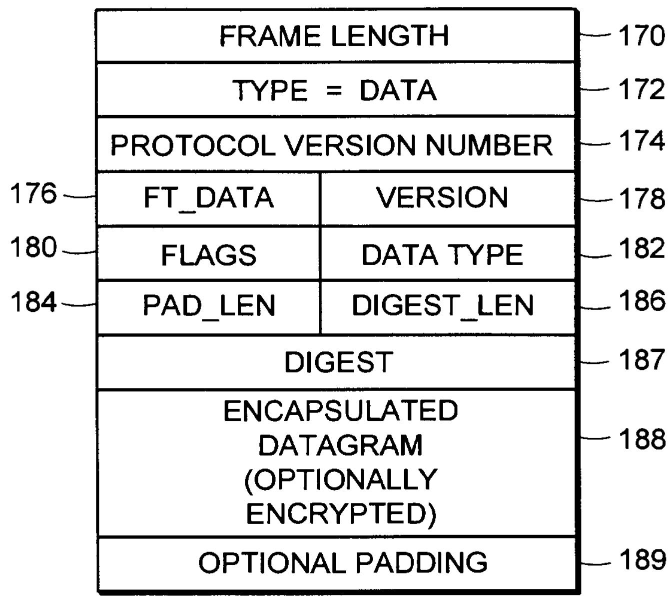

FRAME LENGTH
170
TYPE = DATA
172
PROTOCOL VERSION NUMBER
174
176
FT_DATA
VERSION
178
180
FLAGS
DATA TYPE
182
184
PAD_LEN
DIGEST_LEN
186
DIGEST
187
ENCAPSULATED DATAGRAM (OPTIONALLY ENCRYPTED)
188
OPTIONAL PADDING
189

FIG. 9

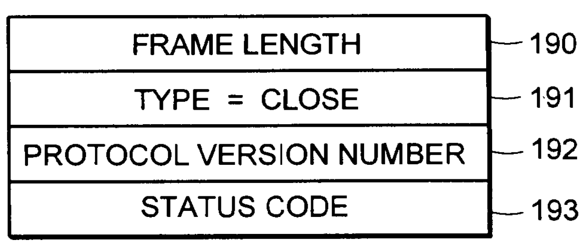

FRAME LENGTH
190
TYPE = CLOSE
191
PROTOCOL VERSION NUMBER
192
STATUS CODE
193

FIG. 10

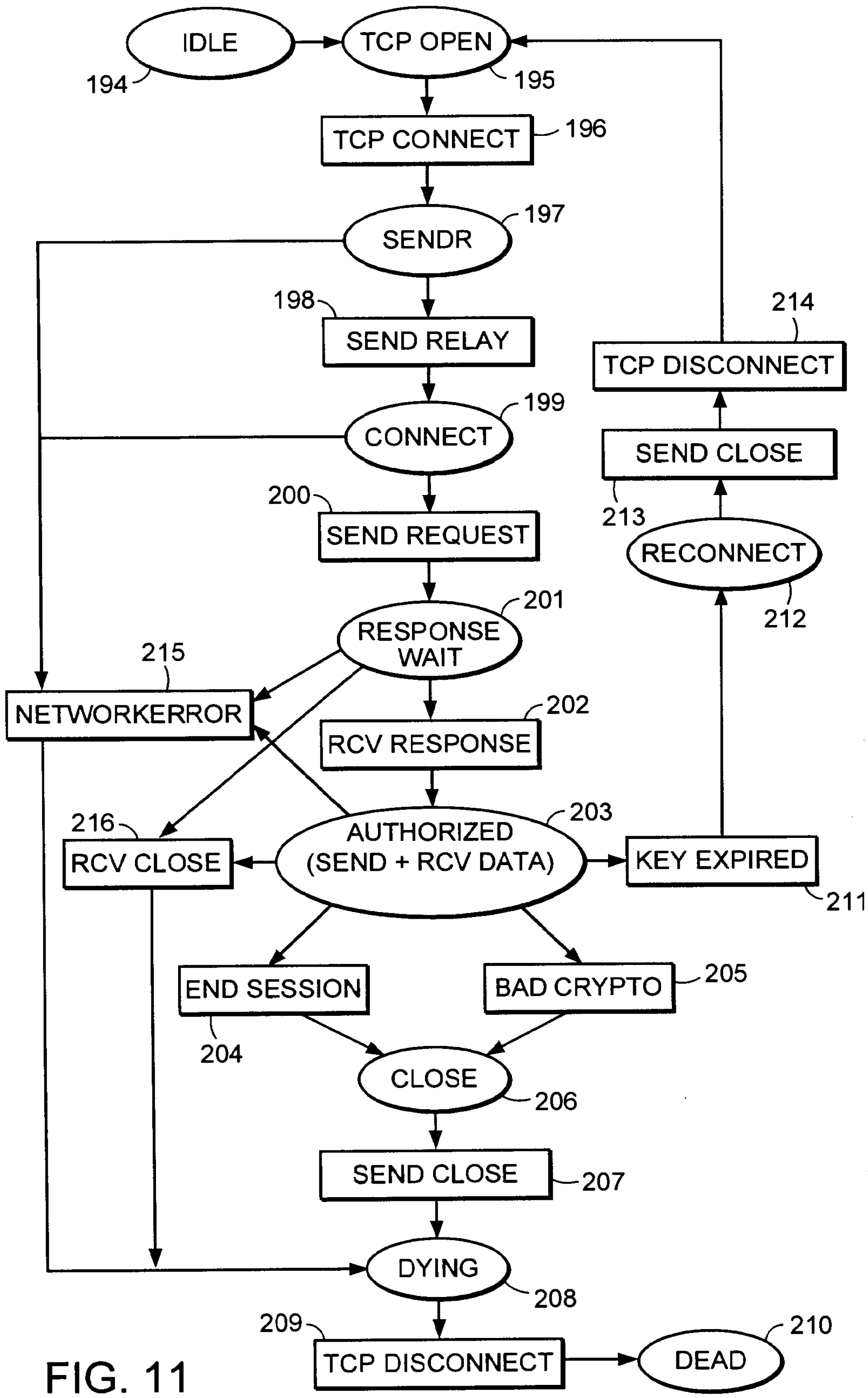

IDLE
194
TCP OPEN
195
TCP CONNECT
196
SENDR
197
198
SEND RELAY
CONNECT
199
200
SEND REQUEST
RESPONSE WAIT
201
215
NETWORKERROR
RCV RESPONSE
202
AUTHORIZED (SEND + RCV DATA)
203
216
RCV CLOSE
KEY EXPIRED
211
END SESSION
204
BAD CRYPTO
205
CLOSE
206
SEND CLOSE
207
DYING
208
209
TCP DISCONNECT
DEAD
210
214
TCP DISCONNECT
SEND CLOSE
213
RECONNECT
212

FIG. 11

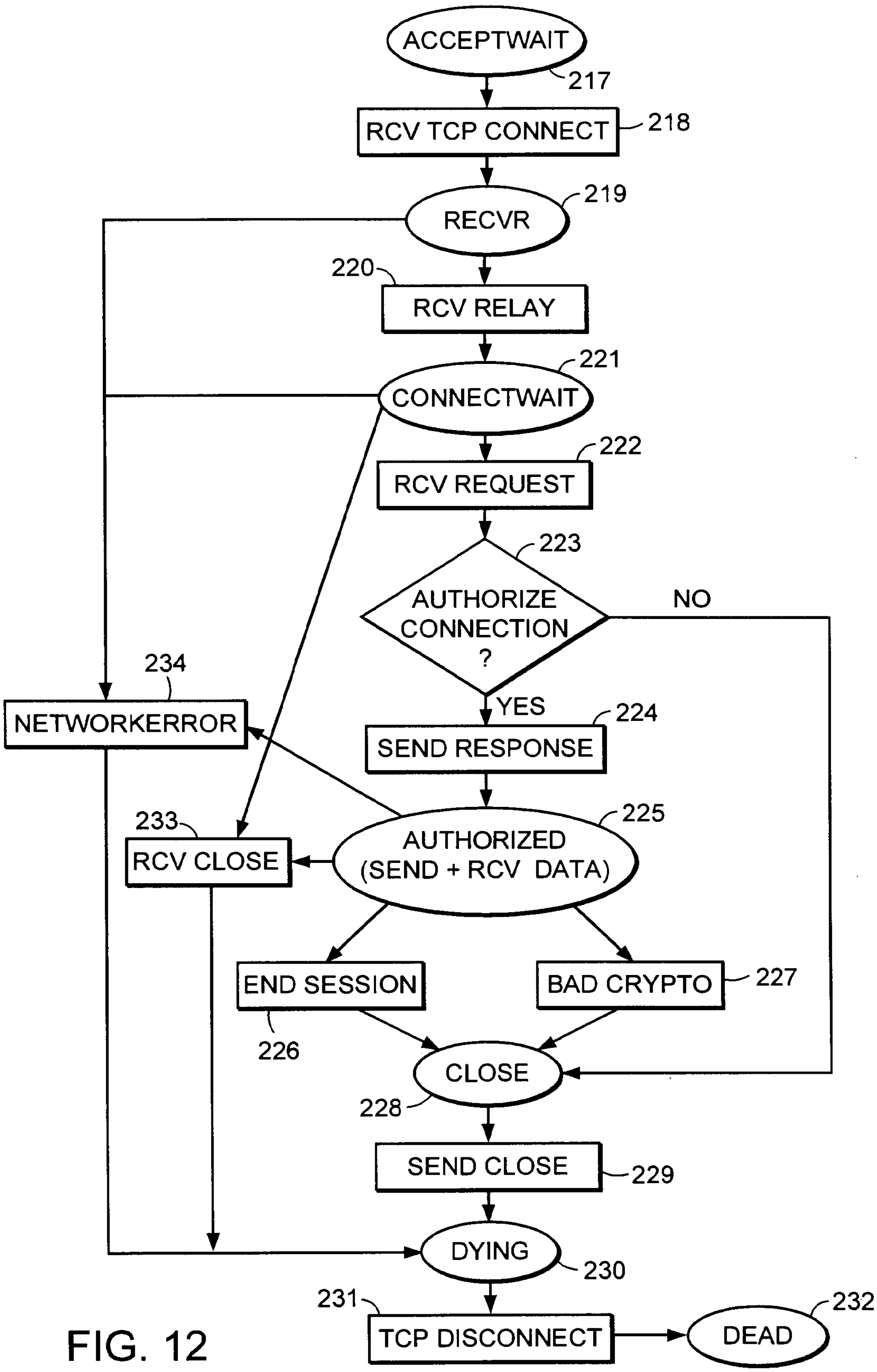

ACCEPTWAIT
217
RCV TCP CONNECT
218
RECVR
219
220
RCV RELAY
221
CONNECTWAIT
222
RCV REQUEST
223
AUTHORIZE
CONNECTION
?
NO
YES
224
SEND RESPONSE
234
NETWORKERROR
233
RCV CLOSE
225
AUTHORIZED
(SEND + RCV DATA)
END SESSION
226
BAD CRYPTO
227
CLOSE
228
SEND CLOSE
229
DYING
230
231
TCP DISCONNECT
DEAD
232

FIG. 12

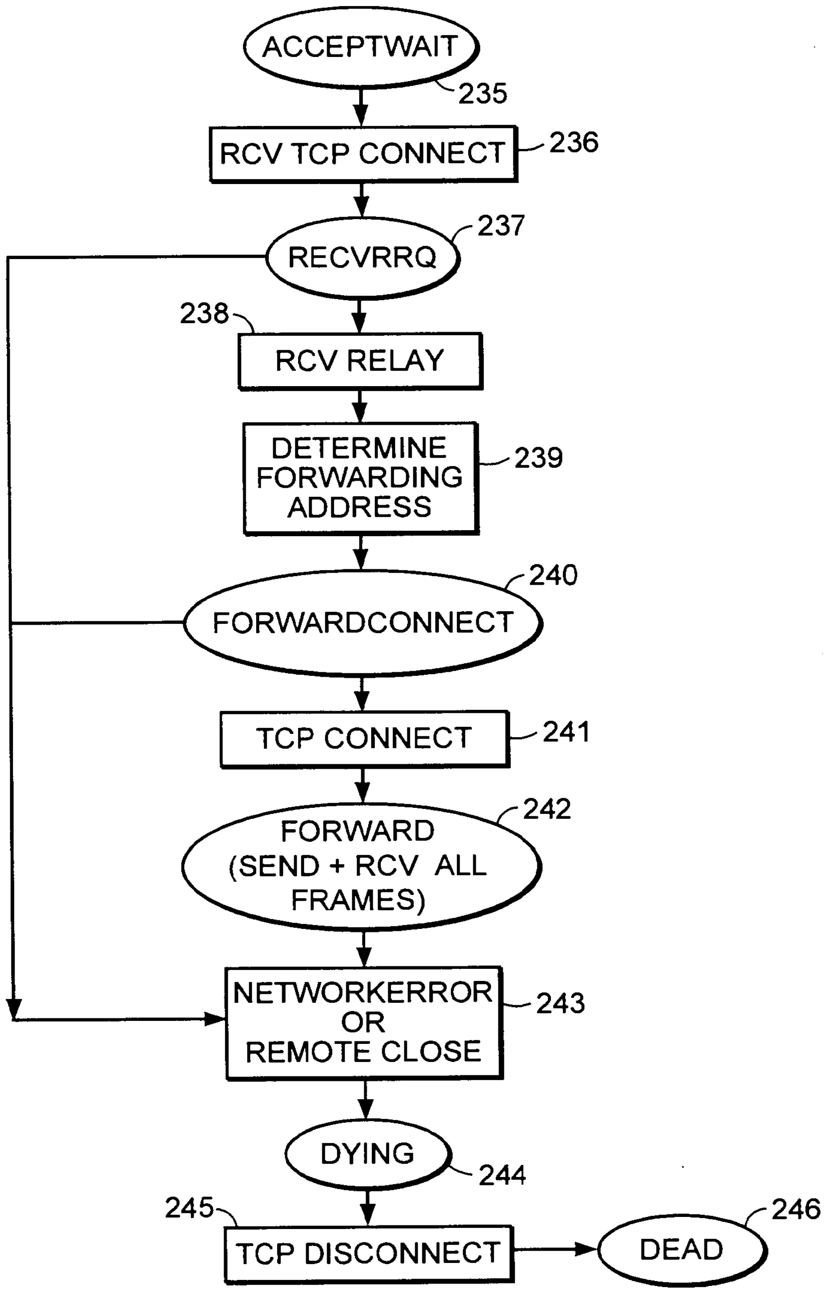

ACCEPTWAIT
235
RCV TCP CONNECT
236
RECVRRQ
237
238
RCV RELAY
DETERMINE FORWARDING ADDRESS
239
FORWARDCONNECT
240
TCP CONNECT
241
FORWARD (SEND + RCV ALL FRAMES)
242
NETWORKERROR OR REMOTE CLOSE
243
DYING
244
245
TCP DISCONNECT
DEAD
246

FIG. 13

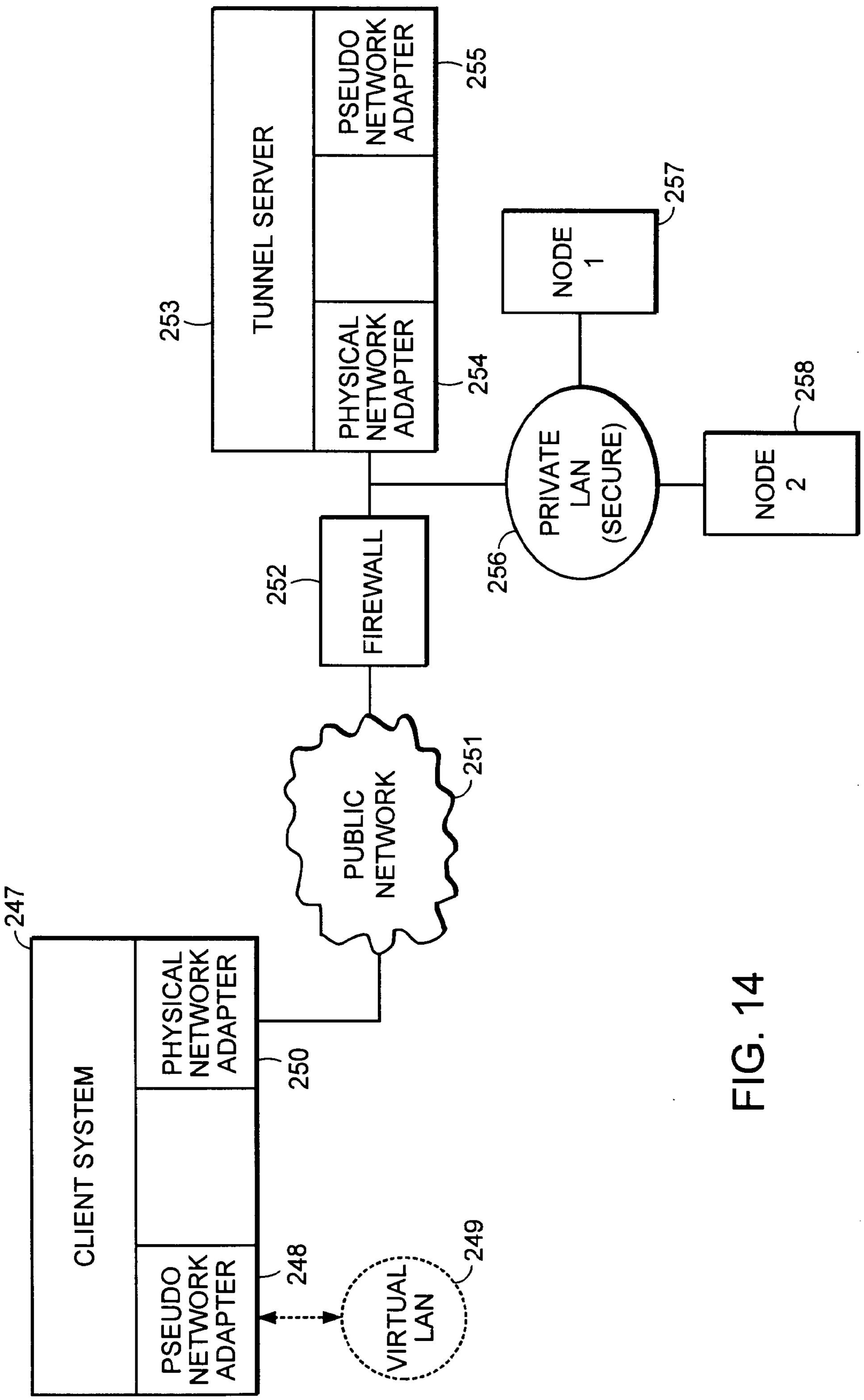

CLIENT SYSTEM
247
PSEUDO NETWORK ADAPTER
248
PHYSICAL NETWORK ADAPTER
250
VIRTUAL LAN
249
PUBLIC NETWORK
251
FIREWALL
252
TUNNEL SERVER
253
PHYSICAL NETWORK ADAPTER
254
PSEUDO NETWORK ADAPTER
255
PRIVATE LAN (SECURE)
256
NODE 1
257
NODE 2
258

FIG. 14

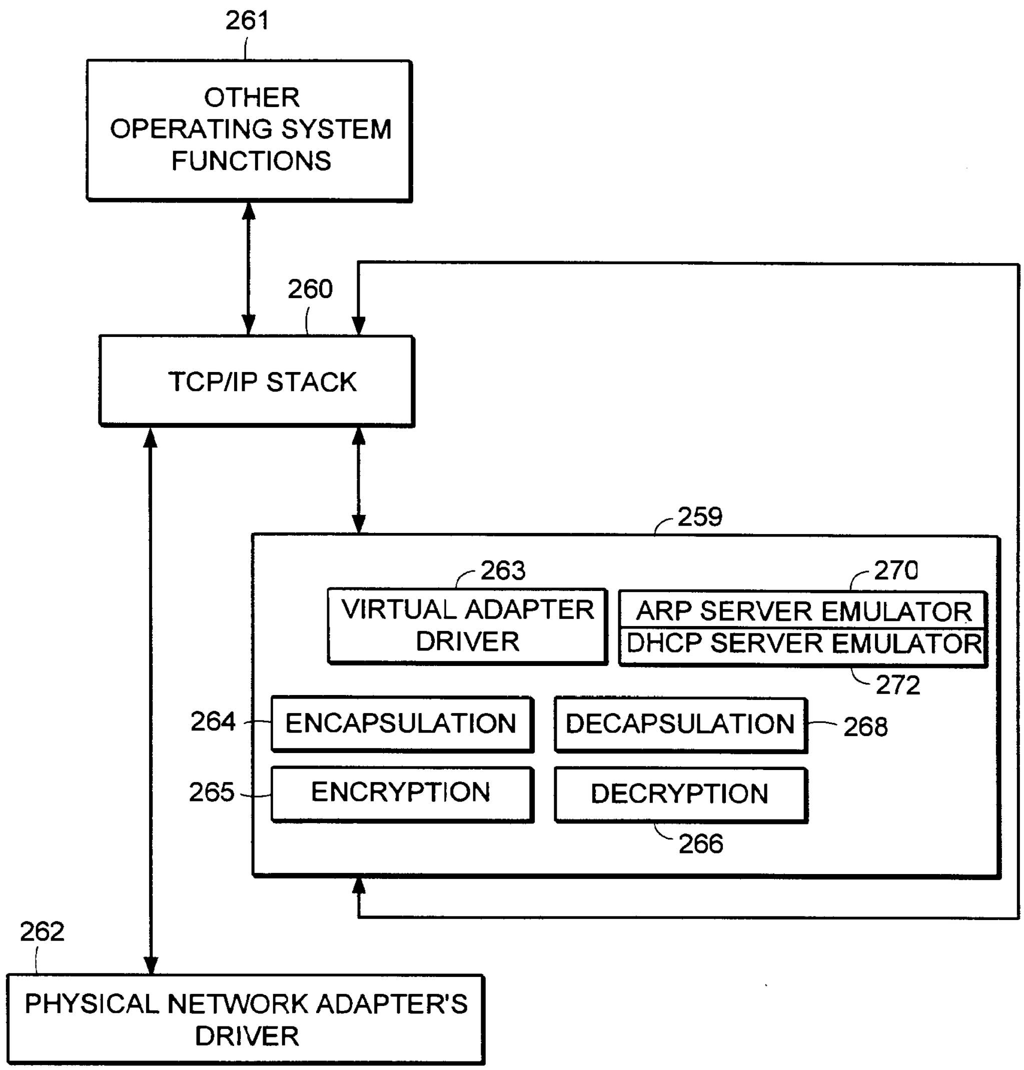

261
OTHER OPERATING SYSTEM FUNCTIONS
260
TCP/IP STACK
259
263
VIRTUAL ADAPTER DRIVER
270
ARP SERVER EMULATOR
DHCP SERVER EMULATOR
272
264
ENCAPSULATION
DECAPSULATION
268
265
ENCRYPTION
DECRYPTION
266
262
PHYSICAL NETWORK ADAPTER'S DRIVER

FIG. 15

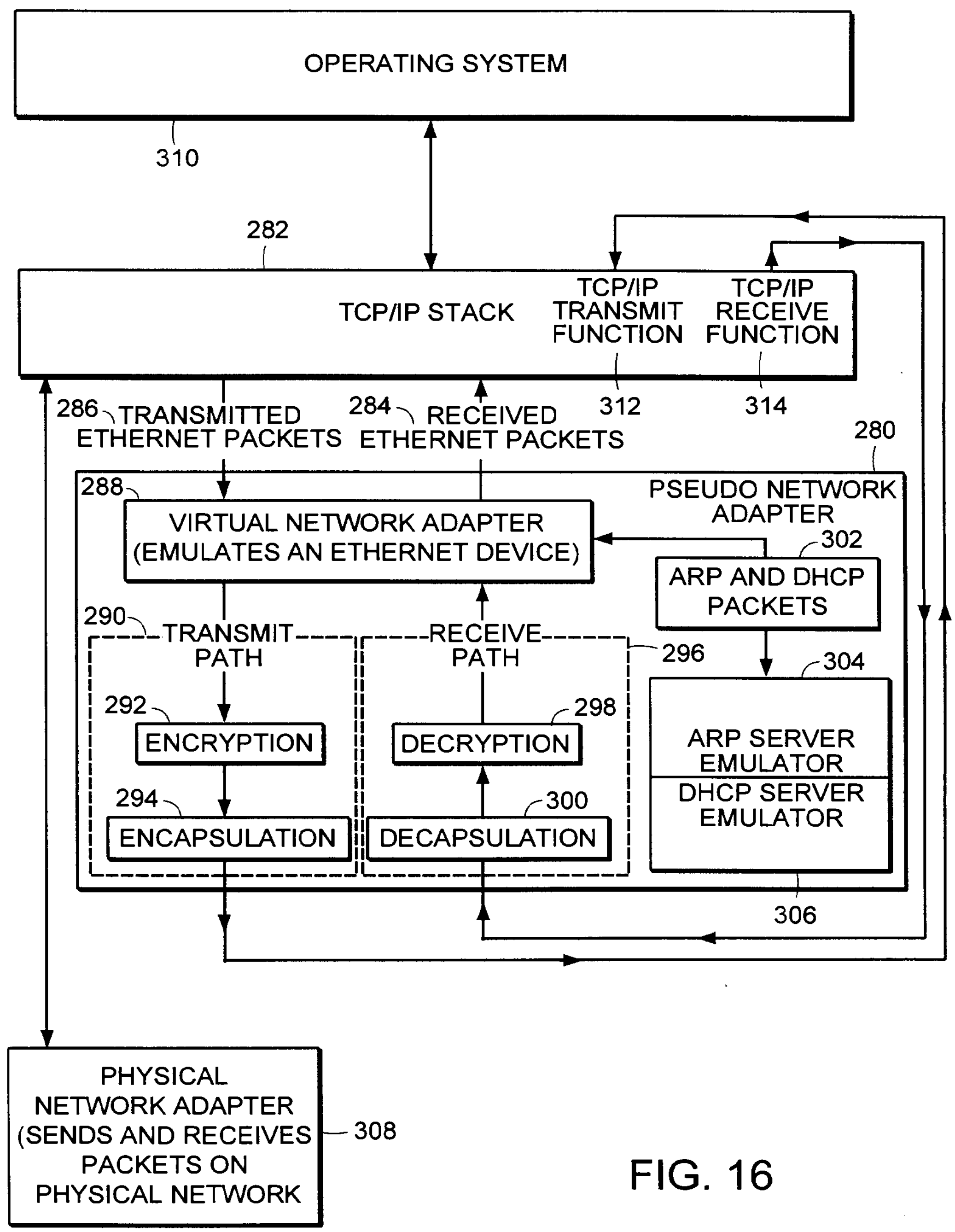

OPERATING SYSTEM
310
282
TCP/IP STACK
TCP/IP TRANSMIT FUNCTION
TCP/IP RECEIVE FUNCTION
312
314
286 TRANSMITTED ETHERNET PACKETS
284 RECEIVED ETHERNET PACKETS
280
288
PSEUDO NETWORK ADAPTER
VIRTUAL NETWORK ADAPTER (EMULATES AN ETHERNET DEVICE)
302
ARP AND DHCP PACKETS
290 TRANSMIT PATH
RECEIVE PATH
296
304
292
ENCRYPTION
DECRYPTION
298
ARP SERVER EMULATOR
DHCP SERVER EMULATOR
294
ENCAPSULATION
300
DECAPSULATION
306
PHYSICAL NETWORK ADAPTER (SENDS AND RECEIVES PACKETS ON PHYSICAL NETWORK
308

FIG. 16

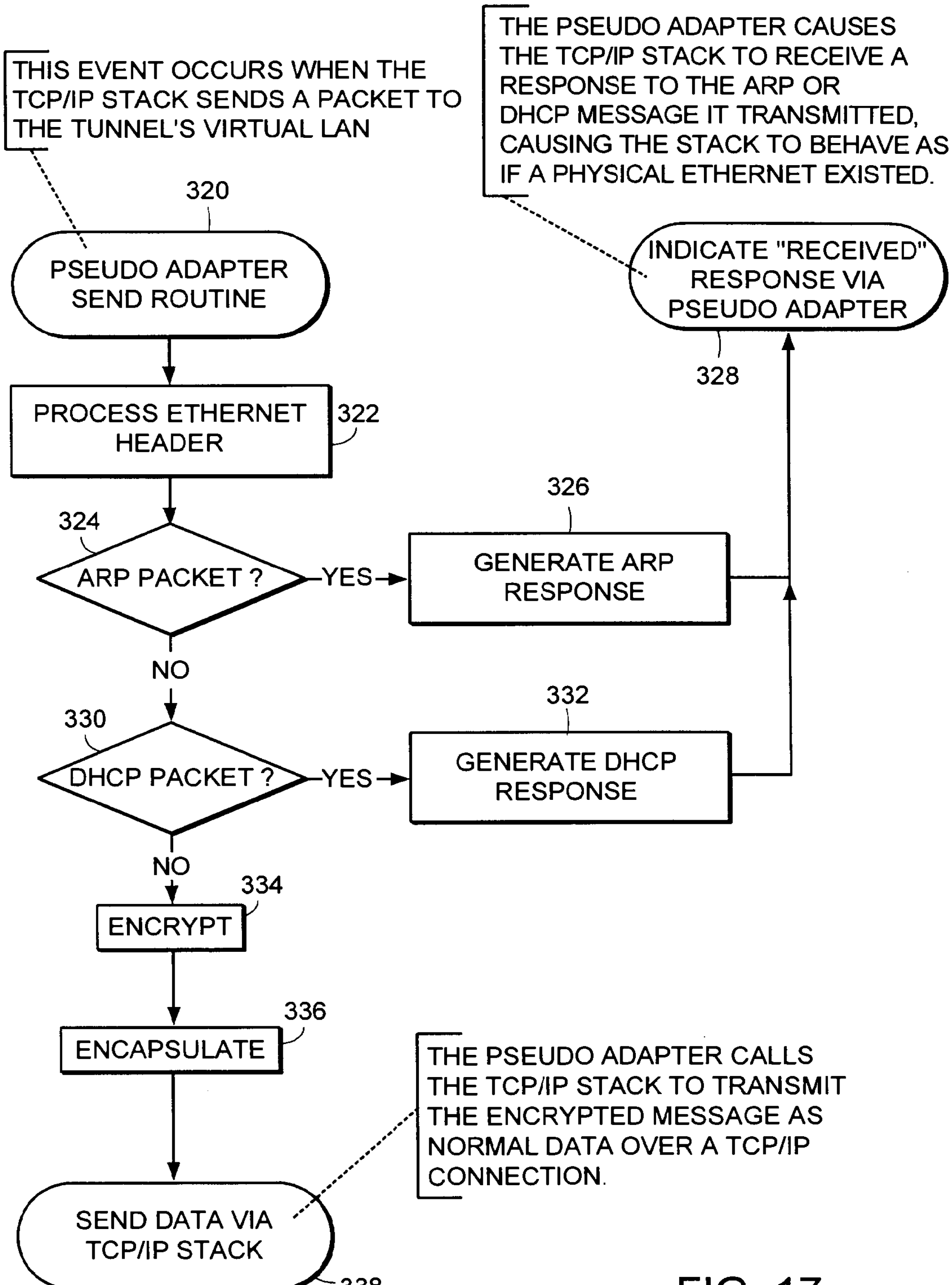

THIS EVENT OCCURS WHEN THE TCP/IP STACK SENDS A PACKET TO THE TUNNEL'S VIRTUAL LAN
THE PSEUDO ADAPTER CAUSES THE TCP/IP STACK TO RECEIVE A RESPONSE TO THE ARP OR DHCP MESSAGE IT TRANSMITTED, CAUSING THE STACK TO BEHAVE AS IF A PHYSICAL ETHERNET EXISTED.
320
PSEUDO ADAPTER SEND ROUTINE
INDICATE "RECEIVED" RESPONSE VIA PSEUDO ADAPTER
328
PROCESS ETHERNET HEADER
322
326
324
ARP PACKET ?
YES
GENERATE ARP RESPONSE
NO
332
330
DHCP PACKET ?
YES
GENERATE DHCP RESPONSE
NO
334
ENCRYPT
336
ENCAPSULATE
THE PSEUDO ADAPTER CALLS THE TCP/IP STACK TO TRANSMIT THE ENCRYPTED MESSAGE AS NORMAL DATA OVER A TCP/IP CONNECTION.
SEND DATA VIA TCP/IP STACK
338

FIG. 17

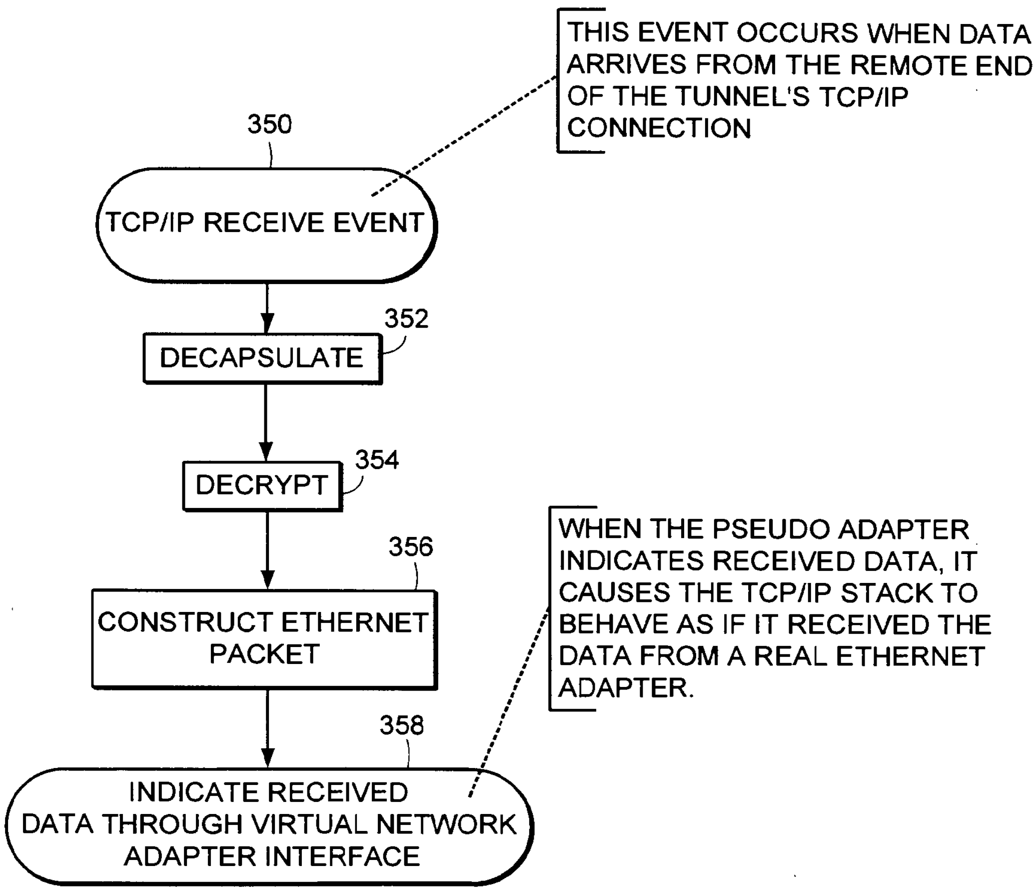

THIS EVENT OCCURS WHEN DATA ARRIVES FROM THE REMOTE END OF THE TUNNEL'S TCP/IP CONNECTION
350
TCP/IP RECEIVE EVENT
352
DECAPSULATE
354
DECRYPT
356
CONSTRUCT ETHERNET PACKET
WHEN THE PSEUDO ADAPTER INDICATES RECEIVED DATA, IT CAUSES THE TCP/IP STACK TO BEHAVE AS IF IT RECEIVED THE DATA FROM A REAL ETHERNET ADAPTER.
358
INDICATE RECEIVED DATA THROUGH VIRTUAL NETWORK ADAPTER INTERFACE

FIG. 18

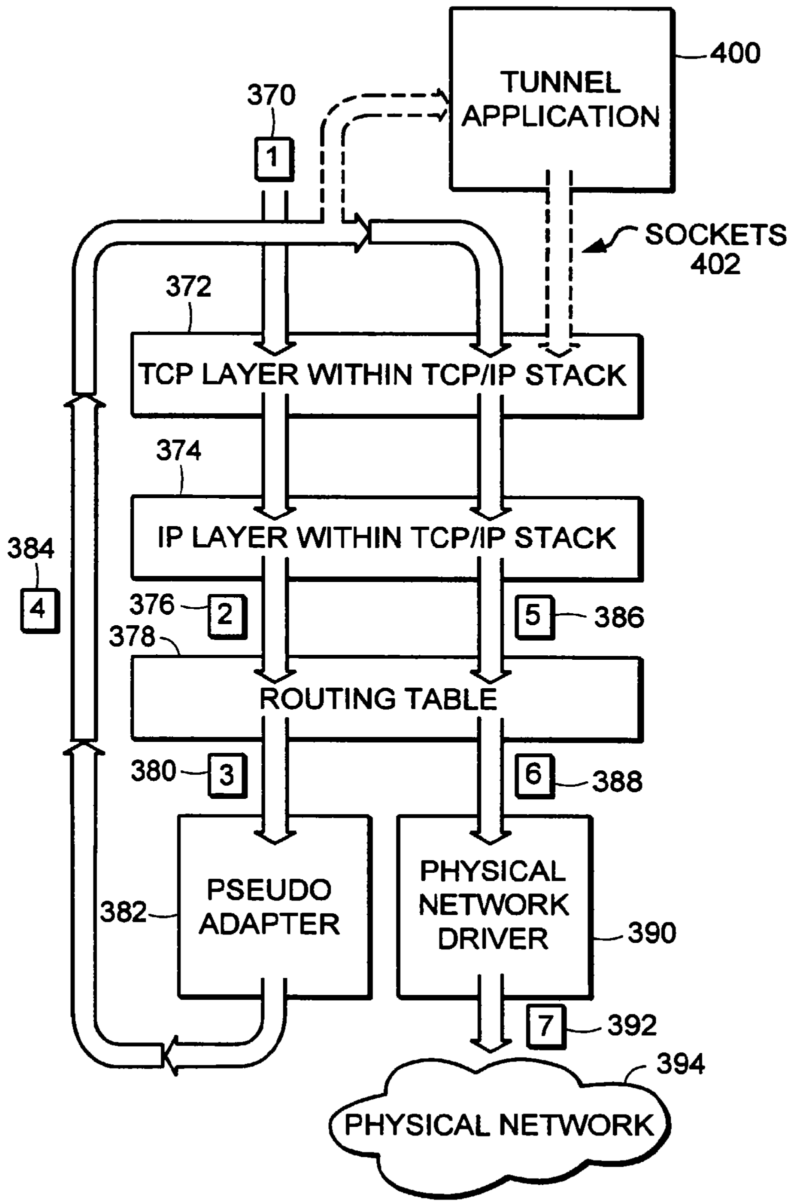

TUNNEL APPLICATION
400
370
1
SOCKETS 402
372
TCP LAYER WITHIN TCP/IP STACK
374
IP LAYER WITHIN TCP/IP STACK
384
4
376
2
378
5
386
ROUTING TABLE
380
3
6
388
PSEUDO ADAPTER
382
PHYSICAL NETWORK DRIVER
390
7
392
394
PHYSICAL NETWORK

FIG. 19

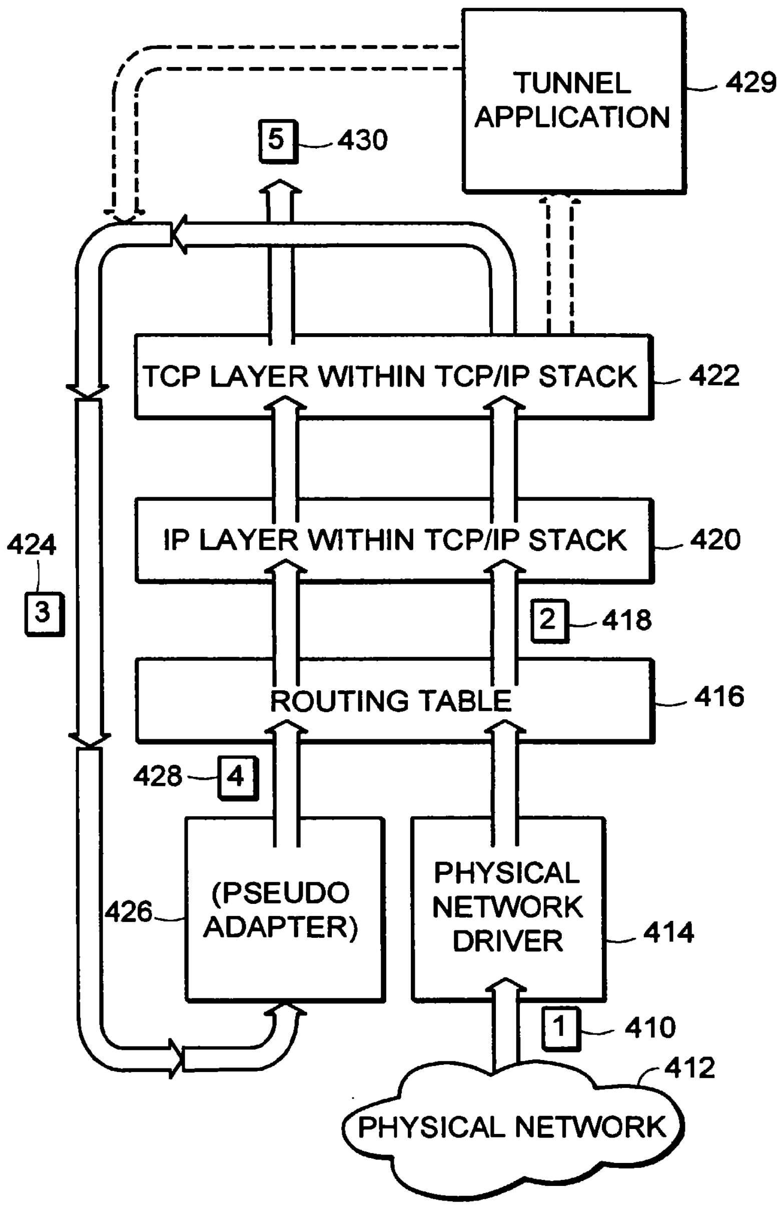

TUNNEL APPLICATION
429
5
430
TCP LAYER WITHIN TCP/IP STACK
422
IP LAYER WITHIN TCP/IP STACK
420
424
3
2
418
ROUTING TABLE
416
428
4
(PSEUDO ADAPTER)
426
PHYSICAL NETWORK DRIVER
414
1
410
412
PHYSICAL NETWORK

FIG. 20

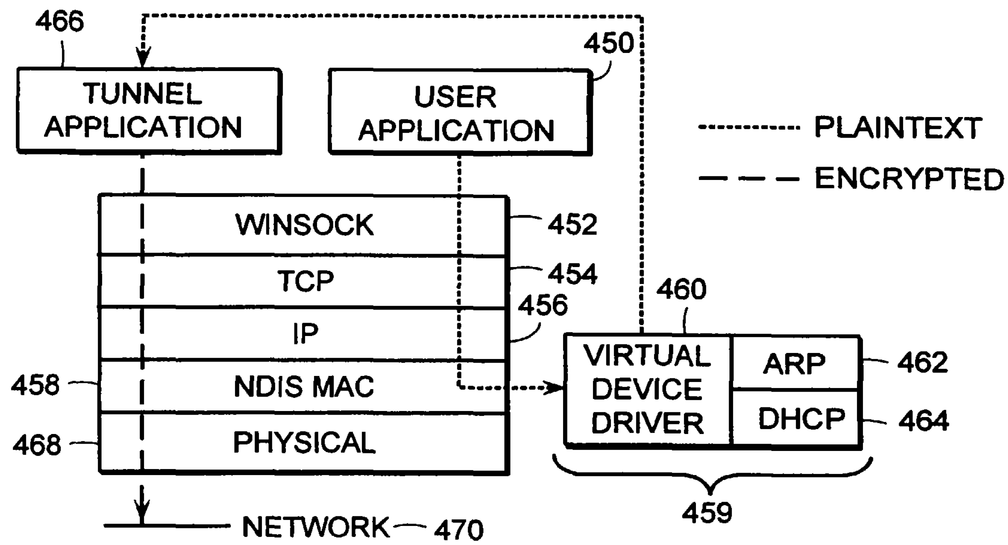

466
450
TUNNEL APPLICATION
USER APPLICATION
PLAINTEXT
ENCRYPTED
WINSOCK
452
TCP
454
456
460
IP
VIRTUAL DEVICE DRIVER
ARP
462
458
NDIS MAC
DHCP
464
468
PHYSICAL
NETWORK
470
459

FIG. 21

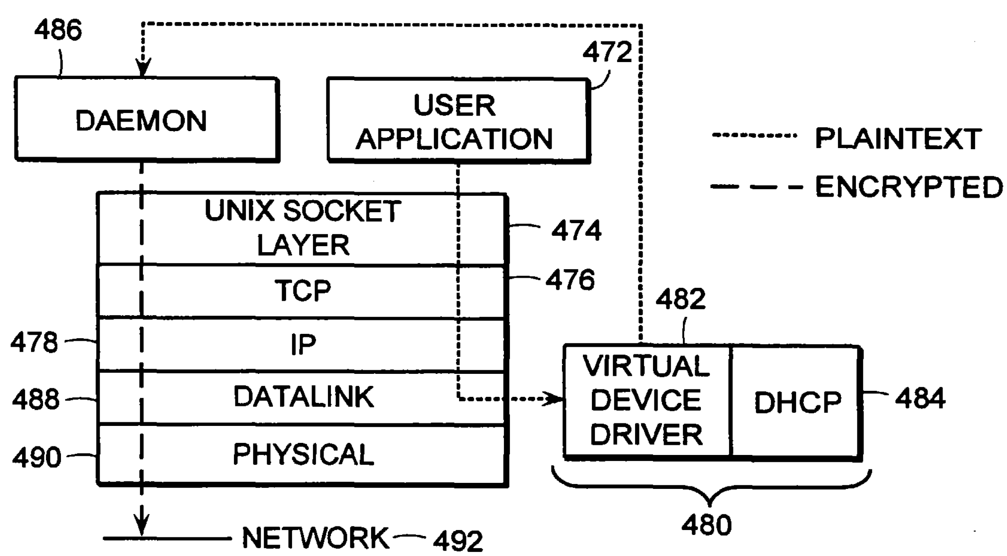

486
472
DAEMON
USER APPLICATION
PLAINTEXT
ENCRYPTED
UNIX SOCKET LAYER
474
TCP
476
482
478
IP
VIRTUAL DEVICE DRIVER
DHCP
484
488
DATALINK
490
PHYSICAL
NETWORK
492
480

FIG. 22

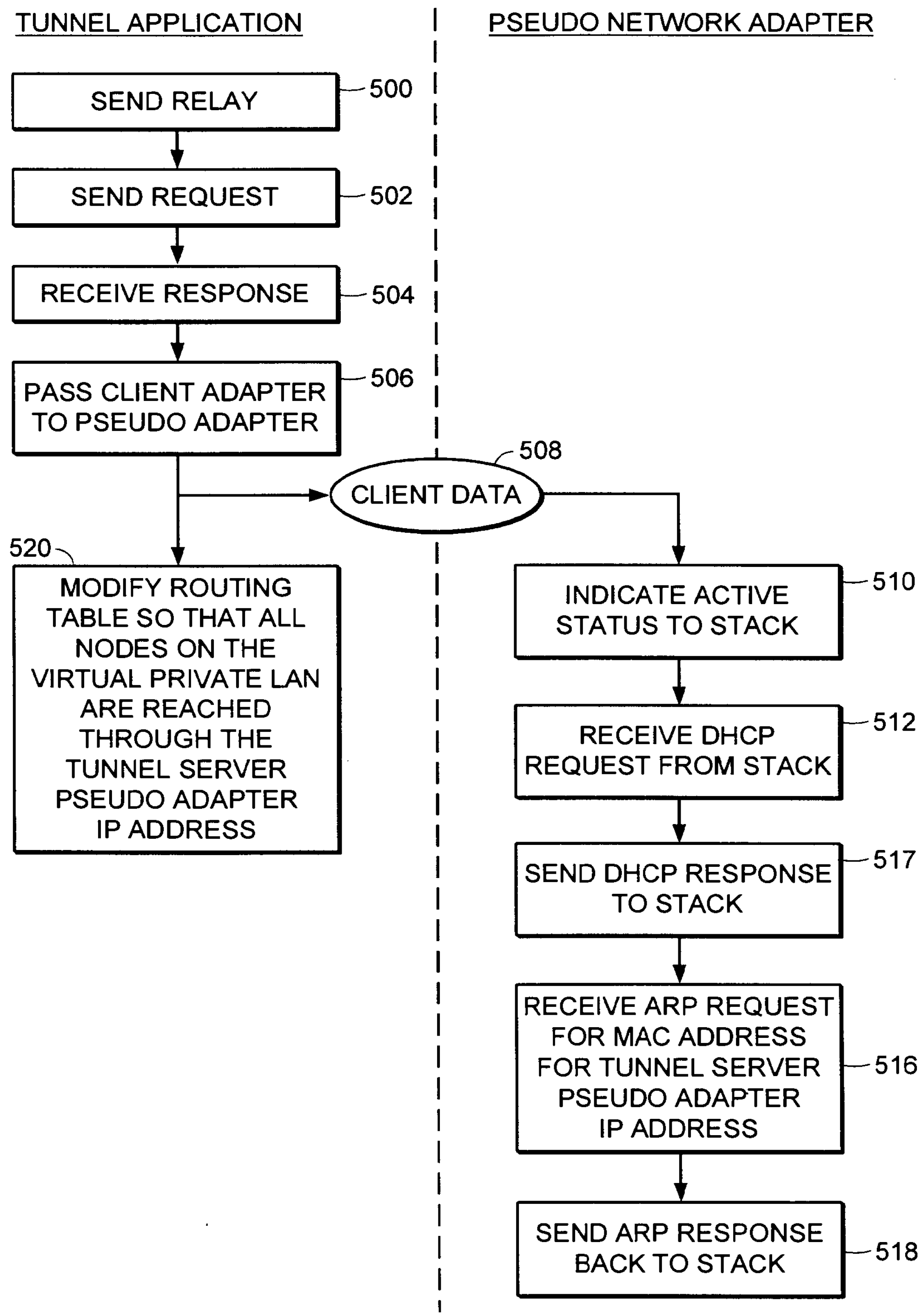

TUNNEL APPLICATION
PSEUDO NETWORK ADAPTER
SEND RELAY
500
SEND REQUEST
502
RECEIVE RESPONSE
504
PASS CLIENT ADAPTER TO PSEUDO ADAPTER
506
CLIENT DATA
508
520
MODIFY ROUTING TABLE SO THAT ALL NODES ON THE VIRTUAL PRIVATE LAN ARE REACHED THROUGH THE TUNNEL SERVER PSEUDO ADAPTER IP ADDRESS
INDICATE ACTIVE STATUS TO STACK
510
RECEIVE DHCP REQUEST FROM STACK
512
SEND DHCP RESPONSE TO STACK
517
RECEIVE ARP REQUEST FOR MAC ADDRESS FOR TUNNEL SERVER PSEUDO ADAPTER IP ADDRESS
516
SEND ARP RESPONSE BACK TO STACK
518

FIG. 23

GB2317792

Publication Title:

Virtual Private Network for encrypted firewall

Abstract:

Abstract of GB2317792

A system (10) for regulating the flow of messages through a firewall (18) having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer where if the message is not encrypted, it passes the unencrypted message up the network protocol stack to an application level proxy (50), and if the message is encrypted, it decrypts the message and passes the decrypted message up the network protocol stack to the application level proxy. The step of decrypting the message includes the step of executing a process at the IP layer to decrypt the message. Data supplied from the esp@cenet database - Worldwide

(12) UK Patent Application (19) GB (11) 2 317 792 (13) A

(43) Date of A Publication 01.04.1998

(21) Application No 9719816.2

(22) Date of Filing 17.09.1997

(30) Priority Data
(31) 08715343 (32) 18.09.1996 (33) US
08715668 18.09.1996

(71) Applicant(s)
Secure Computing Corporation

(Incorporated in USA - Delaware)

2675 Long Lake Road, Roseville, Minnesota 55113-2536, United States of America

(72) Inventor(s)
Spence Minear
Edward B Stockwell
Troy De Jongh

(74) Agent and/or Address for Service
Beresford & Co
2-5 Warwick Court, High Holborn, LONDON, WC1R 5DJ, United Kingdom

(51) INT CL^6
H04L 9/00

(52) UK CL (Edition P)
H4P PPEB
U1S S2124 S2209

(56) Documents Cited
WO 97/26735 A1 WO 97/26734 A1 WO 97/26731 A1
WO 97/23972 A1 WO 97/13340 A1

(58) Field of Search
UK CL (Edition P) **H4P PDCSA PDCSC PPEB**
INT CL^6 **H04L 9/00 9/32 29/06 29/08**
Online: WPI, INSPEC

(54) **Virtual Private Network for encrypted firewall**

(57) A system (10) for regulating the flow of messages through a firewall (18) having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer where if the message is not encrypted, it passes the unencrypted message up the network protocol stack to an application level proxy (50), and if the message is encrypted, it decrypts the message and passes the decrypted message up the network protocol stack to the application level proxy. The step of decrypting the message includes the step of executing a process at the IP layer to decrypt the message.

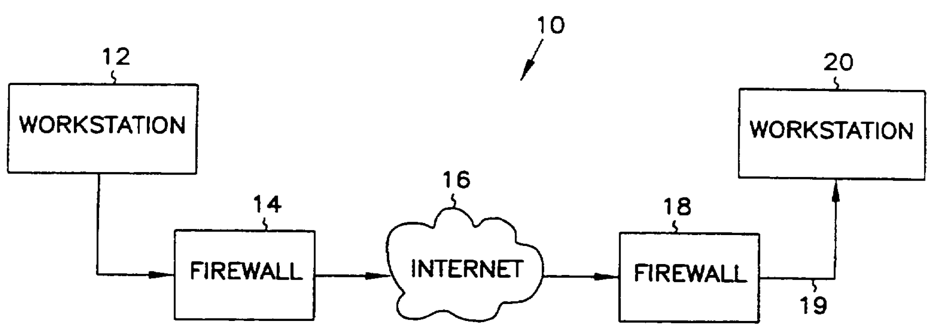


FIG. 1

GB 2 317 792 A

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.

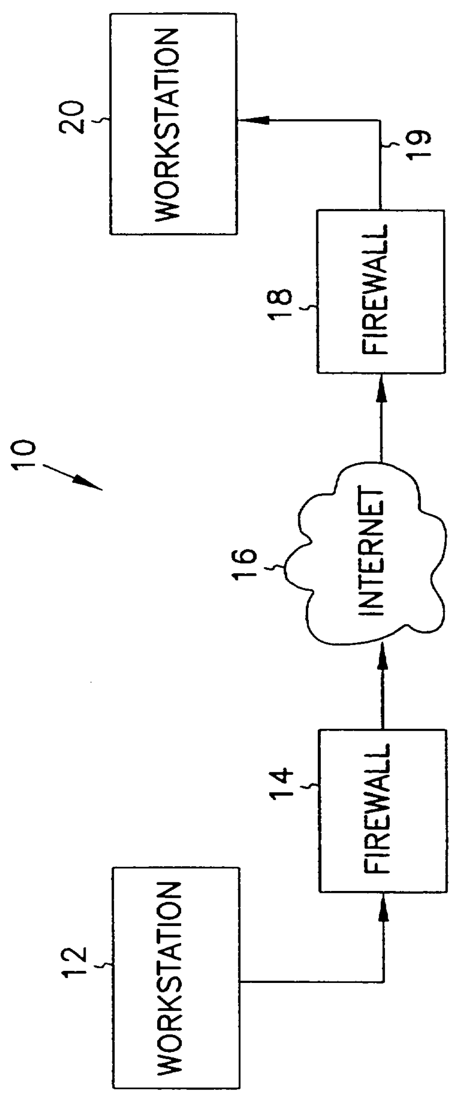

10
WORKSTATION
12
FIREWALL
14
INTERNET
16
FIREWALL
18
19
WORKSTATION
20

FIG. 1

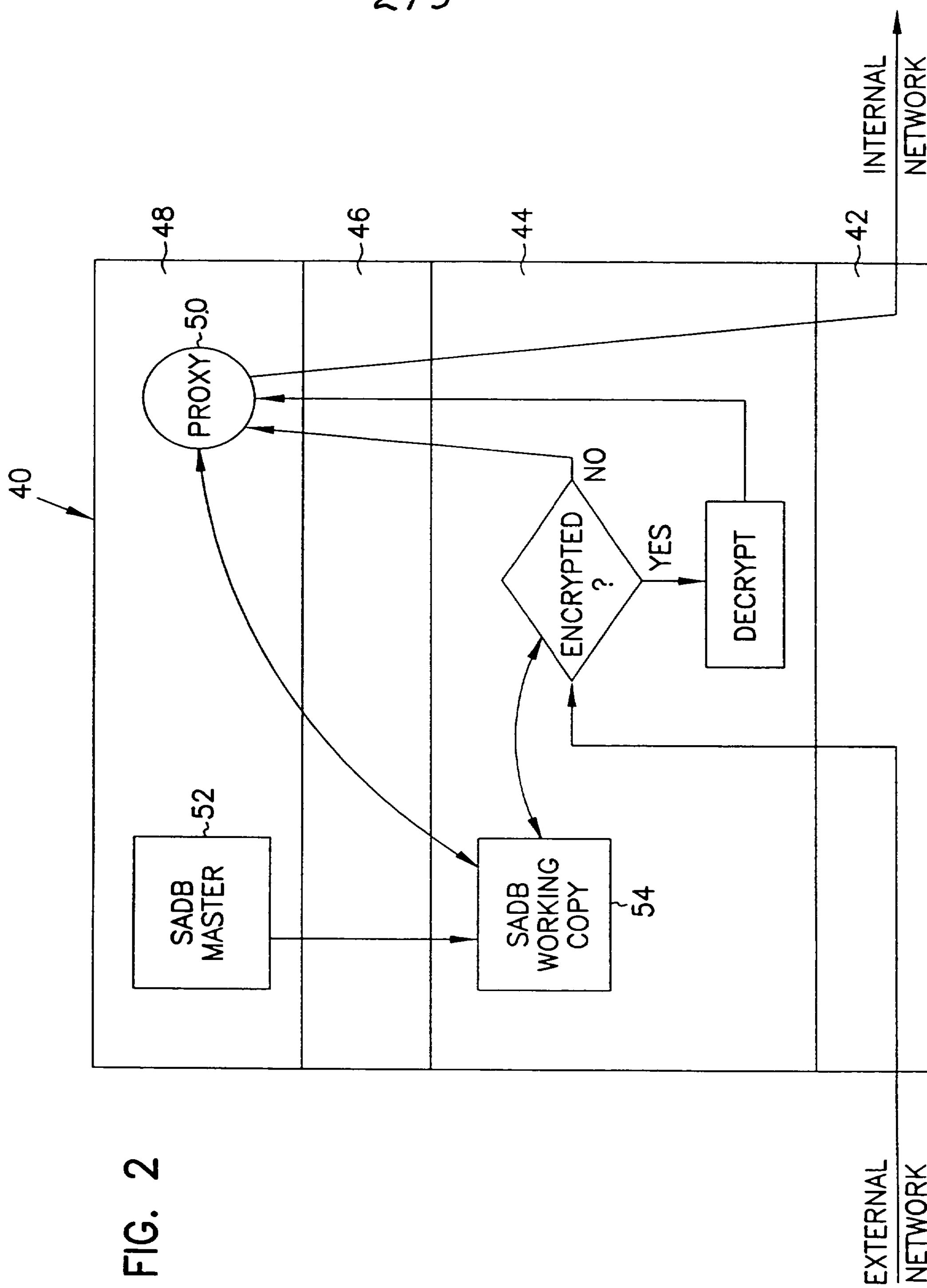
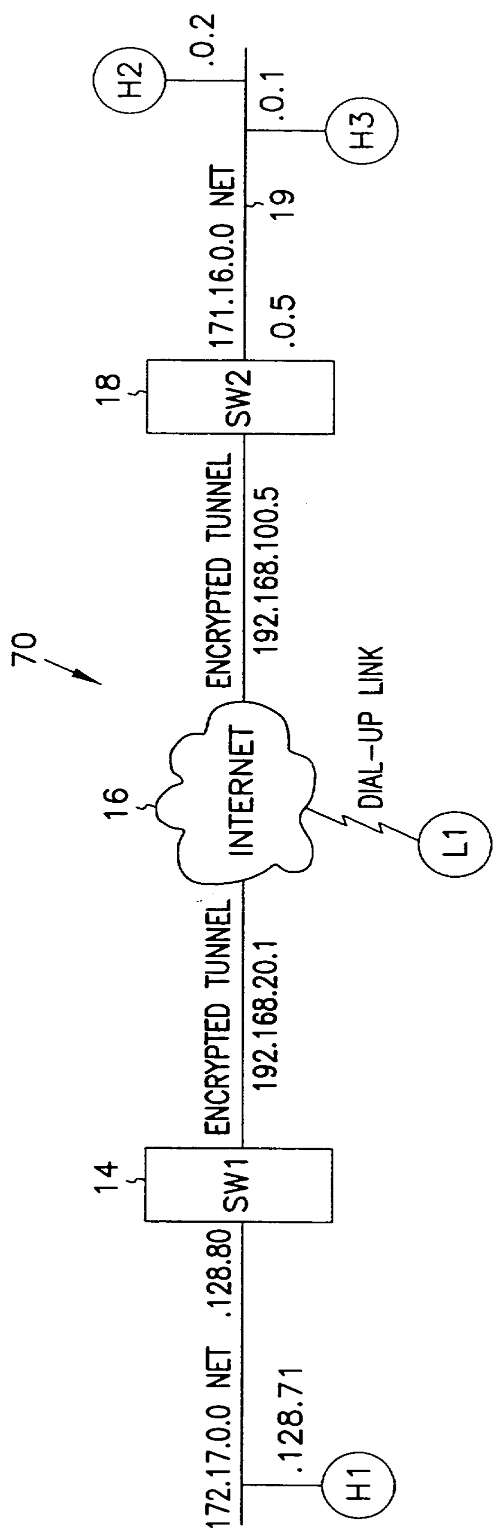

40
48
46
44
42
PROXY
50
SADB MASTER
52
SADB WORKING COPY
54
ENCRYPTED ?
YES
NO
DECRYPT
INTERNAL NETWORK
EXTERNAL NETWORK

FIG. 2

3/5



70
14
SW1
172.17.0.0 NET
.128.80
.128.71
H1
ENCRYPTED TUNNEL
192.168.20.1
16
INTERNET
DIAL-UP LINK
L1
ENCRYPTED TUNNEL
192.168.100.5
18
SW2
171.16.0.0 NET
.0.5
19
.0.1
.0.2
H2
H3

FIG. 3

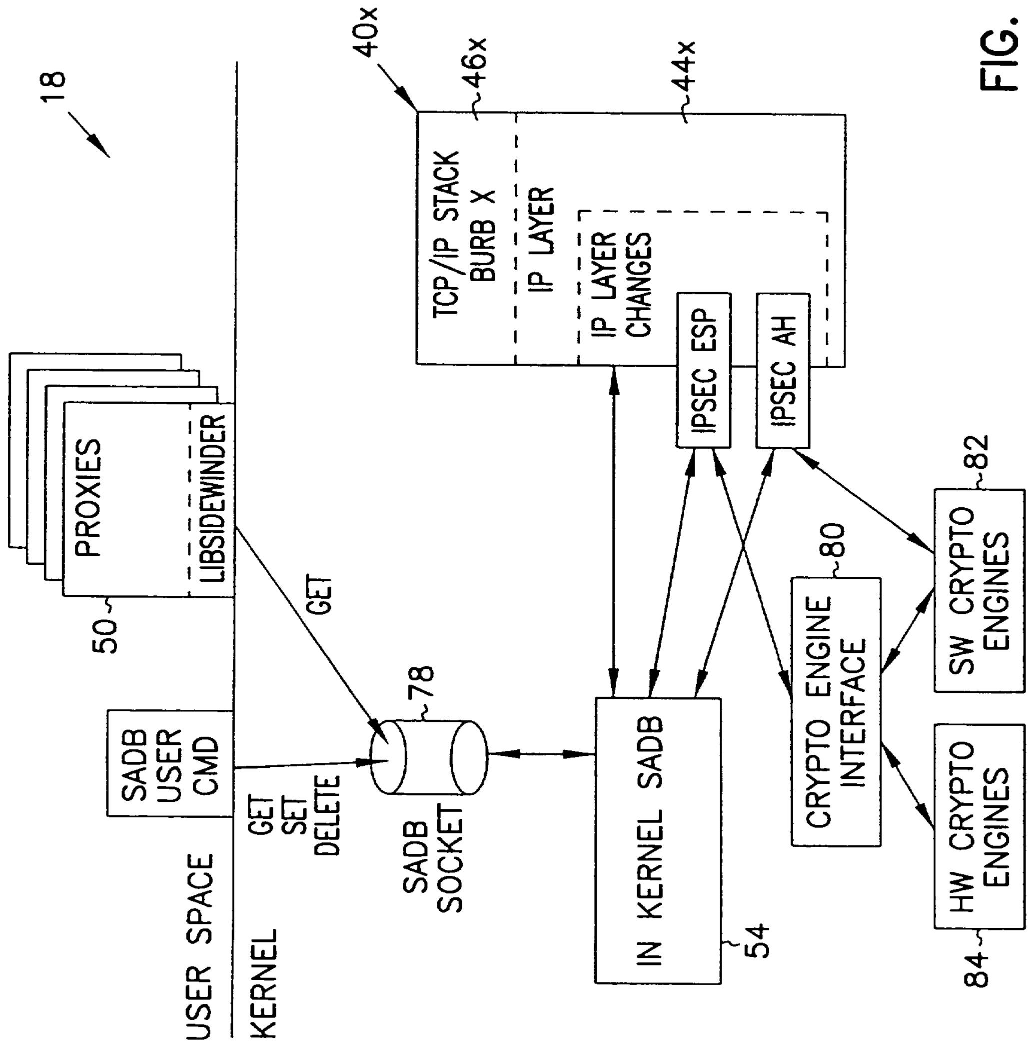

18
USER SPACE
KERNEL
SADB
USER
CMD
GET
SET
DELETE
PROXIES
LIBSIDEWINDER
50
GET
78
SADB
SOCKET
IN KERNEL SADB
54
40x
46x
44x
TCP/IP STACK
BURB X
IP LAYER
IP LAYER
CHANGES
IPSEC ESP
IPSEC AH
CRYPTO ENGINE
INTERFACE
80
SW CRYPTO
ENGINES
82
HW CRYPTO
ENGINES
84

FIG. 4

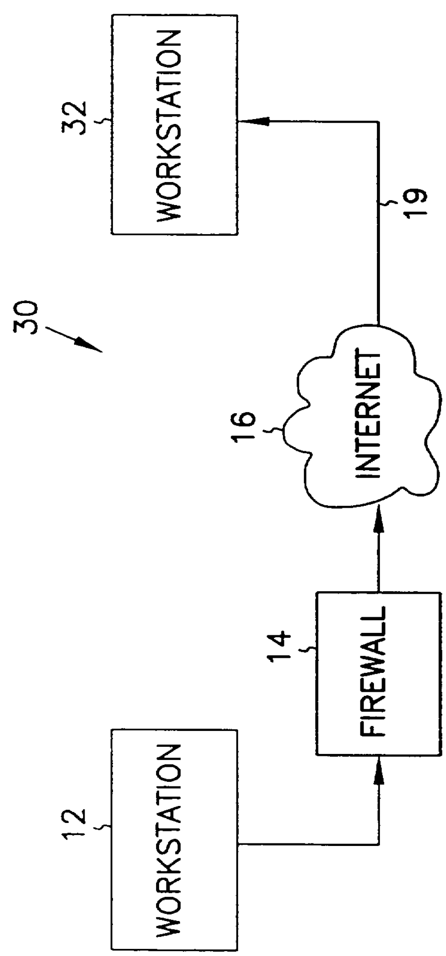

30
WORKSTATION
12
FIREWALL
14
INTERNET
16
19
WORKSTATION
32

FIG. 5

2317792

VIRTUAL PRIVATE NETWORK ON APPLICATION GATEWAY

Background of the Invention

Field of the Invention

The present invention pertains generally to network communications, and in particular to a system and method for securely transferring information between firewalls over an unprotected network.

Background Information

Firewalls have become an increasingly important part of network design. Firewalls provide protection of valuable resources on a private network while allowing communication and access with systems located on an unprotected network such as the Internet. In addition, they operate to block attacks on a private network arriving from the unprotected network by providing a single connection with limited services. A well designed firewall limits the security problems of an Internet connection to a single firewall computer system. This allows an organization to focus their network security efforts on the definition of the security policy enforced by the firewall. An example of a firewall is given in "SYSTEM AND METHOD FOR PROVIDING SECURE INTERNETWORK SERVICES" by Boebert et al. (PCT Published Application No. WO 96/13113, published on May 2, 1996), the description of which is hereby incorporated by reference. Another description of a firewall is provided by Dan Thomsen in "Type Enforcement: the new security model", *Proceedings: Multimedia: Full-Service Impact on Business, Education, and the Home,* SPIE Vol. 2617, p. 143, August 1996. Yet another such system is described in "SYSTEM AND METHOD FOR ACHIEVING NETWORK SEPARATION" by Gooderum et al. (PCT Published Application No. WO 97/29413, published on August 14, 1997), the description of which is hereby incorporated by reference. All the above systems are examples of application level gateways. Application level gateways use proxies or other such mechanisms operating at the application layer to process traffic through the firewall. As such, they can review not only the message traffic but also message content. In addition, they provide authentication and identification services, access control and auditing.

Data to be transferred on unprotected networks like the Internet is susceptible to electronic eavesdropping and accidental (or deliberate) corruption. Although a firewall can protect data within a private network from attacks launched from the unprotected network, even that data is vulnerable to both eavesdropping and corruption when transferred from the private network to an external machine. To address this danger, the Internet Engineering Task Force (IETF) developed a standard for protecting data transferred between firewalls over an unprotected network. The Internet Protocol Security (IPSEC) standard calls for encrypting data before it leaves the first firewall, and then decrypting the data when it is received by the second firewall. The decrypted data is then delivered to its destination, usually a user workstation connected to the second firewall. For this reason IPSEC encryption is sometimes called *firewall-to-firewall encryption* (FFE) and the connection between a workstation connected to the first firewall and a client or server connected to the second firewall is termed a *virtual private network*, or VPN.

The two main components of IPSEC security are data encryption and sender authentication. Data encryption increases the cost and time required for the eavesdropping party to read the transmitted data. Sender authentication ensures that the destination system can verify whether or not the encrypted data was actually sent from the workstation that it was supposed to be sent from. The IPSEC standard defines an encapsulated payload (ESP) as the mechanism used to transfer encrypted data. The standard defines an authentication header (AH) as the mechanism for establishing the sending workstation's identity.

Through the proper use of encryption, the problems of eavesdropping and corruption can be avoided; in effect, a protected connection is established from the internal network connected to one firewall through to an internal network connected to the second firewall. In addition, IPSEC can be used to provide a protected connection to an external computing system such as a portable personal computer.

IPSEC encryption and decryption work within the IP layer of the network protocol stack. This means that all communication between two IP addresses will be protected because all interfirewall communication must go through the IP layer. Such an approach is preferable over encryption and decryption at higher levels in the network protocol stack since when encryption is performed at layers higher than the IP layer more work is required to ensure that all supported communication is properly protected. In addition, since IPSEC encryption is handled below the Transport layer, IPSEC can encrypt data sent by any application. IPSEC therefore becomes a transparent add-on to such protocols as TCP and UDP.

Since, however, IPSEC decryption occurs at the IP layer, it can be difficult to port IPSEC to an application level gateway while still maintaining control at the proxy over authentication, message content, access control and auditing. Although the IPSEC specification in RFC 1825 suggests the use of a mandatory access control mechanism in a multi-level secure (MLS) network to compare a security level associated with the message with the security level of the receiving process, such an approach provides only limited utility in an application level gateway environment. In fact, implementations on application level gateways to date have simply relied on the fact that the message was IPSEC-encrypted as assurance that the message is legitimate and have simply decoded and forwarded the message to its destination. This creates, however, a potential chink in the firewall by assuming that the encrypted communication has access to all services.

What is needed is a method of handling IPSEC messages within an application level gateway which overcomes the above deficiencies. The method should allow control over access by an IPSEC connection to individual services within the internal network.

<u>Summary of the Invention</u>

The present invention is a system and method for regulating the flow of messages through a firewall having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of determining, at the IP layer, if a message is encrypted, if the message is not encrypted, passing the unencrypted message up the network protocol stack to an application level proxy, and if the message is encrypted, decrypting the message and passing the decrypted message up the network protocol stack to the application level proxy, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message.

According to another aspect of the present invention, a system and method is described for authenticating the sender of a message within a computer system having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of determining, at the IP layer, if the message is encrypted, if the message is encrypted, decrypting the message, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message, passing the decrypted message up the network protocol stack to an application level proxy, determining an authentication protocol appropriate for the message, and executing the authentication protocol to authenticate the sender of the message.

Brief Description of the Drawings

In the following detailed description of example embodiments of the invention, reference is made to the accompanying drawings which form a part hereof, and which is shown by way of illustration only, specific embodiments in which the invention may be practiced. It is to be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the present invention.

In the drawings, where like numerals refer to like components throughout the several views:

Figure 1 is a functional block diagram of an application level gateway-implemented firewall-to-firewall encryption scheme according to the present invention;

Figure 2 is a block diagram showing access control checking of both encrypted and unencrypted messages in network protocol stack according to the present invention;

Figure 3 is a block diagram of a representative application level gateway-implemented firewall-to-firewall encryption scheme;

Figure 4 is a block diagram of one embodiment of a network-separated protocol stack implementing IPSEC according to the present invention; and

Figure 5 is a functional block diagram of a firewall-to-workstation encryption scheme according to the present invention.

Description of the Preferred Embodiments

In the following detailed description of the preferred embodiment, references made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific preferred embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that structural, logical, physical, architectural, and electrical changes may be made without departing from the spirit and scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims and their equivalents.

A system 10 which can be used for firewall-to-firewall encryption (FFE) is shown in Figure 1. In Figure 1, system 10 includes a workstation 12 communicating through a firewall 14 to an unprotected network 16 such as the Internet. System 10 also includes a workstation 20 communicating through a firewall 18 to unprotected network 16. In one embodiment, firewall 18 is an application level gateway.

As noted above, IPSEC encryption and decryption work within the IP layer of the network protocol stack. This means that all communications between two IP addresses will be protected because all interfirewall communication must pass through the IP layer. IPSEC takes the standard

Internet packet and converts it into a carrier packet. The carrier packet is designed to do two things: to conceal the contents of the original packet (encryption) and to provide a mechanism by which the receiving firewall can verify the source of the packet (authentication). In one embodiment of the present invention, each IPSEC carrier packet includes both an authentication header used to authenticate the sending machine and an encapsulated payload containing encrypted data. The authentication header and the encapsulated payload features of IPSEC can, however, be used independently. As required in RFC 1825, DES-CBC is provided for use in encrypting the encapsulated payload while the authentication header uses keyed MD5.

To use IPSEC, you must create a *security association* (SA) for each destination IP address. In one embodiment, each SA contains the following information:

- Security Parameters Index (SPI) - The index used to find a SA on receipt of an IPSEC datagram.
- Destination IP address - The address used to find the SA and trigger use of IPSEC processing on output.
- The peer SPI - The SPI value to put on a IPSEC datagram on output.
- The peer IP address - The destination IP address to be put into the packet header if IPSEC Tunnel mode is used.
- The Encryption Security Payload (ESP) algorithm to be used.
- The ESP key to used for decryption of input datagrams.
- The ESP key to used for encryption of output datagrams.
- The authentication (AH) algorithm to be used.
- The AH key to be used for validation of input packets.
- The AH key to be used for generation of the authentication data for output datagrams.

The combination of a given Security Parameter Index and Destination IP address uniquely identifies a particular "Security Association." In one embodiment, the sending firewall uses the sending userid and Destination Address to select an appropriate Security Association (and hence SPI value). The receiving firewall uses the combination of SPI value and Source address to obtain the appropriate Security Association.

A security association is normally one-way. An authenticated communications session between two firewalls will normally have two Security Parameter Indexes in use (one in each direction). The combination of a particular Security Parameter Index and a particular Destination Address uniquely identifies the Security Association.

More information on the specifics of an IPSEC FFE implementation can be obtained from the standards developed by the IPSEC work group and documented in *Security Architecture for IP* (RFC 1825) and in RFC's 1826-1829.

When a datagram is received from unprotected network 16 or is to be transmitted to a destination across unprotected network 16, the firewall must be able to determine the algorithms, keys, etc. that must be used to process the datagram correctly. In one embodiment, this information is obtained via a security association lookup. In one such embodiment, the lookup routine is passed several arguments: the source IP address if the datagram is being received from network 16 or the destination IP address if the datagram is to be transmitted across network 16, the SPI, and a flag that is used to indicate whether the lookup is being done to receive or transmit a datagram.

When an IPSEC datagram is received by firewall 18 from unprotected network 16, the SPI and source IP address are determined by looking in the datagram. In one embodiment a Security Association Database (SADB) stored within firewall 18 is searched for the entry with a matching SPI. In one such embodiment, security associations can be set up based on network address as well as a more granular host address. This allows the network administrator to create a security association between two firewalls with only a couple of lines in a configuration file on each machine. For such embodiments, the entry in the Security Association Database that has both the matching SPI and the longest address match is selected as the SA entry. In another such embodiment, each SA has a prefix length value associated with the address. An address match on a SA entry means that the addresses match for the number of bits specified by the prefix length value.

There are two exceptions to this search process. First, when an SA entry is set marked as being dynamic it implies that the user of this SA may not have a fixed IP address. In this case the match is fully determined by the SPI value. Thus it is necessary that the SPI values for such SA entries be unique in the SADB. The second exception is for SA entries marked as tunnel mode entries. In this case it is normally the case that the sending entity will hide its source address so that all that is visible on the public wire is the destination address. In this case, like in the case where the SA entries are for dynamic IP addresses, the search is done exclusively on the basis of the SPI.

When transmitting a datagram across unprotected network 16 the SADB is searched using only the destination address as an input. In this case the entry which has the longest address match is selected and returned to the calling routine.

In one embodiment, if firewall 18 receives datagrams which are identified as either an IP_PROTO_IPSEC_ESP or IP_PROTO_IPSEC_AH protocol datagram, there must be a corresponding SA in the SADB or else firewall 18 will drop the packet and an audit message will be generated. Such an occurrence might indicate a possible attack or it might simply be a symptom of an erroneous key entry in the Security Association Database.

In a system such as system 10, application level gateway firewall 18 acts as a buffer between unprotected network 16 and workstations such as workstation 20. Messages coming from unprotected network 16 are reviewed and a determination is made as to whether execution of an authentication and identification protocol is warranted. In contrast to previous systems, system 10 also performs this same determination on IPSEC-encrypted messages. If desired, the same authentication and identification can be made on messages to be transferred from workstation 20 to unprotected network 16. Figure 2 illustrates one way of authenticating both encrypted and unencrypted messages in a system such as system 10.

In the system of Figure 2 a network protocol stack 40 includes a physical layer 42, an Internet protocol (IP) layer 44, a Transport layer 46 and an application layer 48. Such a protocol stack exists, for instance on application level gateway firewall 18 of Figure 1. An application executing in application layer 48 can communicate to an application executing on another system by preparing a message and transmitting it through one of the existing transport services executing on transport layer 46. Transport layer 46 in turn uses a process executing in IP layer 44 to continue the transfer. Physical layer 42 provides the software needed to transfer data through the communication hardware (e.g., a network interface card or a modem). As noted above, IPSEC executes within IP layer 44. Encryption and authentication is transparent to the host as long as the network administrator has the Security Association Database correctly configured and a key management mechanism is in place on the firewall.

In application level gateway firewall 18, a proxy 50 operating within application layer 48 processes messages transferred between internal and external networks. All network-to-network traffic must pass through one of the proxies within application layer 48 before being the transfer across networks is allowed. A message arriving from external network 16 is examined at IP layer 44 and an SADB is queried to determine if the source address and SPI are associated with an SA. In the embodiment shown in Figure 2, an SADB Master copy 52 is maintained in persistent memory at application layer 48 while a copy 54 of SADB is maintained in volatile memory within the kernel. If the message is supposed to be encrypted, the message is decrypted based on the algorithm and key associated with the particular SA and the message is transferred up through transport layer 46 to proxy 50. Proxy 50 examines the source and destination addresses and the type of service desired and decides whether authentication of the sender is warranted. If so, proxy 50 initiates an authentication protocol. The protocol may be as simple as requesting a user name and password or it may include a challenge/response authentication process. Proxy 50 also looks to see whether the message coming in was encrypted or not and may factor that into whether a particular type of authentication is needed. In Telnet, for instance, user name/password authentication may be sufficient for an FFE link while the security policy may dictate that a more stringent challenge/response protocol is needed for unencrypted links. In that case, proxy 50 will be a Telnet proxy and it will base its authentication protocol on whether the link was encrypted or not.

Since IPSEC executes within IP layer 44 there is no need for host firewalls to update their applications. Users that already have IPSEC available on their own host machine will, however, have to request that the firewall administrator set up SA's in the SADB for their traffic.

In the embodiment shown in Figure 2, a working copy 54 of the Security Association Database consisting of all currently active SA's is kept resident in memory for ready access by IP layer processing as datagrams are received and transmitted. In addition, a working master copy 52 of the SADB is maintained in a file in nonvolatile memory. During system startup and initialization processing the content of all of the required SA's in master SADB 52 is added to the working copy 54 stored in kernel memory.

In one embodiment, firewall 18 maintains different levels of security on internal and external network interfaces. It is desirable for a firewall to have different levels of security on both the internal and external interfaces. In one embodiment, firewall 18 supports three different levels, numbered 0 through 2. These levels provide a simple policy mechanism that controls permission for both in-bound and out-bound packets.

- Level 0 - do not allow any in-bound or out-bound traffic unless there is a security association between the source and destination.

- Level 1 - Allow both in-bound and out-bound non-IPSEC traffic but force the use of IPSEC if a SA exists for the address. (To support this firewall 18 must look for a SA for each in-bound datagram.)
- Level 2 - allow NULL security associations to exist. NULL associations are just like normal security associations, except no encryption or authentication transform is performed on in-bound or out-bound packets that correspond to this NULL association. With Level 2 enabled, the machine will still receive unprotected traffic, but it will not transmit unless Level 1 is enabled.

The default protection level established when the Security Association Database (SADB) is initialized at boot time is 1 for in-bound traffic and 2 for out-bound traffic.

An Access Control List, or ACL, is a list of rules that regulate the flow of Internet connections through a firewall. These rules control how a firewall's servers and proxies will react to connection attempts. When a server or proxy receives an incoming connection, it performs an ACL check on that connection.

An ACL check compares a set of parameters associated with the connection against a list of ACL rules. The rules determine whether the connection is allowed or denied. A rule can also have one or more side effects. A side effect causes the proxy to change its behavior in some fashion. For example, a common side effect is to redirect the destination IP address to an alternate machine. In addition to IP connection attempts, ACL checks can also made on the console logins and on logins made from serial ports. Finally, ACL checks can also be made on behalf of IP access devices, such as a Cisco box, through the use of the industry standard TACACS+ protocol.

In one embodiment, the ACL is managed by an acld daemon running in the kernel of firewalls 10 and 30. The acld daemon receives two types of requests, one to query the ACL and one to administer it. In one such embodiment, the ACL is stored in a relational database such as the Oracle database for fast access. By using such a database, query execution is asynchronous and many queries can be executing concurrently. In addition, these types of databases are designed to manipulate long lists of rules quickly and efficiently. These qualities ensure that a given query cannot hang up the process that issued the query for any appreciable time (> 1-2 seconds).

In one such embodiment, the database can hold up to 100,000 users and up to 10,000 hosts but can be scaled up to the capacity of the underlying database engine. The results of an ACL check is cached, allowing repeated checks to be turned around very quickly.

Applications on firewalls 10 and 30 can query acld to determine if a given connection attempt should be allowed to succeed. In one embodiment, the types of applications (i.e. "agents") that can make ACL queries can be divided into four classes:

1) Proxies. These allow connections to pass through firewall 10 or 30 in order to provide access to a remote service. They include tnauthp (authenticated telnet proxy), pftp (FTP proxy), httpp (HTTP proxy), and tcpgsp (TCP generic service proxy).

2) Servers. These provide a service on the firewall itself. They include ftpd and httpd.

3) Login agents. Login agent is a program on the firewall that can create a Unix shell. It is not considered a server because it cannot receive IP connections. One example is /usr/bin/login when used to create a dialup session or a console session on firewall 10 or 30. Another example is the command *srole*.

4) Network Access Servers (NAS). NAS is a remote IP access device, typically a dialup box manufactured by such companies as Cisco or Bridge. The NAS usually provides dialup telnet service and may also provide SLIP or PPP service.

Proxies, servers, login agents, and NASes make queries to acld to determine if a given connection attempt should be allowed to succeed. All of the agents except NAS make their queries directly. NAS, because it is remote, must communicate via an auxiliary daemon that typically uses an industry standard protocol such as RADIUS or TACACS+. The auxiliary daemon (e.g., tacradd) in turn forwards the query to local acld.

As a side effect of the query, acld tells the agent if authentication is needed. If no authentication is needed, the connection proceeds immediately. Otherwise acld provides (as another side effect) a list of allowed authentication methods that the user can choose from. The agent can present a menu of choices or simply pick the first authentication method by default. Typical authentication methods include plain password, SNK DSS, SDI SecurID, LOCKout DES, and LOCKout FORTEZZA. In one embodiment, the list of allowed authentication methods varies depending on the host name, user name, time of day, or any combination thereof.

In the case of a Level 0 policy, it would be safe to assume that all incoming traffic is encrypted or authenticated. In the case of Levels 1 through 2, a determination must be made whether or not a security association exists for a given peer. Otherwise an application may believe that in-bound traffic has been authenticated when it really has not. (That is why it is necessary to look for an SA on input of each non-IPSEC datagram.)

In one embodiment, a flag which accompanies the message as it is sent from IP layer 44 to proxy 50 indicates whether the incoming message was or was not encrypted. In another embodiment, proxy 50 accesses Security Association Database 54 (the table in the kernel can be queried via an SADB routing socket (PF-SADB)) to determine whether or not a security association exists for a given peer.. The SADB socket is much like a routing socket found in the stock BSD 4.4 kernel (protocol family PF-ROUTE) except that PF-SADB sockets are used to maintain the Security Association Database (SADB) instead of the routing table. Because the private keys used for encryption, decryption, and keyed authentication are stored in this table, access must be strictly prohibited and allowed to only administrators and key management daemons. Care must be taken when allowing user-level daemons access to `/dev/mem` or `/dev/kmem` as well, since the keys are stored in kernel memory and could be exposed with some creative hacking.

In one embodiment, a command-line tool called `sadb` is used to support the generation and maintenance of in-kernel version 54 of SADB. The primary interface between this tool and the SADB is the PF-SADB socket. The kernel provides socket processing to receive client requests to add, update, or change entries in in-kernel SADB 54. As noted above, the default protection level established when the Security Association Database (SADB) is initialized at boot time is 1 for in-bound traffic and 2 for out-bound traffic. This may be changed by the use of the `sadb` command.

The existing `sadb` command was derived from the NIST implementation of IPSEC. As noted above, this tool is much like `route` in that it uses a special socket to pass data structures in and out of the kernel. There are three commands recognized by the `sadb` command: ***get, set, delete***. The following simple shell script supports adding and removing a single SA entry to SADB 54. It shows one embodiment of a parameter order for adding a SA to the SADB.

```
# ! /bin/sh
if [ $# -ne 1 ]
then
     echo "usage: $0 <on>|<off>" >&2
     exit 1
fi
ONOFF=$1

addsa ()
{
IPADDRESS=$2
PEERADDRESS=0.0.0.0
PREFIXLEN=0                 # Num of bits, 0 => full 32
bit match
LOCALADDRESS=0.0.0.0
REALADDRESS=0.0.0.0
PORT=0
PROTOCOL=0
UID=0
DESALG=1                    # I = DES-CBC
IVLEN=4                     # bytes
DESKEY=0b0b0b0b0b0b0b0b
DESKEYLEN=8                 # bytes
AHALG=1                     # 1 = MD5
AHKEY=30313233343536373031323334353637
AHKEYLEN=16                 # bytes
LOCAL_SPI=$1
```

```
PEER_SPI=$1
TUNNEL_MODE=0
AHRESULTLEN=4
COMBINED_MODE=1            # On output, 1 = ESP, then
AH; 0 = AH, then ESP
DYNAMIC_FLAG=0

if [ "$ONOFF" = "on"
then
     ./sadb add dst $IPADDRESS $PREFIXLEN $LOCAL_SPI
$UID $PEERADDRESS $PEER_SPI $TUNNEL_MODE $LOCALADDRESS
$REALADDRESS $PROTOCOL $PORT $DESALG $IVLEN $DESKEYLEN
$DESKEY $DESKEYLEN $DESKEY $AHALG $AHKEYLEN $AHKEY
$AHKEYLEN $AHKEY $AHRESULTLEN $COMBINED_MODE
$DYNAMIC_FLAG
else
     ./sadb delete dst $IPADDRESS $LOCAL-SPI
fi
}

#    Get down to work:
addsa 500 172.17.128.115         # number6.sctc.com
```

The current status of in-kernel SADB 54 can be obtained with the `sadb` command. The get option allows dumping the entire SADB or a single entry. In one embodiment, the complete dump approach uses `/dev/kmem` to find the information. The information may be presented as follows:

```
# sadb get dst

Local-SPI Address-Family Destination-Addr
Preflx_length UID
     Peer-Address Peer-SPI Transport-Type
     Local-Address Real-Address
     Protocol Port
     ESP_Alg_ID ESP_IVEC_Length
          ESP_Enc_Key_length ESP_Enc_ESP_Key
          ESP_Dec_Key_length ESP_Dec_ESP_Key
     AH_Alg_ID AH_Data_Length
          AH_Gen_Key_Length AH_Gen_Key
          AH_Check_Key_Length AH_Check_Key
     Combined_Mode  Dynamic_Flag
```

```
--------------------------------------------------------------------------
-
500 INET: number6.sctc.com 0 0
     0.0.0.0   500 Transport(0) 0
     0.0.0.0 0.0.0.0
     None None
     DES/CBC-RFC1829(1) 4
          8 0b0b0b0b0b0b0b0b
          8 0b0b0b0b0b0b0b0b
     MD5-RFC1828(1) 4
          16 30313233343536373031323334353637
          16 30313233343536373031323334353637
     ESP+AH(1) 0
501 INET: spokes.sctc.com 0 0
     0.0.0.0   501 Transport(0) 0
     0.0.0.0.0.0.0.0
     None None
     DES/CBC-RFC1829(1) 4
          8 0b0b0b0b0b0b0b0b
          8 0b0b0b0b0b0b0b0b
     MD5-RFC1828(l) 4
          16 30313233343536373031323334353637
          16 30313233343536373031323334353637
     ESP+AH(1) 0

End of list.
```

When a new entry is added to in-kernel SADB 54, the add process first checks to see that no existing entry will match the values provided in the new entry. If no match is found then the entry is added to the end of the existing SADB list.

To illustrate the use and administration of an FFE, we'll go through an example using FFE 70 in Figure 3. Firewalls 14 and 18 are both application level gateway firewalls implemented according to the present invention. Workstations H2 and H3 both want to communicate with Hl. For the administrator of firewalls 14 and 18, this is easy to accomplish. The administrator sets up a line something like this (we'll only show the IP address part and SPI parts of the SA, since they're the trickiest values to configure. Also, assume that we are using tunnel mode):

```
#  Hypothetical SW1 Config File
```

```
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
realsrcaddr= localaddr= key=

# The following entry sets up a tunnel between hosts
behind SW1
# and hosts behind SW2.
src=172.16.0.0 localSPI=666 peer=192.168.100.5
peerSPI=777 \
     realsrcaddr=192.168.100.5 localaddrs=0.0.0.0
     key=0xdeadbeeffadebabe

#  Hypothetical SW2 Config File
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
     realsrcaddr= localaddr= key=

#  The following entry sets up a tunnel between hosts
behind SW1 and
#  hosts behind SW2.
src=172.17.0.0 localSPI=777 peer=192.168.20.1
peerSPI=666 \
     realsrcaddr=192.168.20.1 localaddr=0.0.0.0 \
     key=Oxdeadbeeffadebabe
```

With this setup, all traffic is encrypted using one key, no matter who is talking to whom. For example, traffic from H2 to Hl as well as traffic from H3 to Hl will be encrypted with one key. Although this setup is small and simple, it may not be enough.

What happens if H2 cannot trust H3? In this case, the administrator can set up security associations at the host level. In this case, we have to rely on the SPI field of the SA, since the receiving firewall cannot tell from the datagram header which host behind the sending firewall sent the packet. Since the SPI is stored in IPSEC datagrams, we can do a lookup to obtain its value. Below are the sample configuration files for both firewalls again, but this time, each host combination communicates with a different key. Moreover, H2 excludes H3 from communications with Hl, and H3 excludes H2 in the same way.

```
#  Hypothetical SW1 Config File
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
realsrcaddr= localaddr= key=

# The following entry sets up a secure link between H2
and H1
src=172.16.0.2 localSPI=666 peer=192.168.100.5
peerSPI=777 \
     realsrcaddr=192.168.100.5
localaddrs=178.17.128.71 \
     key=0x0a0a0a0a0a0a0a0a

# The following entry sets up a secure link between H3
and H1
src=172.16.0.1 localSPI=555 peer=192.168.100.5
peerSPI=888 \
     realsrcaddr=192.168.100.5
localaddrs=178.17.128.71 \
     key=0x0b0b0b0b0b0b0b0b

#  Hypothetical SW2 Config File
#
#  Fields are laid out in the following manner:
#  srcaddrornet= localSPI= peeraddr= peerSPI=
realsrcaddr= localaddr= key=

# The following entry sets up a secure link between H2
and H1
src=172.17.128.71 localSPI=777 peer=192.168.20.1
peerSPI=666 \
     realsrcaddr=192.168.20.1 localaddrs=172.16.0.2 \
     key=0x0a0a0a0a0a0a0a0a

# The following entry sets up a secure link between H3
and H1
src=172.17.128.71 localSPI=888 peer=192.168.20.1
peerSPI=555 \
     realsrcaddr=192.168.20.1 localaddrs=172.16.0.1 \
     key=0x0b0b0b0b0b0b0b0b
```

Figure 4 is a block diagram showing in more detail one embodiment of an IPSEC-enabled application level gateway firewall 18. Application level gateway firewall 18 provides access control checking of both encrypted and unencrypted messages in a more secure environment due to its network-separated architecture. Network separation divides a system into a set of independent regions or burbs, with a domain and a protocol stack assigned to each burb. Each protocol stack 40x has its own independent set of data structures, including routing information and protocol information. A given socket will be bound to a single protocol stack at creation time and no data can pass between protocol stacks 40 without going through proxy space. A proxy 50 therefore acts as the go-between for transfers between domains. Because of this, a malicious attacker who gains control of one of the regions is prevented from being able to compromise processes executing in other regions. Network separation and its application to an application level gateway is described in "SYSTEM AND METHOD FOR ACHIEVING NETWORK SEPARATION", U.S. Application No. 08/599,232, filed February 9, 1996 by Gooderum et al.

In the system shown in Figure 4, the in-bound and out-bound datagram processing of a security association continues to follow the conventions defined by the network separation model. Thus all datagrams received on or sent to a given burb remain in that burb once decrypted. In one such embodiment SADB socket 78 has been defined to have the type 'sadb'. Each proxy 50 that requires access to SADB socket 78 to execute its query as to whether the received message was encrypted must have create permission to the sadb type.

The following is list of specific requirements that a system such as is shown in Figure 4 must provide. Many of the requirements were discussed in the information provided earlier in this document.

1. Firewall applications may query the IPSEC subsystem to determine if traffic with a given address is guaranteed to be encrypted.
2. Receipt of an unencrypted datagram from an address that has a SA results in the datagram being dropped and an audit message being generated.
3. On receipt of encrypted protocol datagrams the SADB searches will be done using the SPI as the primary key. The source address will a secondary key. The SA returned by the search will be the SA which matches the SPI exactly and has the longest match with the address.

4. A search of the SADB for a SPI that finds an entry that is marked as SA for a dynamic IP will not consider the address in the search process.
5. A search of the SADB for a SPI that finds an entry that is marked as a SA for a tunnel mode connection will to consider the address if it is (0.0.0.0) i.e INADDR.
6. On receipt of a non-IPSEC datagram the SADB will be searched for an entry that matches the src address. If a SA is found the datagram will be dropped and an audit message sent.
7. SADB searches on output will be done using the DST address as key. If more than one SA entry in the SADB has that address the first one with the maximum address match will be returned.
8. The SADB must be structured so that searches are fast regardless if the search is done by SPI or by address.
9. The SADB must provide support for connections to a site with a fixed SPI but changing IP address. SA entries for such connections will be referred to as Dynamic Address Sites, or just Dynamic entries.
10. When a dynamic entry is found by a SPI search, the current datagram's SRC address, which is required to ensure that the return datagrams are properly encrypted, will be recorded in the SA only after the AH checking has passed successfully. (This is because if the address is recorded before AH passes then an attacker can cause return packets of an outgoing connection to be transmitted in the clear.)
11. A failure of an AH check on a dynamic entry results in an audit message.
12. In an embodiment where the firewall requires that all connections use both AH and ESP, on receipt the order should be AH first ESP second.
13. The processing structure on both input and output should try to minimize the number of SADB required lookups.

Returning to Figure 4, in one embodiment firewall 18 includes a crypto engine interface 80 used to encrypt an IPSEC payload. Crypto engine interface 80 may be connected to a software encryption engine 82 or to a hardware encryption engine 84. Engines 82 and 84 perform the actual encryption function using, for example, DES-CBC. In addition, software encryption engine 82 may include the keyed MD5 algorithm used for AH.

In one embodiment, crypto engine interface 80 is a utility which provides a consistent interface between the software and hardware encryption engines. As shown in Figure 4, in one such embodiment interface 80 only supports the use of the use of hardware cryptographic engine 84 for IPSEC ESP processing. The significant design issue that interface 80 must deal with is that use of a hardware encryption engine requires that the processing be down in disjoint steps operating in different interrupt contexts as engine 84 completes the various processing steps.

The required information is stored in a request structure that is bound to the IP datagram being processed. The request is of type `crypto_request_t`. This structure is quite large and definitely does not contain a minimum state set.

In addition to the definition of the request data structure, this software implementing interface 80 provides two functions which isolate the decision of which cryptographic engine to use. The `crypt_des_encrypt` function is for use by the IP output processing to encrypt a datagram. The `crypt_des_decrypt` function is for use by the IP input processing to decrypt a datagram. If hardware encryption engine 84 is present and other hardware usage criteria are met the request is enqueued on a hardware processing queue and a return code indicating that the cryptographic processing is in progress is returned. If software engine 82 is used, the return code indicates that the cryptographic processing is complete. In the former case, the continuation of the IP processing is delayed until after hardware encryption is done. Otherwise it is completed as immediately in the same processing stream.

There are two software cryptographic engines 82 provided in the IPSEC software. One provides the MD5 algorithm used by the IPSEC AH processing, and the other provides the DES algorithm used by the IPSEC ESP processing. This software can be obtained from the US Government IPSEC implementation.

In one embodiment hardware cryptographic engine 84 is provided by a Cylink SafeNode processing board. The interface to this hardware card is provided by the Cylink device driver. A significant aspect of the Cylink card that plays a major part in the design of the IPSEC Cylink driver is that the card functions much like a low level subroutine interface and requires software support to initiate each processing step. Thus to encrypt or decrypt an individual datagram there are a minimum of two steps, one to set the DES initialization vector and one to do the encryption. Since the IP processing can not suspend itself and wait while the hardware completes and then be rescheduled by the hardware interrupt handler, in one embodiment a finite state machine is used to tie sequences of hardware processing elements together. In one such embodiment the interrupt handler looks at the current state, executes a defined after state function, transitions to the state and then executes that state's start function.

One function, `cyl_enqueue_request`, is used to initiate either an encrypt or a decrypt action. This function is designed to be called by cryptographic engine interface 80. All of the information required to initiate the processing as well as the function to be performed after the encryption operation is completed is provided in the request structure. This function will enqueue the request on the hardware request queue and start the hardware processing if necessary.

A system 30 which can be used for firewall-to-workstation encryption is shown in Figure 5. In Figure 5, system 30 includes a workstation 12 communicating through a firewall 14 to an unprotected network 16 such as the Internet. System 30 also includes a workstation 32 communicating directly with firewall 14 through unprotected network 16. Firewall 14 is an application level gateway incorporating IPSEC handling as described above. (It should be noted that IPSEC security cannot be used to authenticate the personal identity of the sender for a firewall to firewall transfer. When IPSEC is used, however, on a single user machine such as a portable personal computer, IPSEC usage should be protected with a personal identification number (PIN). In these cases IPSEC can be used to help with user identification to the firewall.)

According to the IPSEC RFC's, you can use either tunnel or transport mode with this embodiment based on your security needs. In certain situations, the communications must be sent in tunnel mode to hide unregistered addresses.

Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiment shown. This application is intended to cover any adaptations or variations of the present invention. Therefore, it is intended that this invention be limited only by the claims and the equivalents thereof.

What is claimed is:

1. A method of regulating the flow of messages through a firewall having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:

determining, at the IP layer, if a message is encrypted;

if the message is not encrypted, passing the unencrypted message up the network protocol stack to an application level proxy; and

if the message is encrypted, decrypting the message and passing the decrypted message up the network protocol stack to the application level proxy, wherein the step of decrypting the message includes the step of executing a procedure at the IP layer to decrypt the message.

2. A method of authenticating the sender of a message within a computer system having a network protocol stack, wherein the network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:

determining, at the IP layer, if the message is encrypted;

if the message is encrypted, decrypting the message, wherein the step of decrypting the message includes the step of executing a process at the IP layer to decrypt the message;

passing the decrypted message up the network protocol stack to an application level proxy;

determining an authentication protocol appropriate for the message; and

executing the authentication protocol to authenticate the sender of the message.

3. The method according to claim 2 wherein the step of determining an authentication protocol appropriate for the message includes the steps of:

determining a source IP address associated with the message; and

determining the authentication protocol associated with the source IP address.

4. The method according to claim 2 wherein the message includes security parameters index and wherein the step of determining an authentication protocol appropriate for the message includes the steps of:

determining the authentication protocol associated with a dynamic IP address, wherein the step of determining the authentication protocol includes the step of looking up a security association based on the security parameters index;

determining a current address associated with the dynamic source IP address; and

binding the current address to the security parameters index.

5. A firewall, comprising:

a first communications interface;

a second communications interface;

a network protocol stack connected to the first and the second communications interfaces, wherein the network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

a decryption procedure, operating at the IP layer, wherein the decryption procedure decrypts encrypted messages received at one of said first and second communications interfaces and outputs decrypted messages; and

a proxy, connected to the transport layer of said network protocol stack, wherein the proxy receives decrypted messages from the decryption procedure and executes an authentication protocol based on the content of the decrypted message.

6. A firewall, comprising:

a first communications interface;

a second communications interface;

a first network protocol stack connected to the first communications interface, wherein the first network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

a second network protocol stack connected to the second communications interface, wherein the second network protocol stack includes an Internet Protocol (IP) layer and a transport layer;

a decryption procedure, operating at the IP layer of the first network protocol stack, the decryption procedure receiving encrypted messages received by said first communications interface and outputting decrypted messages; and

a proxy, connected to the transport layers of said first and second network protocol stacks, the proxy receiving decrypted messages from the decryption procedure and executing an authentication protocol based on the content of the decrypted message.

7. The firewall according to claim 6 wherein the firewall further includes:

a third communications interface; and

a third network protocol stack connected to the third communications interface and to the proxy, wherein the third network protocol stack includes an Internet Protocol (IP) layer and a transport layer and wherein the second and third network protocol stacks are restricted to first and second burbs, respectively.

8. A method of establishing a virtual private network between a first and a second network, wherein each network includes an application level gateway firewall which uses a proxy operating at the application layer to process traffic through the firewall, wherein each firewall includes a network protocol stack and wherein each network protocol stack includes an Internet Protocol (IP) layer, the method comprising the steps of:

transferring a connection request from the first network to the second network;

determining, at the IP layer of the network protocol stack of the second network's firewall, if the connection request is encrypted;

if the connection request is encrypted, decrypting the request, wherein the step of decrypting the request includes the step of executing a procedure at the IP layer of the second network's firewall to decrypt the message;

passing the connection request up the network protocol stack to an application level proxy;

determining an authentication protocol appropriate for the connection request;

executing the authentication protocol to authenticate the connection request; and

if the connection request is authentic, establishing an active connection between the first and second networks.

9. The method according to claim 8 wherein the step of executing the authentication protocol includes the step of executing program code within the firewall of the second network to mimic a challenge/response protocol executing on a server internal to the second network.

10. The method according to claim 8 wherein the step of executing the authentication protocol includes the step of executing program code to execute the authentication protocol in line to the session.

11. The method according to claim 8 wherein the step of determining an authentication protocol includes the step of determining if the connection request arrived encrypted and selecting the authentication protocol based on whether the connection request was encrypted or not encrypted.

The Patent Office

28

Application No:	GB 9719816.2	**Examiner:**	B.J.SPEAR
Claims searched:	1-11	**Date of search:**	21 January 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): H4P (PPEB,PDCSA,PDCSC)

Int Cl (Ed.6): H04L 9/00, 9/32, 29/06, 29/08

Other: Online:WPI, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
XP	WO97/26734A1	(Raptor Systems) Whole document, eg Figs 1,3 and pages 6-12	1,2.5,6,8 at least
XP	WO97/26731A1	(Raptor Systems) Whole document, eg Figs 1,3 and pages 7-12	1,2.5,6,8 at least
XP	WO97/26735A1	(Raptor Systems) Whole document, eg Figs 1,3 and pages 4-10	1,2.5,6,8 at least
XP	WO97/23972A1	(V-ONE Corp) Whole document, eg Figs 1,2 and claim 1.	1,2.5,6,8 at least
XP	WO97/13340A1	(Digital Secured Networks) Whole document, eg pages 7-13	1,2.5,6,8 at least

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.

Electronic Acknowledgement Receipt

EFS ID:	1304439
Application Number:	10259494
International Application Number:	
Confirmation Number:	5257
Title of Invention:	Agile network protocol for secure communications with assured system availability
First Named Inventor/Applicant Name:	Edmund Colby Munger
Customer Number:	22907
Filer:	Steve S. Chang
Filer Authorized By:	
Attorney Docket Number:	000479.00082
Receipt Date:	09-NOV-2006
Filing Date:	30-SEP-2002
Time Stamp:	17:44:47
Application Type:	Utility

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)	Multi Part /.zip	Pages (if appl.)
1	Information Disclosure Statement (IDS) Filed	00082IDS.pdf	139042	no	5
Warnings:					

Information:					
This is not an USPTO supplied IDS fillable form					
2	NPL Documents	wellsref.pdf	92941	no	1
Warnings:					
Information:					
3	Foreign Reference	WO9827783.pdf	851073	no	24
Warnings:					
Information:					
4	Foreign Reference	GB2334181.pdf	434487	no	15
Warnings:					
Information:					
5	Foreign Reference	EP0814589.pdf	1098918	no	20
Warnings:					
Information:					
6	Foreign Reference	EP0838930.pdf	1731451	no	35
Warnings:					
Information:					
7	Foreign Reference	GB2317792.pdf	1262522	no	35
Warnings:					
Information:					
8	NPL Documents	Davilaarticle.pdf	807564	no	18
Warnings:					
Information:					
Total Files Size (in bytes):			6417998		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

MAR 3 0 2007

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Edmund Colby Munger et al.
Serial Number:	10/259,494
Filing Date:	30 September 2002
Title:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST (As amended)
Examiner:	Krisna Lim.
Art Unit:	2153
Docket No.:	77580-012 (VRN- 1DV3)
Confirmation No.:	5257

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted (571-273-8300) to the US PTO, on the date indicated below.

Date: 30 March 2007

Erin M. Olson

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT "C"

This paper is in response to the non-final Office Action mailed 30 October 2006 for the above-identified application.

The Applicants appreciate the Examiner's thorough examination of the subject application, and request reconsideration and further examination in view of the following:

Amendments to the Claims, in the claim listing beginning on page 2 of this paper; and

Remarks, beginning on page 12 of this paper.

AMENDMENTS TO THE CLAIMS

Please amend claims 1, 5-7, 9, 20, 22-26, 33 and 34, cancel claims 15, 29 and 30 without prejudice, and add new Claims 35-68, as shown below. This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for establishing an encrypted channel between a client and a target~~-computer~~, comprising the steps of:

(i) ~~intercepting~~generating a DNS request ~~sent by~~from the client; and

(ii) based on the DNS request, automatically establishing the encrypted channel between the client and the target,
wherein the encrypted channel is capable of supporting services.

2. (Previously Presented) The method of claim 1, wherein step (ii) comprises steps of:

a) determining whether the client is authorized to access the target identified by the DNS request;

b) when the client is authorized to access the target, initiating the encrypted channel; and

c) when the client is not authorized to access the target, sending an error message to the client.

3. (Original) The method of claim 2, wherein step b) comprises sending encrypted channel parameters to the client.

4. (Original) The method of claim 1, wherein step (ii) occurs in a communication protocol independently of an application program.

5. (Currently Amended) The method of claim 1, wherein ~~step (i)~~the method further comprises~~ a DNS proxy server~~ intercepting the DNS request ~~sent by the client~~using a DNS proxy server.

6. (Currently Amended) The method of claim 1, wherein step (ii) comprises establishing the encrypted channel responsive to ~~intercepting a~~the DNS request for a domain name comprising a predetermined domain name extension.

7. (Currently Amended) A method for establishing an encrypted channel between a client and a secure ~~host~~device, comprising the step of automatically creating the encrypted channel ~~upon intercepting~~as a response to a DNS request for a domain name~~ comprising a predetermined domain name extension~~, wherein the encrypted channel is capable of supporting services.

8. (Original) The method of claim 7, wherein the creating step is performed in a communication protocol independently of an application program.

9. (Currently Amended) A method for establishing an encrypted channel between a client and a secure ~~host~~device, comprising the step of automatically creating the encrypted channel in response to ~~detecting~~ a request for access to ~~a predetermined~~an IP address, wherein the encrypted channel is capable of supporting services.

10. (Original) The method of claim 9, wherein the creating step is performed in a communication protocol independently of an application program.

11. (Original) A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:

(i) determining whether the intercepted DNS request corresponds to a secure server;

(ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer, and

(iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

12. (Original) The data processing device of claim 11, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

13. (Original) The data processing device of claim 12, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

14. (Original) The data processing device of claim 13, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

15. (Canceled)

16. (Original) A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) intercepting a DNS request sent by a client;

(ii) determining whether the intercepted DNS request corresponds to a secure server;

(iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

17. (Original) The computer readable medium of claim 16, wherein step (iii) comprises the steps of

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

18. (Original) The computer readable medium of claim 17, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

19. (Original) The computer readable medium of claim 18, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

20. (Currently Amended) A computer readable medium comprising computer readable instructions that, when executed, ~~cause a domain name server (DNS) proxy module to intercept DNS requests sent by a client and, for each intercepted DNS request, when the intercepted~~perform steps of: when a DNS request corresponds to a secure ~~server, determines~~device, determining whether ~~the~~a client is authorized to access the secure ~~server~~device and, if so, automatically ~~initiates~~initiating an encrypted channel between the client and the secure ~~server~~device, wherein the encrypted channel is capable of supporting services.

21. (Previously Presented) The method of claim 1, wherein step (ii) comprises establishing an IP address hopping scheme between the client and the target.

22. (Currently Amended) The method of claim 1, wherein step (ii) ~~avoids sending~~comprises concealing a true IP address of the target ~~to the client~~.

23. (Currently Amended) The method of claim 7, wherein creating the encrypted channel comprises establishing an IP address hopping scheme between the client and the secure ~~host~~device.

24. (Currently Amended) The method of claim 7, wherein creating the encrypted channel avoids sendingcomprises concealing a true IP address of the secure host to the clientdevice.

25. (Currently Amended) The method of claim 9, wherein automatically creating the encrypted channel comprises establishing an IP address hopping scheme between the client and the secure hostdevice.

26. (Currently Amended) The method of claim 9, wherein automatically creating the encrypted channel avoids sendingcomprises concealing a true IP address of the secure host to the clientdevice.

27. (Previously Presented) The data processing device of claim 11, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

28. (Previously Presented) The data processing device of claim 11, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

29. (Canceled)

30. (Canceled)

31. (Previously Presented) The computer readable medium of claim 16, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

32. (Previously Presented) The computer readable medium of claim 16, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

33. (Currently Amended) The computer readable medium of claim 20, wherein automatically initiating the encrypted channel between the client and the secure ~~server~~device comprises establishing an IP address hopping scheme between the client and the secure ~~server~~device.

34. (Currently Amended) The computer readable medium of claim 20, wherein automatically initiating the encrypted channel between the client and the secure ~~server avoids sending~~device comprises concealing a true IP address of the secure ~~server to the client~~device.

35. (New) The method of claim 1, wherein the encrypted channel uses tunneling.

36. (New) The method of claim 1, wherein the encrypted channel supports various communication protocols.

37. (New) The method of claim 1, wherein the encrypted channel supports a plurality of application programs.

38. (New) The method of claim 1, wherein the encrypted channel supports multiple sessions.

39. (New) The data processing device of claim 11, wherein the data processing device comprises the client.

40. (New) A method for establishing a secure communication link between a first device and a second device, the method comprising steps of: generating a request for access to a second device and automatically initiating a secure communication link between a first device and the second device, wherein the secure communication link is capable of supporting a plurality of services.

41. (New) The method of claim 40, wherein the first device has an IP address, and the second device has an IP address.

42. (New) The method of claim 40, wherein the request uses a DNS request.

43. (New) The method of claim 40, wherein the secure communication link is automatically initiated in response to the request.

44. (New) The method of claim 40, wherein the request is a DNS request that requests an IP address associated with the second device.

45. (New) The method of claim 40, wherein the step of automatically initiating a secure communication link is performed in a communication protocol independently of an application program.

46. (New) The method of claim 40 further comprising a step of automatically establishing the secure communication link between the first device and the second device.

47. (New) The method of claim 40 further comprising a step of automatically establishing the secure communication link between the first device and the second device,

wherein the first device has an IP address, the second device has an IP address, the second device is a secure device, and the secure communication link is automatically initiated in response to the request.

48. (New) The method of claim 40, wherein each of the first device and the second device comprises a computer.

49. (New) The method of claim 40, wherein the secure communication link uses encryption.

50. (New) The method of claim 40, wherein the first device sends a tunneled packet.

51. (New) The method of claim 40, wherein the secure communication link uses tunneling.

52. (New) The method of claim 40, wherein the secure communication link supports various communication protocols.

53. (New) The method of claim 40, wherein the secure communication link supports a plurality of application programs.

54. (New) The method of claim 53, wherein the plurality of application programs comprises video conferencing.

55. (New) The method of claim 53, wherein the plurality of application programs comprises e-mail.

56. (New) The method of claim 53, wherein the plurality of application programs comprises a word processing program.

57. (New) The method of claim 53, wherein the plurality of application programs comprises telephony.

58. (New) The method of claim 53, wherein the plurality of application programs comprises video.

59. (New) The method of claim 53, wherein the plurality of application programs comprises audio.

60. (New) The method of claim 53, wherein the plurality of application programs comprises items selected from a group consisting of the following: video conferencing, e-mail, a word processing program, and telephony.

61. (New) The method of claim 40, wherein the secure communication link supports multiple sessions.

62. (New) The method of claim 40, wherein the secure communication link supports the plurality of services.

63. (New) The method of claim 40, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof.

64. (New) The method of claim 40, wherein the secure communication link is an authenticated link.

65. (New) A machine readable medium comprising machine readable instructions for performing steps of: generating a request for access to a second device and automatically initiating a secure communication link between a first device and the second device, wherein the secure communication link is capable of supporting a plurality of services.

66. (New) A device comprising memory, the device for performing a method for creating a secure communication link, the method comprising steps of: generating a request for access to a second device and automatically initiating a secure communication link between the device and the second device, wherein the secure communication link is capable of supporting a plurality of services.

67. (New) A device comprising memory, the device for performing the steps of:

receiving information as to whether a DNS request is associated with a secure device; and

when the DNS request is associated with a secure device, automatically initiating a secure communication link with the secure device,

wherein the secure communication link is capable of supporting a plurality of services.

68. (New) The device of claim 66, wherein the device is for performing a step of: when the DNS request is not associated with a secure device, receiving an IP address of a nonsecure device.

REMARKS

Claims 1-14, 16-28 and 31-68 remain in the application. Claims 1, 5-7, 9, 20, 22-26, 33 and 34 are currently amended, claims 35-68 added as new claims, and claims 15, 29 and 30 canceled.

Claims 1, 7 and 9 have been rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject mater which applicant regards as the invention. Claims [1-6] have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Sistanizadeh et al. (US Patent No 6101182). Claims 7-10, 15 and 20 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over Sistanizadeh et al. (US Patent No 6101182). Applicants acknowledge with appreciation that claims 11-14, 16-19, 27, 28, 31 and 32 are allowable, while claims 21-26, 29-30, 33 and 34 are objected to, but presumably would be allowable if rewritten in independent form. The rejections and objection are respectfully traversed and reconsideration is requested in view of the foregoing amendments and following remarks.

With regard to the rejection of claims 1, 7 and 9 under 35 U.S.C. § 112, second paragraph, the Examiner states that it is not clearly understood in claim 1 how the encrypted channel is established between the client and the target based on the DNS request, in claim 7 how the encrypted channel is created automatically upon intercepting a DNS request for a domain name comprising a predetermined domain name extension, and in claim 9 how the encrypted channel is created automatically in response to detecting a request for access to a predetermined IP address. Claims 1 has been amended to recite the steps of generating a DNS request from the client; and based on the DNS request, automatically establishing the encrypted channel between the client and the target. Claim 7 recites the step of automatically creating the encrypted channel as a response to a DNS request for a domain name. Claim 9 recites the step of automatically creating the encrypted channel in response to a request for access to an IP address. In all cases the encrypted channel is capable of supporting services. It is submitted that the claim limitations clearly recite how the encrypted channel is established or created automatically. Such recitations are clearly understood and compliant with 35 U.S.C. § 112, paragraph 2.

With regard to the rejection of claims 1-10, 15 and 20 under 35 U.S.C. § 103, it is submitted that the claims are neither anticipated nor made obvious in view of Sistanizadeh et al. (US Patent No 6101182). Applicants' independent claim 1 recites automatically establishing the encrypted channel between the client and the target. Independent claim 7 recites automatically creating the encrypted channel between a client and a secure device. Independent claim 9 recites automatically creating the encrypted channel between a client and a secure device. Independent claim 15 has been canceled. Independent claim 20 recites automatically initiating an encrypted channel between the client and the secure device.

Sistanizadeh et al. (US Patent No 6101182) is a continuation of the previously cited reference to Sistanizadeh et al. (US Patent No 5790548), and thus offers nothing different from this previously cited reference. The Examiner believe that the reference teaches both "a) intercepting a DNS request sent by the client (e.g. see Figs. 6 and 8b), and b) based on the DNS request, establishing the encrypted channel between the client and the target (e.g., see Figs. 6 and 8b, col. 2, lines 43-44, col. 6, lines 62-64, col. 7, lines 33-35, col. 10, lines 62-67).", but then admits that the reference does not "explicitly mention that the encrypted channel is established between the client and the target." He nevertheless concludes that "it would be obvious to one of ordinary skill in the art to recognize that Sistanizadeh's teaching of encryption and authentication between the server and the customer computer would obviously teach the feature of establishing the encrypted channel between the client and the target." It is submitted that the reference is not interested in establishing an encrypted channel between a client and a target. The reference is generally related to methods and systems for providing public and private access to on-line multimedia services through the public telecommunications system. The reference describes how the internet may be viewed as a series of routers connected together with computers connected to the routers. Each machine on the internet has a unique number which comprises four sets of binary numbers, with one of the sets identifying the destination router. Packets from a source router are directed along a path determined by tables contained along the routing path. But identifying the best path for sending packets does not lead one to conclude that one establishes an encrypted channel between a client and a target. The Examiner's reference to column 7, liens 33-35 that "Ethernet switch bridges the user information frames to its output port" and that "this function is needed to prevent the information of one user from being "looped" back to another user for security and privacy considerations" is not a teaching of establishing an encrypted channel between a client and a target.

Dependent claims 2-6, 8, and 10 currently under consideration in the application are dependent from independent Claim 1, 7, or 9 discussed above and therefore are believed to be allowable over the applied references for at least the same reasons. Because each dependent claim is deemed to define an additional aspect of the invention, the individual consideration of each on its own merits is respectfully requested.

Newly added claims 35-68 are also believed patentable over the reference.

Accordingly, all of the pending claims, claims 1-14, 16-28 and 31-68, are believed to be patentable over the cited reference. An early and favorable action thereon is therefore earnestly solicited.

Authorization is hereby given to charge our deposit account, No. 50-1133, for the fees corresponding to a Petition for Extension of Time (two-months) under 37 CFR § 1.136, and for any other fees that may be required for the prosecution of the subject application.

If a telephone conference will expedite prosecution of the application, the Examiner is invited to telephone the undersigned.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Date: 30 March 2007

Toby H. Kusmer, P.C.
Reg. No. 26,418
Attorney for Applicants
28 State Street
Boston, MA 02109-1775
DD Telephone: (617) 535-4065
Facsimile: (617)535-3800
e-mail: tkusmer@mwe.com

BST99 1536399-1.077580.0012

MAR 3 0 2007

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Edmund Colby Munger et al.
Serial Number:	10/259,494
Filing Date:	30 September 2002
Title:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST (As amended)
Examiner:	Krisna Lim.
Art Unit:	2153
Docket No.:	77580-012 (VRN- 1DV3)
Confirmation No.:	5257

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted (571-273-8300) to the US PTO, on the date indicated below.

Date: 30 March 2007

Erin M. Olson

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION FOR TWO-MONTH EXTENSION OF TIME

Dear Sir:

A two-month extension of time to and including 30 March 2007, is requested for a response to the non final Office Action dated 30 October 2006. If a two-month extension of time is insufficient, Applicant requests such an additional extension as is necessary to maintain the pendency of the application.

The Commissioner is hereby authorized to charge the extension fee of $450.00, as well as any additional fees which may be due, or credit any overpayment, to Deposit Account Number 50-1133.

Respectfully submitted,

Date: 30 March 2007

Toby H. Kusmer, Reg. No.: 26,418
McDERMOTT WILL & EMERY LLP
28 State Street
Boston, Massachusetts 02109
Telephone: (617) 535-4082
Facsimile: (617) 535-3800

04/02/2007 FMETEKI1 00000045 501133 10259494
01 FC:1252 450.00 DA

BST99 1536418-1,077580.0012

McDermott
Will&Emery

Boston Brussels Chicago Düsseldorf London Los Angeles Miami Milan
Munich New York Orange County Rome San Diego Silicon Valley Washington, D.C.

FACSIMILE

Date: 30 March 2007 **Time Sent:**

To:	Company:	Facsimile No:	Telephone No:
	USPTO	571-273-8300	

From:	Toby H. Kusmer	*Direct Phone:*	617.535.4037
E-Mail:	tkusmer@mwe.com		
Sent By:	Erin M. Olson	*Direct Phone:*	617.535.3953
Client/Matter/Tkpr:	0077580-0012-08657		No
		Number of Pages, Including Cover:	16

Message:

Main Facsimile: 617.535.3800 Facsimile Operator: 617.535.4000

U.S. practice conducted through McDermott Will & Emery
28 State Street Boston, Massachusetts 02109-1775 Telephone: 617.535.4000

PTO/SB/06 (12-04)
Approved for use through 7/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 10/859,494

APPLICATION AS FILED – PART I

FOR	NUMBER FILED (Column 1)	NUMBER EXTRA (Column 2)	SMALL ENTITY RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))							
SEARCH FEE (37 CFR 1.16(k), (i), or (m))							X
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))							
TOTAL CLAIMS (37 CFR 1.16(i))	34 minus 20 =	*0	X$ 25=		OR	X$50=	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	7 minus 3 =	*0	X$100=			X$200=	
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR						
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))			N/A			N/A	
			TOTAL			TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

3/30/07

AMENDMENT A

	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 65	Minus	** 34	= 31	X =		OR	x 50 =	1550
Independent (37 CFR 1.16(h))	* 10	Minus	*** 7	= 3	X =		OR	x 200 =	600
Application Size Fee (37 CFR 1.16(s))									
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					N/A		OR	N/A	
					TOTAL ADD'T FEE		OR	TOTAL ADD'T FEE	2150

AMENDMENT B

	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	*	Minus	**	=	X =		OR	X =	
Independent (37 CFR 1.16(h))	*	Minus	***	=	X =		OR	X =	
Application Size Fee (37 CFR 1.16(s))									
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))					N/A		OR	N/A	
					TOTAL ADD'T FEE		OR	TOTAL ADD'T FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

PATENT APPLICATION FEE DETERMINATION RECORD

Effective December 8, 2004

10/259,494

CLAIMS AS FILED - PART I

CLAIMS AS FILED - PART I	(Column 1)	(Column 2)	SMALL ENTITY TYPE ☐ RATE	FEE	OR	OTHER THAN SMALL ENTITY RATE	FEE
TOTAL CLAIMS							
FOR	NUMBER FILED	NUMBER EXTRA	BASIC FEE	150.00	OR	BASIC FEE	~~300.00~~ 740
TOTAL CHARGEABLE CLAIMS	20 minus 20=	*	X$ 25=		OR	X$50=	
INDEPENDENT CLAIMS	6 minus 3 =	*	X100=		OR	~~X200=~~ 84	252
MULTIPLE DEPENDENT CLAIM PRESENT		☐	+180=		OR	+360=	
			TOTAL		OR	TOTAL	992

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

9/13/04

AMENDMENT A	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE	ADDITIONAL FEE	OR	OTHER THAN SMALL ENTITY RATE	ADDITIONAL FEE
Total	* 34	Minus	** 20	= 14	X$ 25=		OR	~~X$50=~~ 18	252
Independent	* 7	Minus	*** 6	= 1	X100=		OR	~~X200=~~	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐	+180=		OR	+360=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	252

1/6/06

AMENDMENT B	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
Total	* 34	Minus	** 34	= 0	X$ 25=		OR	X$50=	
Independent	* 7	Minus	*** 7	= 0	X100=		OR	X200=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐	+180=		OR	+360=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

7/27/06

AMENDMENT C	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
Total	* 34	Minus	** 34	= 0	X$ 25=		OR	X$50=	
Independent	* 7	Minus	*** 7	= 0	X100=		OR	X200=	
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐	+180=		OR	+360=	
					TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20."
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3."
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

FORM PTO-875 (Rev. 10/04) Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 04/10/2007

RLLOYD	SALE	#00000001	Mailroom Dt:	03/30/2007	501133	10259494
		01	FC : 1202	1,550.00 DA		
		02	FC : 1201	600.00 DA		

McDermott
Will&Emery

Boston Brussels Chicago Düsseldorf London Los Angeles Miami Munich
New York Orange County Rome San Diego Silicon Valley Washington, D.C.
Strategic alliance with MWE China Law Offices (Shanghai)

RECEIVED
CENTRAL FAX CENTER
JUN 0 7 2007

FACSIMILE

Date: June 7, 2007 **Time Sent:**

To:	**Company:**	**Facsimile No:**	**Telephone No:**
Commissioner for Patents – Group Art Unit: 2153	U.S. Patent and Trademark Office	1.571.273.8300	

From:	Toby H. Kusmer, P.C.	*Direct Phone:*	617.535.4065
E-Mail:	tkusmer@mwe.com	*Direct Fax:*	617.535.3800
Sent By:	Cynthia Joseph	*Direct Phone:*	617.535.4111
Client/Matter/Tkpr:	77580-012/5496	*Original to Follow by Mail:*	No
		Number of Pages, Including Cover:	

Re: In re Application of: Edmund Colby Munger, et al.
Serial No.: 10/259,494
Filing Date: September 30, 2002
Title: Improvement To An Agile Network Protocol For Secure Communications With Assured System Availability
Docket No.: 77580-012 (VRNK-1CPDV3)

Message:

Please enter the attached Change of Correspondence Address And Revocation Of Power Of Attorney And Appointment of New Attorney.

IF YOU DO NOT RECEIVE ALL OF THE PAGES, PLEASE CALL AS SOON AS POSSIBLE.

Main Facsimile: 617.535.3800 Facsimile Operator: 617.535.4000

U.S. practice conducted through McDermott Will & Emery LLP.
28 State Street Boston, Massachusetts 02109-1775 Telephone: 617.535.4000

RECEIVED
CENTRAL FAX CENTER

JUN 0 7 2007

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of:	Edmund Colby Munger, et al.
Serial No.:	10/259,494
Filing Date:	September 30, 2002
Title:	Improvement To An Agile Network Protocol For Secure Communications With Assured System Availability
Group Art Unit:	2153
Confirmation No.:	5257
Docket No.:	77580-012 (VRNK-1CPDV3)

MAIL STOP Amendment
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER

Applicants transmit herewith the following document in the above-identified application:

Change Of Correspondence Address And Revocation Of Power Of Attorney And Appointment Of New Power.

The Commissioner is authorized to charge any filing fee or credit any overpayment to Deposit Account No. 50-1133.

Respectfully submitted,

Toby H. Kusmer, P.C.
Reg. No. 26,418
McDermott Will & Emery LLP
28 State Street
Boston, MA 02109-1775
Telephone: 617.535.4065
Facsimile: 617.535.3800
e-mail: tkusmer@mwe.com

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted, via Facsimile No. 571.273.8300, to the U.S. Patent and Trademark Office on the date indicated below and is addressed to Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450.

June 7, 2006
Date

Cynthia Joseph

RECEIVED
CENTRAL FAX CENTER
JUN 07 2007

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of:	Edmund Colby Munger, et al.
Serial No.:	10/259,494
Filing Date:	September 30, 2002
Title:	Improvement To An Agile Network Protocol For Secure Communications With Assured System Availability
Docket No.:	77580-012 (VRNK-1CPDV3)

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

CHANGE OF CORRESPONDENCE ADDRESS AND REVOCATION OF POWER OF ATTORNEY AND APPOINTMENT OF NEW ATTORNEY

Dear Sir:

Applicants in the above-identified patent application hereby revoke all powers of attorney previously given in connection with the above-identified patent application and hereby appoint the following attorneys, with full power of substitution, to transact all business in the Patent and Trademark Office connected therewith:

All attorneys associated with CUSTOMER NUMBER 23630

BST99 1533853-1 077580.0012

RECEIVED
CENTRAL FAX CENTER
JUN 07 2007

It is requested that all correspondence regarding this patent application be directed to:

Toby H. Kusmer, P.C.
McDermott Will & Emery LLP
28 State Street
Boston, Massachusetts 02109-1775
Telephone: (617) 535-4065
Facsimile: (617) 535-3800
E-mail: tkusmer@mwe.com

Respectfully submitted,

Date: 5/24/07

By: [signature]
Name: Randall Carsen
Title: President and CEO

BST99 1533853-1.077580.0012

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

22907 7590 06/15/2007

BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

EXAMINER	
LIM, KRISNA	

ART UNIT	PAPER NUMBER
2153	

MAIL DATE	DELIVERY MODE
06/15/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

Office Action Summary	Application No.	Applicant(s)	
	10/259,494	MUNGER ET AL.	
	Examiner	Art Unit	
	Krisna Lim	2153	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *30 March 2007*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14,16-28 and 31-68* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☒ Claim(s) *11-14,16-19,27,28,31,32 and 39* is/are allowed.

6)☒ Claim(s) *1-10,21-26,33-38 and 40-68* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

1. Claims 1-14, 16-28 and 31-39 are still pending for examination, and claims 40-67 are newly presented for examination.

2. The following is a quotation of 35 U.S.C. § 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-10, 20-26 and 34-67 are rejected under 35 U.S.C. § 103(a) as being unpatentable over Sistanizadeh et al. [U.S. Patent No. 6,101,182].

4. Sistanizadeh et al. disclose (e.g., see Figs. 1-22) the invention substantially as claimed. Taking claims 1, 21, 22 and 35-38 as exemplary claims, the reference discloses a method for establishing an encrypted channel between a client and a target computer, comprising the steps of:

a) intercepting a DNS request sent by the client (e.g., see Figs. 6 and 8b).

b) based on the DNS request, establishing the encrypted channel between the client and the target (e.g., see Figs. 6 and 8b, col. 2, lines 43-44, col. 6, lines 62-64, col. 7, lines 33-35, col. 10, lines 62-67).

While Sistanizadeh discloses: a) IP addresses DNS are provided and connected to routers in the network (e.g., see col. 6, lines 62-67); b) the establish session of DNS (e.g., see col. 6, lines 62-67); c) requesting authentication (e.g., see col. 9, lines 65-66), d) encryption and authentication between the server and customer computer (e.g., see col. 10, lines 46-47), Sistanizadeh did not explicitly mention that the encrypted channel is established between the client (a customer computer) and the target (the server). To the extent of the claimed language, one of ordinary skilled in the art would obviously recognize that Sistanizadeb's customer computer is obviously the client while Sistanizadeb's server is obviously the target.

5. As to claim 2, Sistanizadeh further disclosed the steps of: a) determining whether the client is authorized to access the target identified by the DNS request; b) when the client is authorized to access the target, initiating the encrypted channel; and when the client is not authorized to access the target, sending an error message to the client (e.g., see verification and authentication of Fig. 7, encryption and authentication between the server and customer computer at col. 10, lines 46-47).

6. As to claim 3, Sistanizadeh further disclosed the step of sending encrypted channel parameter to the client (e.g., see the functions of router 330 of Fig. 3, encryption and authentication between the server and customer computer (e.g., see col. 10, lines 46-47).

7. As to claim 4, Sistanizadeh further disclosed a communication protocol independently of an application program (e.g., see col. 3, line 52, to col. 4, line 20).

8. As to claim 5, Sistanizadeh further disclosed a DNS proxy server (DHCP server 334, 348 and 350).

9. As to claim 6, the feature of predetermined domain name extension. It would have been obvious to one of ordinary skill in the art to recognize that such predetermined domain name extension (e.g., .gov, .org, etc.) is well known feature in the art.

10. Claims 7-10, 20, 23-26, 33, 34 and 40-67 are rejected for the same rationale as claims 1-6, 21-22 and 35-38, since they recite substantially identical subject matter.

Any differences between the claims do not result in patentably distinct claims and all of the limitations are taught by the above cited references.

11. Claims 11-14, 16-19, 27, 28, 31, 32 and 39 are allowable.

12. Applicant's arguments filed 3/30/07 have been fully considered but they are not deemed to be persuasive.

13. In the remarks, applicants argued in substance that it is submitted that the reference is not interested in establishing an encrypted channel between a client and a target. The reference is generally related to methods and systems for providing public and private access to on-line multimedia services through the public telecommunication system. In reply, Examiner respectfully disagrees because Sistanizadeb discloses a method for establishing an encrypted channel between a client and a target computer (e.g., see Figs. 6 and 8b, col. 2, lines 43-44, col. 6, lines 62-64, col. 7, lines 33-35, col. 10, lines 62-67).

14. A shortened statutory period for response to this action is set to expire 3 (three) months and 0 (zero) days from the mail date of this letter. Failure to respond within the period for response will result in **ABANDONMENT** of the application (see 35 U.S.C 133, M.P.E.P 710.02, 710.02(b)).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Krisna Lim whose telephone number is 571-272-3956 The examiner can normally be reached on Monday to Wednesday and Friday from 7:30 AM to 4:00 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton Burgess, can be reached on 571-272-3949. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

KI

June 11, 2007

KRISNA LIM
PRIMARY EXAMINER

Date 11/09/06

PTO/SB/08a (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10259494
	Filing Date	2002-09-30
	First Named Inventor	Edmund Colby Munger
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00082

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code1	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
KL	1	5870610	A	1999-02-09	Beyda et al.	

If you wish to add additional U.S. Patent citation information please click the Add button.

U.S.PATENT APPLICATION PUBLICATIONS

Examiner Initial*	Cite No	Publication Number	Kind Code1	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button.

FOREIGN PATENT DOCUMENTS

Examiner Initial*	Cite No	Foreign Document Number3	Country Code2 i	Kind Code4	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T5
KL	1	0 838 930	EP	A	1998-04-29	Compaq Computer Corp.		☐
KL	2	0 814 589	EP	A	1997-12-29	AT&T Corp.		☐
KL	3	2 334 181	GB	A	1999-08-11	NEC Technologies; Globalmart Ltd.		☐

EFS Web 2.0

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10259494
	Filing Date	2002-09-30
	First Named Inventor	Edmund Colby Munger
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00082

KL	4	9827783	WO	A	1998-06-25	Northern Telecom Limited; Antonio, G; Hui, Margare		☐
KL	5	2 317 792	GB	A	1998-04-01	Secure Computing Corporation		☐
	6							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button

NON-PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T5
KL	1	Laurie Wells (LANCASTERBIBELMAIL MSN COM); "Subject: Security Icon" USENET NEWSGROUP, 19 October 1998, XP002200606	☐
KL	2	Davila J et al, "Implementation of Virtual Private Networks at the Transport Layer", Information Security, Second International Workshop, ISW '99. Proceedings (Lecture Springer-Verlag Berlin, Germany, [Online] 1999, pages 85-102, XP002399276, ISBN 3-540-66695-B, retrieved from the Internet: URL: http://www.springerlink.com/content/4uac0tb0heccma89/fulltext.pdf> (Abstract)	☐
	3		☐
	4		☐

If you wish to add additional non-patent literature document citation information please click the Add button

EXAMINER SIGNATURE

Examiner Signature	[signature]	Date Considered	6/11/07

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

EFS Web 2.0

Date 11/09/06

PTO/SB/08a (08-03)
Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	10259494
	Filing Date	2002-09-30
	First Named Inventor	Edmund Colby Munger
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00082

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code1	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
KL	1	5870610	A	1999-02-09	Beyda et al.	

If you wish to add additional U.S. Patent citation information please click the Add button.

U.S.PATENT APPLICATION PUBLICATIONS

Examiner Initial*	Cite No	Publication Number	Kind Code1	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button.

FOREIGN PATENT DOCUMENTS

Examiner Initial*	Cite No	Foreign Document Number3	Country Code2 j	Kind Code4	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T5
KL	1	0 838 930	EP	A	1998-04-29	Compaq Computer Corp.		☐
KL	2	0 814 589	EP	A	1997-12-29	AT&T Corp.		☐
KL	3	2 334 181	GB	A	1999-08-11	NEC Technologies; Globalmart Ltd.		☐

EFS Web 2.0

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	10259494
	Filing Date	2002-09-30
	First Named Inventor	Edmund Colby Munger
	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	000479.00082

KL	4	9827783	WO	A	1998-06-25	Northern Telecom Limited; Antonio, G; Hui, Margare		☐
KL	5	2 317 792	GB	A	1998-04-01	Secure Computing Corporation		☐
	6							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button

NON-PATENT LITERATURE DOCUMENTS

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T5
KL	1	Laurie Wells (LANCASTERBIBELMAIL MSN COM); "Subject: Security Icon" USENET NEWSGROUP, 19 October 1998, XP002200606	☐
KL	2	Davila J et al, "Implementation of Virtual Private Networks at the Transport Layer", Information Security, Second International Workshop, ISW '99. Proceedings (Lecture Springer-Verlag Berlin, Germany, [Online] 1999, pages 85-102, XP002399276, ISBN 3-540-66695-B, retrieved from the Internet: URL: http://www.springerlink.com/ content/4uac0tb0heccma89/fulltext.pdf> (Abstract)	☐
	3		☐
	4		☐

If you wish to add additional non-patent literature document citation information please click the Add button

EXAMINER SIGNATURE

Examiner Signature	[signature]	Date Considered	6/11/07

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

EFS Web 2.0

Index of Claims

	Application/Control No.	Applicant(s)/Patent under Reexamination	
	10/259,494	MUNGER ET AL.	
	Examiner	**Art Unit**	
	Krisna Lim	2153	

R	Rejected	—	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Claim		Date							
Final	Original	6/11/07							
	1	R							
	2	R							
	3	R							
	4	R							
	5	R							
	6	R							
	7	R							
	8	R							
	9	R							
	10	R							
	11	=							
	12	=							
	13	=							
	14	=							
	15								
	16	=							
	17	=							
	18	=							
	19	=							
	20	R							
	21	R							
	22	R							
	23	R							
	24	R							
	25	R							
	26	R							
	27	=							
	28	=							
	29								
	30								
	31	=							
	32	=							
	33	R							
	34	R							
	35	R							
	36	R							
	37	R							
	38	R							
	39	=							
	40	R							
	41	R							
	42	R							
	43	R							
	44	R							
	45	R							
	46	R							
	47	R							
	48	R							
	49	R							
	50	R							

Claim		Date							
Final	Original	6/11/07							
	51	R							
	52	R							
	53	R							
	54	R							
	55	R							
	56	R							
	57	R							
	58	R							
	59	R							
	60	R							
	61	R							
	62	R							
	63	R							
	64	R							
	65	R							
	66	R							
	67	R							
	68	R							
	69								
	70								
	71								
	72								
	73								
	74								
	75								
	76								
	77								
	78								
	79								
	80								
	81								
	82								
	83								
	84								
	85								
	86								
	87								
	88								
	89								
	90								
	91								
	92								
	93								
	94								
	95								
	96								
	97								
	98								
	99								
	100								

Claim		Date								
Final	Original									
	101									
	102									
	103									
	104									
	105									
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

Docket No.: 077580-0012 **PATENT**

OIPE NOV 13 2007 PATENT & TRADEMARK OFFICE

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	Edmund C. Munger et al.
Appl. No.	:	10/259,494
Filed	:	September 30, 2002
Title	:	AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY
Grp./A.U.	:	2153
Examiner:	:	LIM, Krisna

Customer No.: 23,630
Confirmation No.: 5257

CERTIFICATE OF MAILING (37 CFR. § 1.8(a))

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as First-Class Mail under 37 CFR 1.8(a) in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on November 9, 2007.

Cynthia Joseph

INFORMATION DISCLOSURE STATEMENT

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

In accordance with the provisions of 37 C.F.R. 1.56, 1.97 and 1.98, the attention of the Patent and Trademark Office is hereby directed to the documents listed on the attached form PTO-1449. It is respectfully requested that the documents be expressly considered during the prosecution of this application, and that the documents be made of record therein and appear among the "References Cited" on any patent to issue therefrom.

This Information Disclosure Statement is being filed before the receipt of a Final Office Action for the above-referenced application. The commissioner is authorized to charge a submission fee of $180.00 to our Deposit Account No. 50-1133.

References C1, C3, C5-C8, and C11 were cited by or submitted to the U.S. Patent and Trademark Office in parent application Serial No. 09/504,783, filed February 15, 2000, now U.S. Patent No. 6,502,135, which is relied upon for an earlier filing date under 35 USC 120. Thus, in accordance with 37 CFR 1.98(d), copies of these references are not attached. Applicants will be pleased to provide copies of the reference if requested by the Examiner.

11/14/2007 TNGUYEN2 00000048 501133 10259494
01 FC:1806 180.00 DA

BST99 1556722-1.077580.0012

This Statement is not to be interpreted as a representation that the cited publications are material, that an exhaustive search has been conducted, or that no other relevant information exists. Nor shall the citation of any publication herein be construed *per se* as a representation that such publication is prior art. Moreover, the Applicant understands that the Examiner will make an independent evaluation of the cited publications.

Please charge any shortage in fees due in connection with the filing of this paper, including extension of time fees, to Deposit Account 50-1133 and please credit any excess fees to such deposit account.

Respectfully submitted,

MCDERMOTT WILL & EMERY LLP

Atabak R. Royaee
Registration No. 59,037

28 State Street
Boston, MA 02109
Phone: 617-535-4108
Facsimile: 617-535-3800
Date: November 9, 2007

Please recognize our Customer No. 23630 as our correspondence address.

10/259,494

INFORMATION DISCLOSURE CITATION IN AN APPLICATION (PTO-1449)	ATTY. DOCKET NO. 077580-0012	SERIAL NO. 10/259,494
	APPLICANT Munger et al.	
	FILING DATE Sep. 20, 2002	GROUP 1633

U.S. PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Document Number Number-Kind Code2 (if known)		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	A1	US	4,933,846 A	6/12/1990	Humphrey et al.	
	A2	US	4,988,990 A	1/29/1991	Warrior	
	A3	US	5,276,735 A	1/4/1994	Boebert et al	
	A4	US	5,311,593 A	5/10/1994	Carmi	
	A5	US	5,329,521 A	7/12/1994	Walsh et al.	
	A6	US	5,341,426 A	8/23/1994	Barney et al.	
	A7	US	5,367,643 A	11/22/1994	Chang et al	
	A8	US	5,559,883 A	9/24/1996	Williams	
	A9	US	5,561,669 A	10/1/1996	Lenney et al	
	A10	US	5,588,060 A	12/24/1996	Aziz	
	A11	US	5,625,626 A	4/29/1997	Umekita	
	A12	US	5,654,695 A	8/5/1997	Olnowich et al	
	A13	US	5,682,480 A	10/28/1997	Nakagawa	
	A14	US	5,689,566 A	11/18/1997	Nguyen	
	A15	US	5,740,375 A	4/14/1998	Dunne et al.	
	A16	US	5,774,660 A	6/30/1998	Brendel et al	
	A17	US	5,787,172 A	7/28/1998	Arnold	
	A18	US	5,796,942 A	8/18/1998	Esbensen	
	A19	US	5,805,801 A	9/8/1998	Holloway et al.	
	A20	US	5,842,040 A	11/24/1998	Hughes et al.	
	A21	US	5,845,091 A	12/1/1998	Dunne et al.	
	A22	US	5,867,650 A	2/2/1998	Osterman	
	A23	US	5,870,610 A	2/9/1999	Beyda et al.	
	A24	US	5,878,231 A	5/2/1999	Baehr et al	
	A25	US	5,892,903 A	4/6/1999	Klaus	
	A26	US	5,898,830 A	4/27/1999	Wesinger, Jr. et al.	
	A27	US	5,905,859 A	5/18/1999	Holloway et al.	
	A28	US	5,918,019 A	6/29/1999	Valencia	
	A29	US	5,996,016 A	11/30/1999	Thalheimer et al.	
	A30	US	6,006,259 A	12/21/1999	Adelman et al.	
	A31	US	6,006,272 A	12/21/1999	Aravamudan et al	

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

10/259,494

INFORMATION DISCLOSURE CITATION IN AN APPLICATION (PTO-1449)	ATTY. DOCKET NO. 077580-0012	SERIAL NO. 10/259,494
	APPLICANT **Munger et al.**	
	FILING DATE **Sep. 20, 2002**	GROUP **1633**

U.S. PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Document Number Number-Kind Code2 (if known)		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	A32	US	6,016,318 A	1/18/2000	Tomoike	
	A33	US	6,016,512	1/18/2000	Huitema	
	A34	US	6,041,342 A	3/21/2000	Yamaguchi	
	A35	US	6,052,788 A	4/18/2000	Wesinger, Jr. et al.	
	A36	US	6,055,574 A	4/25/2000	Smorodinsky et al.	
	A37	US	6,061,736 A	5/9/2000	Rochberger et al	
	A38	US	6,079,020 A	6/20/2000	Liu	
	A39	US	6,092,200 A	7/18/2000	Muniyappa et al.	
	A40	US	6,119,171 A	9/12/2000	Alkhatib	
	A41	US	6,119,234 A	9/12/2000	Aziz et al.	
	A42	US	6,147,976 A	11/14/2000	Shand et al.	
	A43	US	6,157,957 A	12/5/2000	Berthaud	
	A44	US	6,158,011 A	12/5/2000	Chen et al.	
	A45	US	6,168,409 B1	1/2/2001	Fare	
	A46	US	6,175,867 B1	1/16/2001	Taghadoss	
	A47	US	6,178,409 B1	1/23/2001	Weber et al.	
	A48	US	6,178,505 B1	1/23/2001	Schneider et al	
	A49	US	6,179,102 B1	1/30/2001	Weber, et al.	
	A50	US	6,222,842 B1	4/24/2001	Sasyan et al.	
	A51	US	6,226,751 B1	5/1/2001	Arrow et al	
	A52	US	6,233,618 B1	5/15/2001	Shannon	
	A53	US	6,243,360 B1	6/5/2001	Basilico	
	A54	US	6,243,749 B1	6/5/2001	Sitaraman et al.	
	A55	US	6,243,754 B1	6/5/2001	Guerin et al	
	A56	US	6,256,671 B1	7/3/2001	Strentzsch et al.	
	A57	US	6,263,445 B1	7/17/2001	Blumenau	
	A58	US	6,286,047 B1	9/4/2001	Ramanathan et al	
	A59	US	6,301,223 B1	10/9/2001	Hrastar et al	
	A60	US	6,308,274 B1	10/23/2001	Swift	
	A61	US	6,311,207 B1	10/30/2001	Mighdoll et al	
	A62	US	6,324,161 B1	11/27/2001	Kirch	
	A63	US	6,330,562 B1	12/11/2001	Boden et al.	

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

10/259,494

SHEET 3 OF 4

INFORMATION DISCLOSURE CITATION IN AN APPLICATION (PTO-1449)	ATTY. DOCKET NO. 077580-0012	SERIAL NO. 10/259,494
	APPLICANT Munger et al.	
	FILING DATE Sep. 20, 2002	GROUP 1633

U.S. PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Document Number Number-Kind Code2 (if known)		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
	A64	US	6,332,158 B1	12/18/2001	Risley et al.	
	A65	US	6,353,614 B1	3/5/2002	Borella et al.	
	A66	US	6,430,155 B1	8/6/2002	Davie et al	
	A67	US	6,430,610 B1	8/6/2002	Carter	
	A68	US	6,487,598 Bi	11/26/2002	Valencia	
	A69	US	6,502,135 B1	12/31/2002	Munger et al	
	A70	US	6,505,232 B1	1/7/2003	Mighdoll et al	
	A71	US	6,510,154 B1	1/21/2003	Mayes et al	
	A72	US	6,549,516 B1	4/15/2003	Albert et al	
	A73	US	6,557,037 B1	4/29/2007	Provino	
	A74	US	6,571,296 B1	5/27/2002	Dillon	
	A75	US	6,571,338 B1	5/27/2003	Shaio et al.	
	A76	US	6,581,166 B1	7/17/2003	Hirst et al.	
	A77	US	6,618,761 B2	9/9/2003	Munger et al.	
	A78	US	6,671,702 B2	12/30/2003	Kruglikov et al	
	A79	US	6,687,551 B1	2/3/2004	Steindl	
	A80	US	6,714,970 B1	3/30/2004	Fiveash et al.	
	A81	US	6,717,949 B1	4/6/2004	Boden et al.	
	A82	US	6,760,766 B1	7/6/2004	Sahlqvist	
	A83	US	6,826,616 B2	11/30/2004	Larson et al.	
	A84	US	6,839,759 B2	1/4/2005	Larson et al.	
	A85	US	7,010,604 B1	3/7/2006	Munger et al.	
	A86	US	7,133,930 B2	11/7/2006	Munger et al.	
	A87	US	7,188,180 B2	3/6/2007	Larson et al.	
	A88	US	7,197,563 B2	3/27/2007	Sheymov et al.	
	A89	US	2002/0004898 A1	1/10/2002	Droge	
	A90	US	2005/0055306 A1	3/10/2005	Miller et al.	

FOREIGN PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Foreign Patent Document Country Code3-Number4-Kind Code5 (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines Where Relevant Figures Appear	Translation Yes	No
	B1	GB 2334181 A	8/11/1999	Noblet			
	B2	EP 836306A1	4/15/1998	Sasyan et al.			

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

SHEET 4 OF 4

INFORMATION DISCLOSURE CITATION IN AN APPLICATION (PTO-1449)	ATTY. DOCKET NO. **077580-0012**	SERIAL NO. **10/259,494**
	APPLICANT **Munger et al.**	
	FILING DATE **Sep. 20, 2002**	GROUP **1633**

FOREIGN PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Foreign Patent Document Country Code3-Number4-Kind Codes (*if known*)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines Where Relevant Figures Appear	Translation Yes	No
	B3	WO 9827783 A	6/25/1998	Tello et al.			
	B4	WO 00/17775	3/30/2000	Miller et al.			
	B5	WO 00/70458	11/23/2000	Sheymov			
	B6	WO 98 55930	12/10/1998	Tang			

OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER'S INITIALS	CITE NO.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	
	C1	Alan 0. Frier et al., "The SSL Protocol Version 3.0", Nov. 18, 1996, printed from http://www.netscape.com/eng/ss13/ draft302.txt on Feb. 4, 2002, 56 pages.	
	C2	Davila J et al, "Implementation of Virtual Private Networks at the Transport Layer", Information Security, Second International Work-shop, ISW'99. Proceedings (Lecture Springer-Verlag Berlin, Germany, [Online] 1999, pp. 85-102, XP002399276, ISBN 3-540-66695-B, retrieved from the Internet: URL: http://www. springerlink.com/content/4uac0tb0hecoma89/fulltext.pdf>	
	C3	Dolev, Shlomi and Ostrovsky, Rafil, "Efficient Anonymous Multicast and Reception" (Extended Abstract), 16 pages.	
	C4	F. Halsall, "Data Communications, Computer Networks and Open Systems", Chapter 4, Protocol Basics, 1996, pp. 198–203.	
	C5	Glossary for the Linux FreeS/WAN project, printed from http://liberty.freeswan.org/ freeswan_trees/freeswan-1.3/ doc/glossary.html on Feb. 21, 2002, 25 pages.	
	C6	J. Gilmore, "Swan: Securing the Internet against Wiretapping", printed from http:// liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/rationale.html on Feb. 21, 2002, 4 pages.	
	C7	Linux FreeS/WAN Index File, printed from http://liberty.freewan.org/freeswan trees/freeswan-1.3/doc/ on Feb. 21, 2002, 3 Pages.	
	C8	Reiter, Michael K. and Rubin, Aviel D. (AT&T Labs—Research), "Crowds: Anonymity for Web Transactions", pp. 1-23.	
	C9	RFC 2401-Security Architecture for the Internet Protocol (RTP)	
	C10	RFC 2543-SIP: Session Initiation Protocol (SIP or SIPS)	
	C11	Rubin, Aviel D., Geer, Daniel, and Ranum, Marcus J. (Wiley Computer Publishing), "Web Security Sourcebook", pp. 82-94.	
	C12	Search Report, IPER (dataed Nov. 13, 2002), International Application No. PCT/US01/04340.	
	C13	Search Report, IPER (dated Feb. 06, 2002), International Application No. PCT/US01/13261.	
	C14	Search Report, IPER (dated Jan. 14, 2003), International Application No. PCT/US01/13260.	
	C15	Shankar, A.U. "A verified sliding window protocol with variable flow control". Proceedings of ACM SIGCOMM conference on Communications architectures & protocols. pp. 84-91, ACM Press, NY,NY 1986.	

EXAMINER	DATE CONSIDERED

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

(12) UK Patent Application (19) GB (11) 2 334 181 (13) A

(43) Date of A Publication 11.08.1999

(21) Application No 9802545.5

(22) Date of Filing 06.02.1998

(71) Applicant(s)
NEC Technologies (UK) Ltd
(Incorporated in the United Kingdom)
Castle Farm Campus, Priorslee, TELFORD, Shropshire, TF2 9SA, United Kingdom

(72) Inventor(s)
Charles Marie Herve Noblet

(74) Agent and/or Address for Service
J W White
NEC Technologies (UK) Ltd, Level 3, The Imperium, Imperial Way, READING, Berks, RG2 0TD, United Kingdom

(51) INT CL^{6}
H04Q 7/32

(52) UK CL (Edition Q)
H4L LDSC

(56) Documents Cited
US 5613204 A US 5109403 A

(58) Field of Search
UK CL (Edition P) H4L LDSC LDSU LECC LECX
INT CL^{6} H04Q 7/32 7/38
Online: WPI

(54) Abstract Title

Over-the-air re-programming of radio transceivers

(57) A method of downloading reprogramming data from a network for installation in a mobile station makes use of a dedicated small bandwidth pilot channel. The mobile station obtains from the base station the radio access parameters of a second channel. The second channel is a large bandwidth (bootstrap) channel suitable for fast transfer of data. The bootstrap channel is logically mapped onto a local transmission mode such as DECT or GSM by the mobile station and re-programming data may be downloaded from the base station via the bootstrap channel.

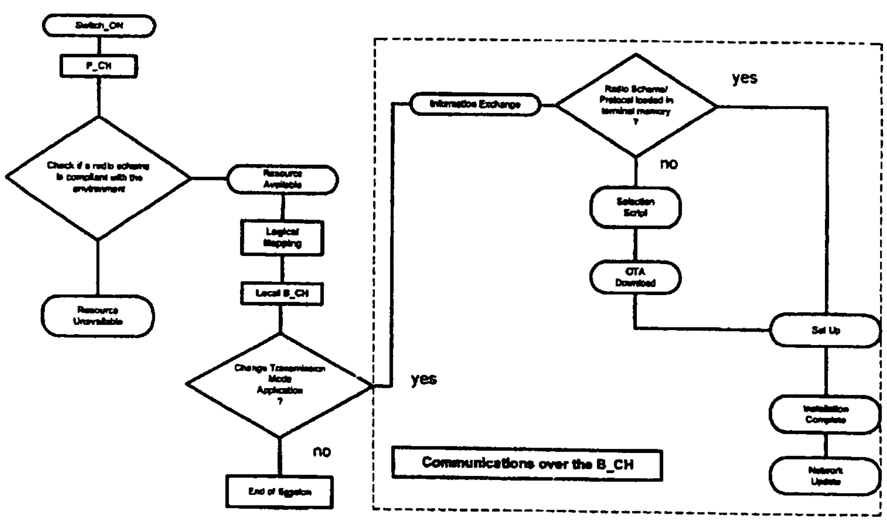


Figure 2

GB 2 334 181 A

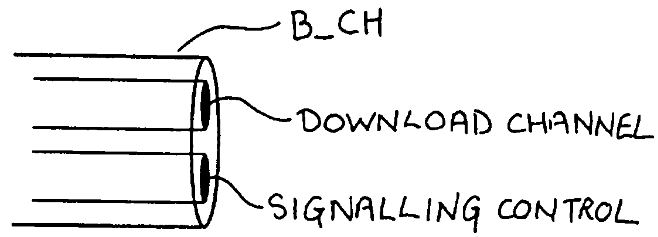
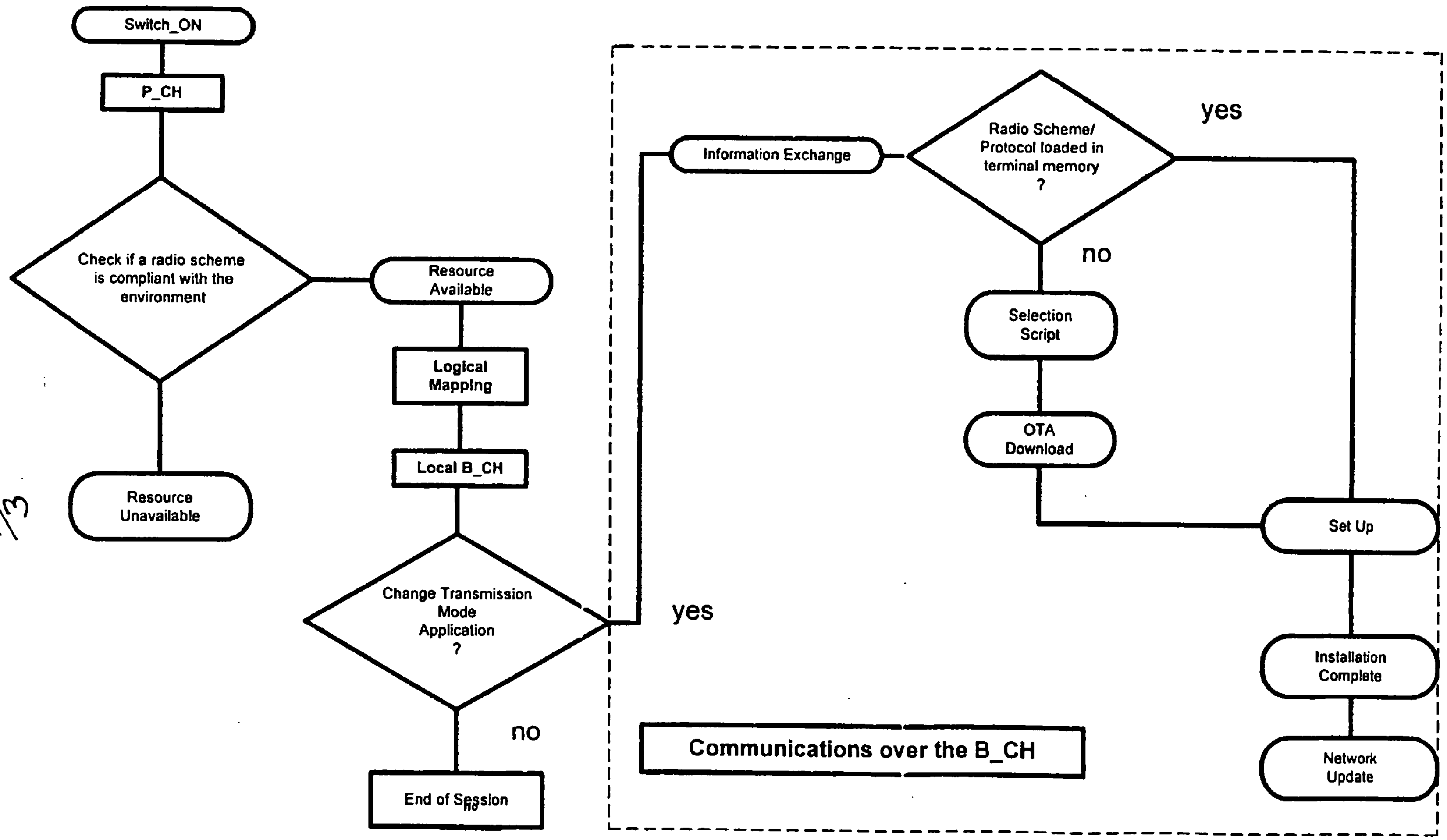


Figure: 1

2/3



Switch_ON
P_CH
Check if a radio scheme is compliant with the environment
Resource Unavailable
Resource Available
Logical Mapping
Local B_CH
Change Transmission Mode Application ?
no
End of Session
yes
Information Exchange
Radio Scheme/ Protocol loaded in terminal memory ?
yes
no
Selection Script
OTA Download
Set Up
Installation Complete
Network Update
Communications over the B_CH

Figure 2

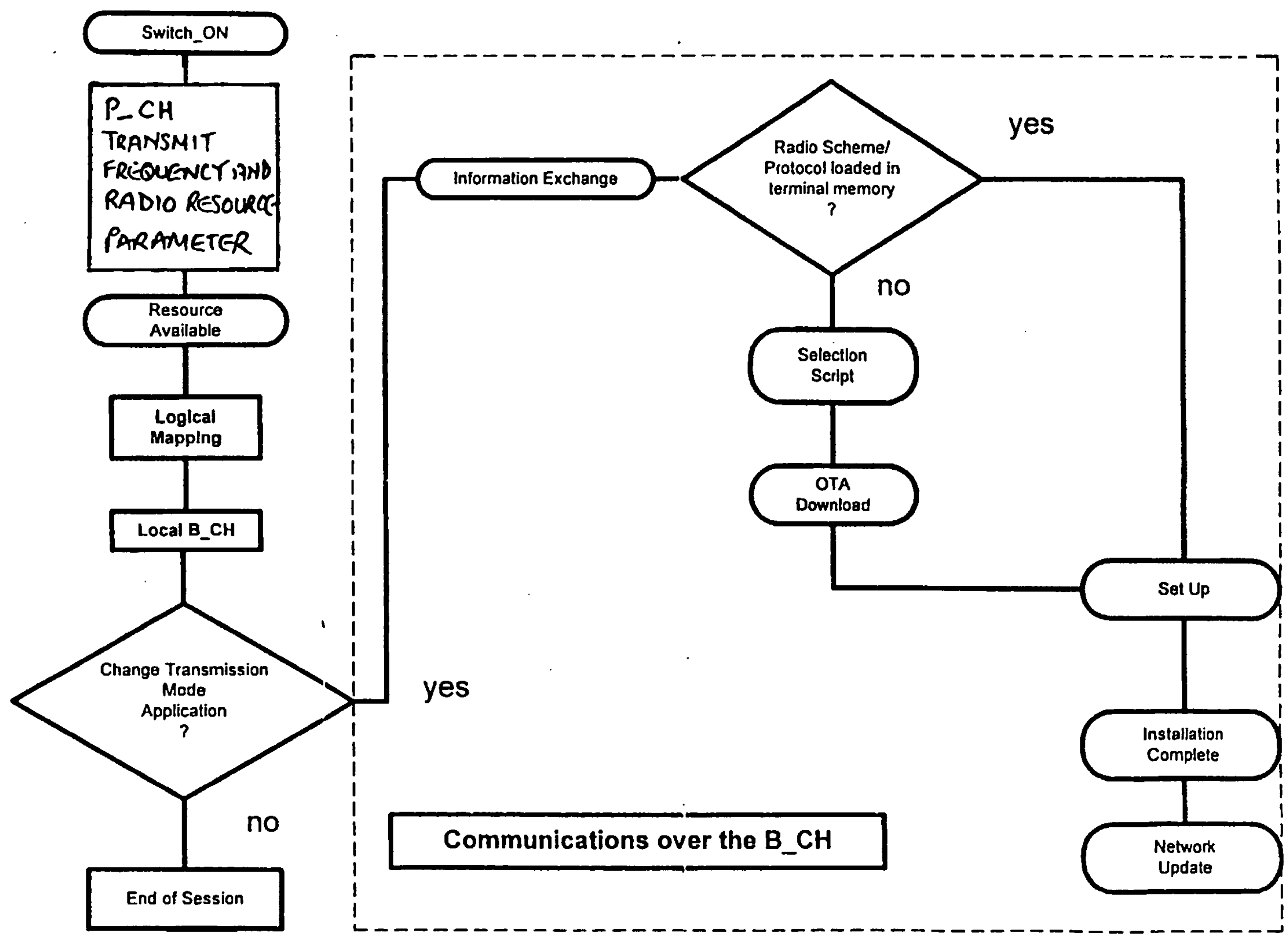

Switch_ON
P_CH TRANSMIT FREQUENCY AND RADIO RESOURCE PARAMETER
Resource Available
Logical Mapping
Local B_CH
Change Transmission Mode Application ?
no
End of Session
yes
Information Exchange
Radio Scheme/ Protocol loaded in terminal memory ?
yes
no
Selection Script
OTA Download
Set Up
Installation Complete
Network Update
Communications over the B_CH

Figure 3

2334181

Over-the-air re-programming of radio transceivers

This invention relates to radio transmitter/receivers and in particular it relates to a method of re-programming radio transmitter/receivers over-the - air.

A radio transmitter/receiver (transceiver) such as a radiotelephone is designed for operation with particular types of networks such as GSM 900 or DCS 1800. Intended use of the radiotelephone with a particular network(s) in a restricted geographical area, however, requires that the telephone be configured so as properly to communicate with the particular network (s). The user of a radiotelephone will usually have a telephone which has been configured for communication with a so called "home network". The home network is the local network usually most used by the subscriber.

The area within which a user of e.g. a GSM radiotelephone may operate, however, is considerable and is not limited to the home network but may be used on many other networks throughout the world. Use of a handset outside the home network is known as "roaming".

When the radiotelephone is to be used in roaming it is often necessary for it to have a configuration different to that for use with the home network. It is possible for re-configuration of radio transmitter/receivers to be effected by means of signals received across the air interface.

It is also convenient for the radio to be re-configurable over the air interface so as to support different types of communication and user applications e.g. addition of address book manager, whether or not it is located in the home network.

Over the air re-programming of radio receivers is well known in the art and reference may be made to US patent 5 381 138 for example. The capability to obtain programming data from a network is particularly useful for a roaming radio transmitter/receiver.

When beginning operation in an area for which the radiotelephone is not configured and it is required to download the data for reconfiguration from one of the available networks, a communication link must first be established with the network of interest. It has been proposed that a pilot channel be established in all areas from which the roaming radiotelephone may obtain the data necessary for reconfiguration.

A pilot channel of this type, however, will require a relatively large bandwidth to allow a sufficiently fast transfer of the data required.

According to the invention there is provided a method of downloading reprogramming data from a network for installation in a radio transmitter/receiver comprising initial communication from a first dedicated channel of relatively small bandwidth broadcasting at least the frequency and radio access parameters of a second channel of relatively large bandwidth from which reprogramming data may be downloaded.

Examples of the invention will now be described in more detail with reference to the accompanying figures in which

figure 1 Illustrates the logical structure of the bootstrap channel

figure 2 Is a flow diagram of a reconfiguration process

figure 3 Is a flow diagram of an alternative reconfiguration process

A roaming radio transmitter/receiver (mobile) is located in a region served by one or more networks and the user wishes to communicate with a network from which he can obtain reprogramming data and subsequently begin communicating with the network in the communication mode selected.

A pilot channel broadcast is maintained in the region and contained in the pilot channel broadcast there is at least sufficient information for the mobile to connect to a second channel which we shall call the bootstrap channel. Conveniently the pilot channel will be broadcast in all regions over a standardised radio interface. Only a small bandwidth is required for the pilot channel because of the small amount of information contained in the broadcast.

The small bandwidth requirement makes the task of standardisation much easier with respect to the pilot channel. The wider bandwidth channels are more conveniently assigned locally for ease of implementation.

The Pilot Channel (P_CH) broadcasts a list of sets of parameters corresponding to networks available in the region. The mobile receives the network transmission through the P_CH. If the existing configuration of the mobile is matched to the available regional radio schemes, then a second channel the bootstrap channel (B_CH) is logically mapped onto the selected transmission mode. The base station and mobile exchange information over this dedicated logical channel.

The Bootstrap channel is logically mapped on top of one of the default modes of the terminal; a mapping of a logical B_CH onto the physical GSM channel for instance may be implemented. Once the mapping has been effected the terminal may download data from the base station. The bootstrap channels provided by each operator may accommodate differing services with regard to the applications available for downloading.

The flow diagram shown at fig 3 depicts a reconfiguration procedure.

When the mobile is switched on, it reads the Pilot Channel broadcast. The mobile must be configured to support the (standardised) radio interface of the Pilot Channel. The Pilot Channel carries local radio parameters (standards supported in the regional environment in which the mobile is located). After processing the received information, the mobile communicates with the base station through the Bootstrap Channel, provided that the mobile has the minimum resources required by its local radio environment. Prior to the change of channel, P_CH to B_CH, a logical mapping of the Bootstrap Channel is performed within the mobile on the selected air interface.

When operation on a local B_CH transmission has been established, the user may wish to change some properties or the performance of his mobile and can request supply of the desired services from the network. If no changes are required then the mobile adopts the default transmission mode in stand-by and releases the allocated B_CH.

If the user requests a change then communication between the base station and mobile is maintained for the exchange, the nature of which will depend on the capabilities of both mobile and network. At least 3 conditions can affect the nature of this information exchange.

Firstly, the mobile may not be able to support the required software. Where the mobile is not able to support the required software, no communication channel is available to the mobile from the existing network resources and use of the mobile within the region will therefore not be possible.

Secondly, the required software may be stored already in the mobile's memory. In this situation there is no need to download a software module but the allocated B_CH connection is maintained for further operations as described.

Thirdly, the software module required to support a different type of communication or user application may need to be downloaded from the base station. Where the download of a software module is required, initially a selection script is downloaded to the mobile followed by downloading and installation of the required software.

When the installation of the required software into the mobile has been completed, the mobile signals to the network the achievement of correct reconfiguration. On receipt of the "correct reconfiguration" signal from the mobile details of the mobile identity and its present configuration are entered on the network database (to license the product for instance) .

With reference to figure 1, the logical structure of the bootstrap channel will include 2 logical sub-channels : a download channel and a signalling control channel (S_CH). The signalling control channel assists in the reduction of errors in transmission so as to allow correct software download.

In the above example, the first channel, the Pilot Channel, is standardised and the mobile must be configured to support the radio interface for the Pilot Channel. The second (bootstrap) channel may be subject to local definition through logical mapping on a local transmission mode e.g. GSM, DECT and the mobile is not initially configured to support the radio interface for the bootstrap channel..

An example of a method of reprogramming providing greater flexibility will now be given. In this example the mobile is configured to support the radio interfaces for both the first, dedicated relatively small bandwidth (Pilot) channel and the second relatively large bandwidth (bootstrap) channel. That is to say that when the mobile is switched on in most and preferably all regions, the network can communicate with the mobile via both pilot and bootstrap channels.

In order for the mobile always to have the appropriate radio interface for the bootstrap channel then this channel would need also to be standardised (in addition to the Pilot Channel). The parameters of the bootstrap channels provided in different regions may have local variations in terms of e.g. allocated frequency, data rate and available user applications.

With reference to figure 3 which is a flow diagram of the reconfiguration process for this example, the mobile when switched on reads the Pilot Channel broadcast. The allocated frequency and radio resource parameters for the bootstrap channel contained in the pilot channel broadcast are processed and any required logical mapping effected. After processing the received information, the mobile communicates with the base station through the Bootstrap Channel.

The condition likely to be experienced in the previous example whereby the mobile is not able to support the required software and no communication channel is available to the mobile from the existing network resources does not apply in this arrangement. The communication via the bootstrap channel allows the request for and supply of the software module necessary to establish communication with the network. The transfer to the bootstrap channel does not depend on the existing configuration of the mobile since the bootstrap channel is standardised in this example and the mobile is equipped to interface, via the pilot channel, with the bootstrap channel.

The services and structure offered by the Bootstrap Channel are common for both of the above examples, however, the requirements on the terminals and networks differ.

The bootstrap channel will provide the following services by means of over -the-air (OTA) reconfiguration :

capability Exchange - the terminal provides some information to the network on its current configuration and capabilities.

module Selection : at this stage the user specifies the software that his terminal requires to download. This operation could be compared to an installation script.

data download : transfer of the data. In some cases software code will have to be downloaded whilst in other cases the software may already be implemented in the mobile. In the latter case, a set-up mechanism would be sufficient to initiate the reconfiguration.

Once the mobile and the base station are synchronised on the bootstrap channel, information exchange can begin.

Claims

1. A method of downloading reprogramming data from a network for installation in a radio transmitter/receiver comprising initial communication from a first dedicated channel of relatively small bandwidth broadcasting at least the frequency and radio access parameters of a second channel of relatively large bandwidth from which reprogramming data may be downloaded.

2. A method of downloading reprogramming data from a network as in claim 1 where first, dedicated relatively small bandwidth channel has a standard radio interface common to many network locations.

3. A method of downloading reprogramming data from a network as in claim 2 where second relatively large bandwidth channel has a standard radio interface common to many network locations.

4. A method of downloading reprogramming data from a network as in claims 1 to 3 where first, dedicated relatively small bandwidth channel broadcasts a list of sets of parameters corresponding to networks available in the region.

5. A method of downloading reprogramming data from a network as in claim 1 where the radio transmitter/receiver is configured to support the radio interfaces for both the first, dedicated relatively small bandwidth channel and the second relatively large bandwidth channel.

The Patent Office

10

Application No: GB 9802545.5
Claims searched: 1 to 5

Examiner: Glyn Hughes
Date of search: 17 August 1998

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.P): H4L (LDSC, LDSU, LECC, LECX)

Int Cl (Ed.6): H04Q 7/32, 7/38

Other: Online: WPI

Documents considered to be relevant:

Category	Identity of document and relevant passage		Relevant to claims
X	US 5613204	(HABERMAN ET AL) see in particular column 15 lines 48 to 50	1
X	US 5109403	(SUTPHIN) see abstract	1

X Document indicating lack of novelty or inventive step	A Document indicating technological background and/or state of the art.
Y Document indicating lack of inventive step if combined with one or more other documents of same category.	P Document published on or after the declared priority date but before the filing date of this invention.
& Member of the same patent family	E Patent document published on or after, but with priority date earlier than, the filing date of this application.

An Executive Agency of the Department of Trade and Industry

B2

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 836 306 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
15.04.1998 Bulletin 1998/16

(51) Int. Cl.[6]: H04L 29/06, H04Q 11/04

(21) Application number: 96410106.7

(22) Date of filing: 10.10.1996

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV SI

(71) Applicant:
Hewlett-Packard Company
Palo Alto, California 94304 (US)

(72) Inventors:
- Sasyan, Serge
  38170 Seyssinet (FR)
- Roger, Denis
  38760 Varces (FR)
- Terrasse, Denis
  38320 Eybens (FR)

(74) Representative:
Squibbs, Robert Francis
Intellectual Property Section,
Legal Department,
Hewlett-Packard France,
Etablissements de Grenoble
F-38053 Grenoble Cédex 9 (FR)

Remarks:
The application is published incomplete as filed (Article 93 (2) EPC). A claim No.7 is missing.

(54) System providing for multiple virtual circuits between two network entities

(57) Computers sending IP datagrams over an ATM network are generally capable of operating multiple simultaneous virtual circuits over the network. However, in doing so, they normally only set up one virtual circuit to each destination IP address so that in order to test the simultaneous operation of N virtual circuits by a computer under test, N target computers are needed. To enable a single computer (T) to provide the destination endpoints for multiple virtual circuits (SVC) from a computer (M) under test, both computers (M,T) are allocated a plurality of virtual IP addresses ($I_{M(i)}, I_{T(i)}$) and the target computer (T) is additionally provided with a module running address-changing processes (70,71) that avoids the IP layers (20) of both computers from rejecting IP datagrams (25A,25B) addressed with the virtual IP addresses. As a result, each computer (M,T) can be addressed with any of a plurality of IP addresses and each will result in the creation of a respective virtual circuit (SVC) between the computers (M,T).

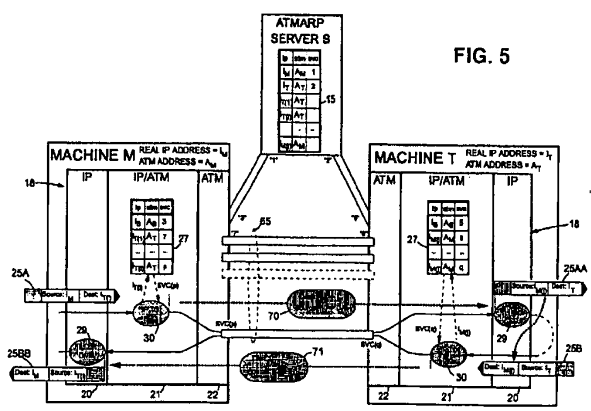


FIG. 5

EP 0 836 306 A1

Printed by Xerox (UK) Business Services
2 15.12/3 4

Description

Field of the Invention

The present invention relates to a system providing for multiple virtual circuits between two network entities for use in particular, but not exclusively, in the testing of network node apparatus providing IP messaging over an ATM network.

Background of the Invention

As is well-known, the Internet Protocol (IP) uses a scheme of IP addresses by which every connection of a node to the Internet has a unique IP address. IP addresses are high-level addresses in the sense that they are independent of the technology used for the underlying network to which a node is connected. Each node will also have a low-level, network-dependent address (often callled the MAC address) that is actually used for addressing at the network level and the IP protocol suite includes a address reolution protocol (ARP), logically positioned below the IP layer itself, that is responsible for translating between IP addresses contained in a message and the local MAC addresses.

An increasingly important technology for local area networks is ATM. ATM (Asynchronous Transfer Mode) is a multiplexing and switching technique for transferring data across a network using fixed sized cells that are synchronous in the sense that they appear strictly periodically on the physical medium. Each cell comprises a payload portion and a header, the latter including a label that associates the cell with an instance of communication between sending and receiving network end systems; this instance of communication may involve the transfer of many cells from the sending end system, possibly to multiple receiving end systems. ATM is asynchronous in the sense that cells belonging to the same instance of communication will not necessarily appear at periodic intervals.

In ATM, the labels appended to the cells are fixed-size context dependent labels, that is, they are only understandable in the light of context information already established at the interpreting network node, the label generally being replaced at one node by the label required for the next node. In other words, ATM is a virtual circuit technology requiring a set up phase for each instance of communication to establish the appropriate label knowledge at each node. Of course, to set up a desired communication, it is still necessary to identify uniquely the nodes forming the communication end points and this is achieved by using ATM addresses, generally of a significance limited to the particular ATM network concerned.

The process of sending IP messages (datagrams) over a ATM network, including the operation of the required ATM ARP system, is set out in RFC 1577 of the IETF Internet Engineering Task Force) dated January 1993. This RFC assumes an arrangement in which a sending node will only establish a single vircuit circuit to a given destination IP address (of course, this one vircuit circuit may carry multiple connections between respective pairings of high-level end points in the nodes).

Figure 1 of the accompanying drawings is a diagram illustrating the basic mechanism by which two machines M and T exchange IP datagrams over a switched virtual circuit (SVC) established across an ATM network. The machines M and T have respective IP addresses I_M and I_T and respective ATM addresses A_M and A_T; each machine knows its own addresses. An ATMARP server S knows the IP and ATM addresses of all active nodes on the network, including machines M and T; more particularly, server S maintains an ARP table 15 associating the IP address of each node with its ATM address. The server S maintains open a respective SVC (switched virtual circuit) to each active node and the identity of this SVC is held in the ARP table 15; thus, in the Figure 1 example, the server S is in communication with machine M over an SVC identified as SVC "1" at the server, and the server S is in communication with machine T over an SVC identified as SVC "2" at the server S. At machines M and T these virtual circuits are independently identified - thus at machine M its SVC to the server S is identified as SVC "3" whilst at machine T its SVC to the server S is identified as SVC "5".

The communications interface 18 in each of the machines M and T comprises three main layers, namely: an IP layer 20 responsible for forming IP datagrams (including source and destination IP addresses) for transmission ad for filtering incoming datagrams; an intermediate IP/ATM layer 21 for determining the SVC corresponding to the destination IP address of an outgoing datagram; and an ATM layer 22, including the low-level network interface hardware, for sending and receiving datagrams packaged in ATM cells over SVCs.

The IP/ATMlayer 21 maintains an ARP cache table 27 which like the table 15 of the server S contains associations between IP address, ATM address and SVC. Thus, table 27 of machine M contains an entry of the IP address I_S, ATM address A_S, and SVC identity "3" for the server S, and similarly, table 27 of machine T contains an entry of the IP address I_S, ATM address A_S, and SVC identity "5" for the server S. The cache table 27 only holds information relevant to current SVCs of the machine concerned so that during the initial establishment of a SVC to a new destination, the cache table must be updated with relevant information from the ATMARP server S; this general process will be described in more detail hereinafter with reference to Figure 2. For the present, it will be assumed that an SVC has already been established between machines M and T and that the cache tables contain the relevant information (in particular, cache table 27 of machine M contains an entry with the IP address I_T, ATM address A_T, and SVC identity "4" for machine T, and cache table

27 of machine T contains an entry with the IP address I_M, ATM address A_M, and SVC identity "9" for machine M).

Considering now the case of a high-level application in machine M wanting to send a message to machine T, this application passes the message to the IP layer 20 together with the destination IP address I_T. IP layer 20 packages the message in one (or more) datagrams 25A with a destination IP address of I_T and source I_P address of I_M. Datagram 25A is then passed to the IP/ATM layer 21 which executes an IP-to-SVC lookup task 30 to determine from table 27 the SVC to be used for sending the datagram to its destination address I_T; in the present case, table 27 returns the SVC identity "4" and the layer 21 passes this identity together with the datagram 25A to the ATM layer 22 which then sends the datagram in ATM cells on SVC "4". The datagram is in due course received by machine T and passed up by layers 22 and 21 to the IP layer 20 where a filtering task 29 determines from the datagram destination address that the datagram is indeed intended for machine T; the contents of the datagram are then passed to the relevant high-level application. In the present example, this high-level application produces a reply message which it passes to the IP layer 20 together with the required return address, namely the source IP address in the received datagram 25A. IP layer 20 generates datagram 25B with the received return address as the destination address, the IP address I_T of machine T being included as the source address. The datagram 25B is passed to IP/ATM layer 21 where IP-to-SVC lookup task 30 determines from cache table 27 that the required destination can be reached over SVC "9". This information together with datagram 25B is then passed to ATM layer 22 which transmits the datagram in ATM cells over SVC "9" to machine M. When the datagram is received at machine M it is passed up to the IP layer 20 where it is filtered by task 29 and its contents then passed on to the relevant high-level application.

Figure 2 of the accompanying drawings illustrates in more detail the functioning of the IP/ATM layers 21 of machines M and T in respect of datagram transmission from machine M to machine T, it being appreciated that the roles of the two layers 21 are reversed for transmission in the opposite direction. More particularly, upon the IP-to-SVC lookup task 30 being requested to send a datagram to IP address I_T, it first carries out a check of the cache table 27 (step 31) to determine if there is an existing entry for I_T (and thus an SVC, assuming that entries are only maintained whilst an SVC exists). Step 32 checks the result of this lookup - if an SVC already exists (in this case, SVC "4"), then step 39 is executed in which the datagram is passed together with the identity of the relevant SVC to the ATM layer 22; however, if the lookup was unsuccessful, task 30 executes steps 33 to 38 to set up an SVC to destination I_T before executing step 39.

The first step 33 of the setup process involves the sending of an ARP request to the ATMARP server S over the relevant SVC requesting the ATM address corresponding to I_T. Server responds with ATM address A_T which is received by task 30 at step 34.

Task 30 now updates the cache table 27 with the IP address I_T and ATM address A_T (step 35). Next, task 30 requests (step 36) the ATM layer 22 to establish a new SVC to ATM address A_T and this initiates an SVC setup process 28 which may be executed in any appropriate manner and will not be described in detail herein. In due course, process 28 returns the identity of the SVC that has been set up to A_T (in this case, SVC "4"), this identity being received at step 37 of task 30. Finally, cache table 30 is updated at step 38 by adding the SVC identity ("4") to the entry already containing I_T and A_T.

In machine T, the setup of the new SVC to the machine from machine M is handled by the setup process 28 of machine T. The process 28 informs the IP/ATM layer that a new SVC has been setup and this triggers execution of an update task 40 to update the cache table 27 of machine T. More particularly, on the new SVC indication being received (step 41), a first update step 42 is carried out to add an entry to the table confining the identity of the new SVC (in the present example "9"), and the ATM address A_M of the node at the other end of the SVC; at this stage, the corresponding IP address is not known to machine T. In order to obtain this IP address, an inverse ARP request is now made to machine M (step 43). In due course a response is received (step 44) containing the IP address of machine M. The cache table 27 is then updated at step 45 with the IP address I_M of machine M and the IP/ATM layer is now ready to effect IP-to-SVC translations for datagrams intended for machine M.

The inverse ARP request sent by machine T to machine M is handled by an inverse ARP task 50 that examines the request (step 51) and on finding that it contains the ATM address A_M, responds with the IP address I_M of machine M (step 52).

To facilitate explanation of the preferred embodiment of the invention hereinafter, the messages across the boundary between the IP/ATM layer 21 and the ATM layer 22 have been labelled in Figure 2 as follows where superscript "T" indicates an outgoing message (that is, from the IP/ATM layer to the ATM layer) and the superscript "R" indicates incoming messages (that is, from the ATM layer to the IP/ATM layer):

- $X1^T$ - outgoing ARP request;
- $X2^R$ - incoming ARP response;
- $X3^T$ - outgoing SVC setup request;
- $X4^R$ - incoming SVC setup done indication;
- $X5^R$ - incoming new SVC indication;
- $X6^T$ - outgoing INARP request;
- $X6^R$ - incoming INARP request;
- $X7^T$ - outgoing INARP response;
- $X8^T$ - outgoing datagram;
- $X8^R$ - incoming datagram.

It will be appreciated that machines connecting to an ATM network, such as machines M and T as well as the server S, are designed to handle a large number of virtual circuits simultanteously. If in testing such a machine (machine M in the following discussion) it is desired to fully stress the machine under test, then the design limit of concurrently operating virtual circuits must be simultaneously used. However, as already indicated, current practice is that only one virtual circuit is established to each distinct IP address. As a result, since generally each machine that might be used to test machine M has only one network connection and therefore only one IP address, if machine M is designed to operate up to N virtual circuits simultaneously, then it requires N machines to test machine M. Such an arrangement is illustrated in Figure 3 where the N machines are constituted by the server S and (N-1) other machines here represented as machines T1 to T(N-1). Such an arrangement is generally impractical as N may be as high as 1024 or more.

It is an object of the present invention to provide a mechanism that enables, inter alia, the foregoing test problem to be overcome.

Summary of the Invention

According to the present invention, there is provided a system in which a plurality of entities are connected to a network ad can exchange messages across virtual circuits set up over the network between said entities, each entity having a operative high-level address on the network, and each entity comprising:

-- high-level messaging means for handling message transmission and receipt on the basis of the aforesaid high-level addresses, the high-level messaging means comprising means for including in outgoing messages the operative high-level address of the entity as a source identifier and the operative high level address of the intended recipient entity as a destination identifier, and means for filtering incoming messages according to the destination identifier contained in the message:

-- virtual-circuit means for providing virtual circuits between the entity and other entities, there being a respective virtual circuit for each different destination identifier in use, and

-- intermediate means for passing an outgoing message from the high-level messaging means to that one of the virtual circuits provided by the virtual-circuit means which corresponds to the destination identifier of the message;

characterised in that each of a first and a second one of the entities has a plurality of virtual high-level addresses associated with it that are different from the operative high-level address of the entity, the virtual high-level addresses being usable by the messaging means of the first and second entities as destination identifiers in outgoing messages; **and in that** between the intermediate means of the first and second entities, there are provided address-changing means responsive to each of at least some of the messages sent between these entities with a said virtual high-level address as its destination identifier, to change that address to the operative high-level address of the corresponding entity and to change the operative high-level address provided as the source identifier of the message into one of the said virtual high-level addresses associated with the sending entity in dependence on the virtual high-level address initially provided as the destination identifier of the same message.

By virtue of this arrangement, it is possible to establish a plurality of virtual circuits between the first and second entities by using the different virtual high-level addresses of the entities as the destination identifiers in messages exchanged between the entities, the receiving high-level adressing means accepting such messages due to the address-changing means having changed the destination identifier to the operative high-level address of the receiving entity. By also changing the source identifier, it is possible to retain in the message information sufficient to associate any reply message with a particular one of the virtual circuits established with the sending entity (in particular, the reply message can be sent back over the same virtual circuit as the message to which it is a reply - however, if desired, it is also possible to use a separate virtual circuit for the reply messages).

Preferably, the address-changing means comprises first address-changing functionality for effecting the aforesaid changes for messages sent from the first entity to the second entity, and second address-changing functionality for effecting these changes for messages sent from the second entity to the first entity, both the first and second address-changing functionalities being provided in the second entity. This configuration is well suited for testing the ability of network node apparatus to concurrently operate a plurality of virtual circuits where the network node apparatus is operative to establish a virtual circuit for each different high-level destination address being handled; more particularly, the network node apparatus serves as the aforesaid first entity, and is caused to send messages to at least some of the vial high-level addresses associated with the second entity. By placing the address-changing means in the second entity, no modifications are needed to the network node apparatus in order for it to be able to establish a plularity of virtual circuits with the second entity.

Advantageously, the address-changing means effects a predetermined transformation on the virtual high-level address forming the initial destination identifier of a said message in order to form the virtual high-level address to be used for the source identifier of that message. For example, this transformation may simply involved changing the address by one (where the address is numeric in form).

The present invention is particularly applicable to systems in which the high-level addresses are IP addresses and the network is an ATM network.

Brief Description of the Drawings

A system embodying the invention will now be described, by way of non-limiting example, with reference to the accompanying diagrammatic drawings, in which:

- **. Figure 1** is a diagram of a known system for sending IP datagrams over a ATM network between two machines M and T;
- **. Figure 2** is a diagram illustrating the steps carried out by the Figure 1 system in establishing a virtual circuit between machines M and T;
- **. Figure 3** is a diagram of a known test arrangement for testing the ability of a machine M to concurrently operate multiple virtual circuits;
- **. Figure 4** is a diagram showing a test arrangement embodying the invention for testing the ability of a machine M to concurrently operate multiple virtual circuits;
- **. Figure 5** is a diagram similar to Figure 1 but showing a system embodying the invention in which multiple virtual circuits are established between machines M and T;
- **. Figure 6** is a diagram illustrating the processing effected by a module VNS disposed in machine T of the Figure 5 system when machine M initiates the opening of a new virtual circuit between machines M and T; and
- **. Figure 7** is a diagram illustrating the processing effected by a module VNS disposed in machine T of the Figure 5 system when machine T initiates the opening of a new virtual circuit between machines M and T.

Best Mode of Carrying Out the Invention

The embodiment of the invention now to be described provides a system in which it is possible to establish a plurality of SVCs (switched virtual circuits) across an ATM network for the exchange of IP datagrams between two machines M and T whereby it is possible to test the ability of machine M to concurrently operate a plurality of virtual circuits without needing to provide a respective destination machine for each SVC operated by machine M. The overall test arrangement is illustrated in Figure 4 where machine M operates N SVCs over ATM network 10, one SVC being with ATMARP server S and (N-1) SVCs being with machine T. According to the preferred embodiment, the establishment of multiple concurrent SVCs between machine M and T is effected without modification to machine M.

Figure 5 shows a system embodying the present invention, this system being similar to that of Figure 1 but being operative to provide a plurality of concurrent SVCs 65 between machines M and T. In the Figure 5 system, the machines M and T and the server S are assumed to operate in the same way and have the same IP and ATM addresses as in Figure 1; in addition, in Figure 5 the same SVCs are established between the server S and the machines M and T as in Figure 1. The Figure 5 system includes, however, added functionality provided by processes 70 and 71 which in Figure 5 are shown independent of machines M and T but in practice would be provided either distributed between machines M and T or wholly in one of these machines; in a preferred embodiment, the processes 70 and 71 are provided in machine T.

In accordance with the present invention, each machine M and T is allocated a number of virtual IP addresses different from its operative (or "real") IP address (this latter address being the one which the IP layer knows about for inclusion as the source address in outgoing datagrams and upon which filtering is carried out by task 29). Thus, machine M is allocated virtual IP addresses $I_{M(1)}$, $I_{M(2)}$,......$I_{M(j)}$...; similarly, machine T is allocated virtual IP addresses $I_{T(1)}$, $I_{T(2)}$,......$I_{T(i)}$....

Each of these virtual IP addresses is entered into table 15 of ATMARP server S together with the ATM address of the corresponding one of the machines M,T; thus virtual IP address $I_{M(j)}$ is associated with ATM address A_M and virtual IP address $I_{T(i)}$ is associated with ATM address A_T.

Now, if the communications interface 18 of machine M is asked to send a message to IP address $I_{T(i)}$, IP layer 20 will construct a datagram 25A having a destination address of $I_{T(i)}$ and a source address of I_M. The IP-to-SVC task 30 of IP/ATM layer 21 then acts in the manner already described to fetch the ATM address corresponding to $I_{T(i)}$ from server S and set up an SVC (here identified by "p") towards machine T; the cache table 27 is updated appropriately. The datagram 25A is now sent by ATM layer over SVC(p) to machine T.

If no further action is taken, the datagram 25A, after receipt at machine T, will be rejected by the filter task 29 as the destination address $I_{T(i)}$ of the datagram differs from the operative IP address I_T known to task 29 of machine T. Accordingly, a process 70 is provided that recognises the destination address of datagram 25A as being a virtual IP address of machine T and substitutes the real IP address of machine T for the virtual address in the destination field of the datagram 25A. The datagram will now be allowed through by filter task 29 of machine T.

However, a further difficultly remains. If only the destination address is changed, the resultant datagram contains no indication that the datagram was not ordinarily sent with the real IP address of machine T; any reply will therefore be sent on an SVC set up to take datagrams from machine M to the real IP address of machine M. This SVC would end up taking all the reply messages for messages sent from machine M to machine T over all the SVCs set up in respect of the virtual IP addresses allocated to machine T. This is clearly undesirable. To avoid this, the source address of datagram 25A is also changed by process 70. More particularly, the source address is changed from the real IP address of machine M to one of the virtual IP addresses $I_{M(j)}$ of this machine, the virtual address chosen being dependent on the original virtual IP address forming the destination address of the datagram. As a result, all datagrams 25A having the same virtual destination address end up after operation of process 70 as datagrams 25AA with the same virtual source address, whereas datagrams 25A having different initial virtual destination addresses end up as datagrams 25AA with different source addresses. The process of changing the source address preferably involves a predetermined transformation of the virtual destination address - for example, to obtain the required virtual source address, the virtual destination address can simply be incremented by one (there would thus exist, for example, a set of even virtual IP addresses for machine M and a corresponding set of odd virtual IP addresses for machine T, each even virtual IP address of machine M being associated with the immediately adjacent, lower-valued, odd virtual IP address of machine T).

The address-changing process 70 must be carried out on datagram 25A after operation of the IP-to-SVC task 30 in machine M and prior to the filter task 29 in machine T. In addition, whilst the two address-hanging operations of process 70 need not be carried out at the same time or at the same location (though it is, of course, convenient to do so), the changing of the source address must be done whilst the initial virtual destination address is still available.

The contents of datagram 25AA are passed by IP layer 20 of machine T to a high-level application which, in the present example, produces a reply that it passes to layer 20 for sending back to IP address $I_{M(j)}$, that is, to the source address contained in datagram 25AA. Layer 20 produces a datagram 25B with source address I_T and destination address $I_{M(j)}$. Next, IP-to-SVC task 30 of layer 21 looks up the destination address in the cache table 27 to find out the SVC to be used for the reply. If, as will normally be the case, the same SVC is to be used for the reply as carried the original datagram 25A with destination address $I_{T(i)}$, then the SVC setup process will have been arranged to enter the address $I_{M(j)}$ in cache table 27 against that SVC (in present case, identified to machine T by "q"); a lookup on $I_{M(j)}$ will thus return "q" as the required SVC. However, if it is desired to use a different SVC for datagrams 25B passing from T to M as used for datagrams 25A passing from M to T, then the first lookup on $I_{M(j)}$ by task 30 will not identify an SVC and task 30 must then initiate set up of a new SVC.

Assuming that the same SVC is to be used for the datagrams 25B with destination address $I_{M(j)}$ as for the datagrams 25A with destination address $I_{T(i)}$, then after task 30 has identified SVC(q) as the appropriate SVC, the datagram 25B is passed to the ATM layer 22 for sending out over SVC(q). In due course, machine M receives this datagram and passes it up to IP layer 20; however, before the datagram reaches this layer, it must undergo address-change processing similar to that carried out on datagram 25A. More particularly, the virtual destination address $I_{M(j)}$ must be changed to the real IP address I_M of machine M, and the real source address I_T of machine T must be changed to the virtual IP address $I_{T(i)}$ of machine T associated with the virtual destination address $I_{M(j)}$. This address-change processing is carried out by process 71.

With regard to the source address change, where the corresponding change was effected for datagram 25A by incrementing by one the virtual destination address $I_{T(i)}$ of that datagram, then for datagram 25B, the source address is changed to the destination address $I_{M(j)}$ decremented by one.

In a similar manner to process 70, process 71 must be carried out on datagram 25B after operation of the IP-to-SVC task 30 in machine T and prior to the filter task 29 in machine M. In addition, whilst the two address-changing operations of process 71 need not be carried out at the same time or at the same location, the changing of the source address must be done whilst the initial virtual destination address is still available.

Following operation of process 71, datagram 25BB with source address $I_{T(i)}$ and destination address I_M is allowed through by filter task 29 and the contents of the datagram are passed to the relevant high-level application.

Having described the general mechanism by which virtual IP addresses can be used for exchanging datagrams 25A and 25B across a SVC between machines M and T, the issue will now be addressed as to how the cache table 27 in machine T is updated on SVC setup to associate the new SVC (that is, SVC(q) at machine T) with the virtual IP address $I_{M(j)}$ of machine M (this is required where the same SVC is to be used for the reply datagram 25B as for the original datagram 25A). It will be appreciated that when the task 40 (see Figure 2) is executed, the INARP request sent to machine M will only return the real IP address I_M of machine M, there being no other information available to the update task 40 by which any other result could be obtained from the INARP task 50; clearly, something additonal needs to be done for update task 40 to be able to associate the virtual IP address $I_{M(j)}$ with the newly created SVC(q) in table 27. In fact, there are a number of ways in which the update task could be informed that the IP address to be associated with SVC(q) is $I_{M(j)}$. For example, the update task 40 could be arranged to send a request back over the newly-created SVC(q) asking machine M to identify the destination IP address $I_{T(i)}$ it associates with that SVC; from this information, the update task could determine the associated virtual IP address $I_{M(j)}$ of machine M (assuming there is a predetermined relation between the two as is the case in the described embodiment) and then update table 27 accordingly. An alternative approach that avoids sending a special request to machine M is to wait for machine M to supply the destination IP address $I_{T(i)}$ in the first IP datagram 25A sent over the new SVC(q), the update task then deriving the required address $I_{M(j)}$ as described above.

A variant of this latter approach is to leave the update task 40 unchanged but provide an additional process that:

(a) delays the INARP request until the destination address $I_{T(i)}$ of the first datagram from machine M to machine T can be captured;
(b) uses the captured address $I_{T(i)}$ as the source address of the INARP request that is now sent on to machine M.

The INARP response from machine M will therefore have a destination address $I_{T(i)}$ and a source address (that forms the substance of the INARP response) of I_M. By ensuring that this response datagram is subject to the processing effected by process 70, the source data in the INARP response will be changed to $I_{M(j)}$ by the time the response reaches the update task 40. Thus, the required updating of the table 27 of machine T can be achieved without modification to the existing tasks of machines M and T but simply by the addition of a further process for effecting steps (a) and (b) described above. This approach is the preferred one for updating table 27 and is the one used in the module described below with reference to Figures 6 and 7.

The above-described system involving the allocation of multiple virtual IP addresses to machines M and T and the provision of the address-changing processes 70 and 71, permits multiple SVCs to be concurrently operated between the machines M and T thereby enabling implementation of the test arrangement depicted in Figure 4. Of course, when testing the machine M, it is desirable that no changes are made to this machine; accordingly, it is preferred for such a test arrangement to implement the address-changing processes 70 and 71 in machine T.

The implementation of the address-changing processes 70 and 71, and of the INARP request modification process, can conveniently by done by inserting a module (hereinafter called the VNS module) between the IP/ATM layer 21 and the ATM layer 22 of machine T; in fact, an instance of this module is created for each SVC, this being relatively easy to implement when using a STREAMS type I/O implementation as provided in most UNIX systems (conveniently one stream is provided for each SVC and the VNS module is pushed onto each stream when the stream is created).

The messages passing across the boundary between layers 21 and 22 have already been described above with reference to Figure 2 and the processing effected by the VNS module on each of these messages will next be described. First, the situation of Figure 5 will be considered where it is machine M that initiates the setting up of a new SVC to machine T. The first message received by the VNS module will be the SVC setup indication message $X5^R$ and this is passed through the VNS module without modification (see Figure 6). Next, the INARP request $X6^T$ is received and is subject to the modification process 82 described above, namely it is delayed until the first IP datagram 25A is received and the address $I_{T(i)}$ extracted and used for the source address of the INARP request. The INARP response $X7^R$ is then received and subject to the address-changing process 70.IP datagrams $X8^R$ from machine M to machine T are also subject to the address-changing process 70. IP datagrams $X8^T$ from machine T to machine M are subject to address-changing process 71.

Figure 7 depicts the processing effected by the VNS module in the situation where it is the machine T rather than the machine M that initiates SVC setup. The messages passing through the VNS module in this case are those shown crossing the boundary between layers 21 and 22 in Figure 2 for machine M. The first four messages $X1^T$, $X3^T$, $X2^R$, and $X4^R$ are passed through without modification. The INARP request received from machine M is subject to the modification process 82, being delayed until the destination address of the first IP datagram from machine T to machine M can be captured and used as the source address of the INARP request. The INARP response $X7^T$ is subjected to process 71 as are IP datagrams $X8^T$ from machine T to machine M. IP datagrams $X8^R$ from machine M to machine T are subjected to process 70.

It will be appreciated that many variants are possible to the above-described embodiment of the invention. It will also be appreciated that the invention is not limited to switched virtual circuits but can equally be applied to permanent virtual circuits. Furthermore, the setting up of multiple virtual circuits between two machines can be used not only for implementing the test arrangement described above with reference to Figure 4 but also for other purposes.

Although the present invention has been described in the context of high-level addresses constituted by IP addresses and virtual circuits set up across an ATM network, the invention can be applied to other types high-level addresses and other types of virtual-circuit network. For example, the high-level addresses could be MAC addresses in the case of a network in the form of a emulated LAN (ELAN) over an ATM network.

Claims

1. A system in which a plurality of entities are con-

nected to a network and can exchange messages across virtual circuits set up over the network between said entities, each entity having an operative high-level address on the network, and each said entity comprising:

-- high-level messaging means for handling message transmission and receipt on the basis of said high-level addresses, said high-level messaging means comprising means for including in outgoing ones of said messages the operative high-level address of the entity as a source identifier and the operative high level address of the intended recipient entity as a destination identifier, and means for filtering incoming ones of said messages according to the destination identifier contained in the message:
-- virtual-circuit means for providing virtual circuits between the entity and other said entities, there being a respective virtual circuit for each different destination identifier in use, and
-- intermediate means for passing an outgoing message from said high-level messaging means to that one of the virtual circuits provided by the virtual-circuit means which corresponds to the destination identifier of the message;

characterised in that each of a first and a second said entity has a plurality of virtual high-level addresses associated with it that are different from said operative high-level address of the entity, said virtual high-level addresses being usable by the messaging means of said first and second entities as destination identifiers in outgoing messages; **and in that** between said intermediate means of said first and second entities, there are provided address-changing means responsive to each of at least some of said messages sent between these entities with a said virtual high-level address as its destination identifier to change that address to the said operative high-level address of the corresponding entity, and to change the operative high-level address provided as the source identifier of the message into one of the said virtual high-level addresses associated with the sending entity in dependence on the virtual high-level address initially provided as the destination identifier of the same message.

2. A system according to claim 1, wherein said address-changing means effects a predetermined transformation on the virtual high-level address forming the initial destination identifier of a said message to which the address-hanging means is responsive in order to form the virtual high-level address to be used for the source identifier of that message.

3. A system according to claim 2, wherein said address-changing means is responsive to messages sent in both directions between said first and second entities with virtual high-level addresses as destination identifiers, the said transformation effected in respect of such messages sent in one said direction being the reverse of the transformation effected in respect of other such messages sent in the opposite said direction.

4. A system according to claim 1, wherein said address-changing means comprises first address-changing functionality for effecting said changes for messages sent from said first entity to said second entity, and second address-changing functionality for effecting said changes for messages sent from said second entity to said first entity, both said first and second address-changing functionalities being provided in said second entity.

5. A system according to claim 1, wherein said address-changing means comprises first address-changing functionality for effecting said changes for messages sent from said first entity to said second entity, and second address-changing functionality for effecting said changes for messages sent from said second entity to said first entity, the two said address-changing functionalities being provided in respective ones of said first and second entities.

6. A system according to claim 1, wherein:

-- each said entity has a low-level address on the network;
-- said intermediate means of each entity further comprises:

-- first association means for providing an association between the destination identifier of a outgoing message and the low-level address of the corresponding said entity,
-- second association means for providing an association between the destination identifier of an outgoing message and a said virtual circuit,

said intermediate means using its second association means to identify from the destination identifier of a said outgoing message which virtual circuit is to be passed the message where such virtual circuit exists, and otherwise first passing a request to the said virtual circuit means of the same entity to establish a virtual circuit to the entity having the low-level address identified by said first association means as associated with the destination identifier of the outgoing message; and

-- the said virtual-circuit means of each entity includes setup means responsive to a said request from the intermediate means of the same entity to establish a virtual circuit to the said entity having the low-level address provided in said request, said setup means causing the intermediate means to update its second association means to associate the newly-established virtual circuit with the said destination identifier relevant to said request;

the first association means of each of said first and second entities serving to provide an association between the virtual high-level addresses of the other of said first and second entities and the low-level address of that other entity.

8. A system according to claim 7, further comprising a network server containing associations between high-level addresses and low-level addresses, said first association means of each said entity comprising means for interrogating said network server for a required association.

9. A system according to claim 7, wherein said second association means comprises cache means for temporarily holding said associations between said destination identifiers and currently corresponding virtual circuits.

10. A system according to any one of claims 1 to 9, wherein said high-level addresses are IP addresses and said network is a ATM network.

11. A system according to any one of claims 1 to 9, wherein said high-level addresses are MAC addresses and said network is a emulated LAN over an ATM network.

12. A method of testing the ability of network node apparatus to operate a plurality of virtual circuits at the same time, said network node apparatus being arranged to establish a virtual circuit for each different high-level destination address being handled, said method involving setting up a system according to claim 4 with said network node apparatus as said first entity, and causing the network node apparatus to send messages to at least some of said virtual high-level addresses associated with said second entity.

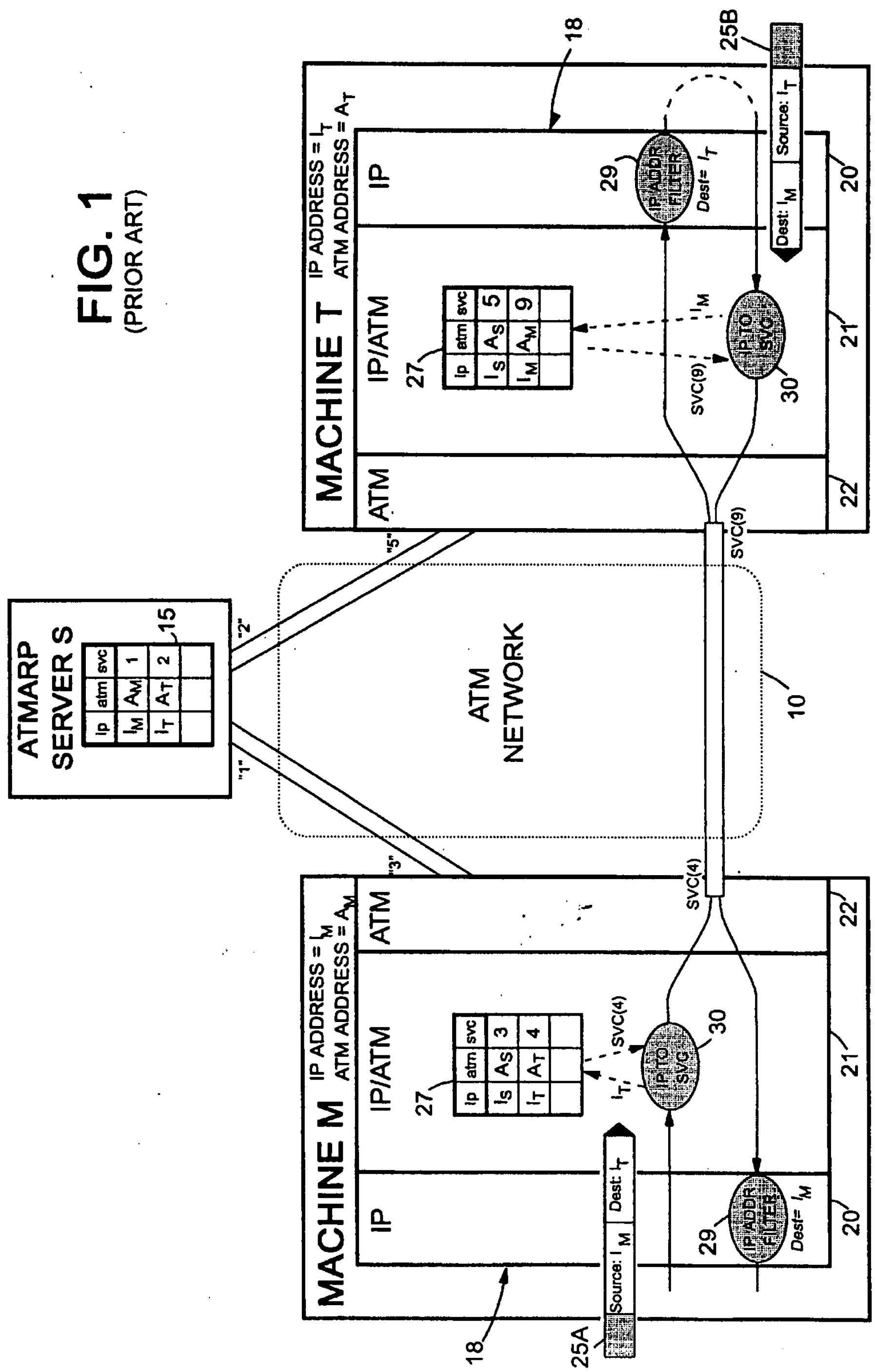


FIG. 1
(PRIOR ART)

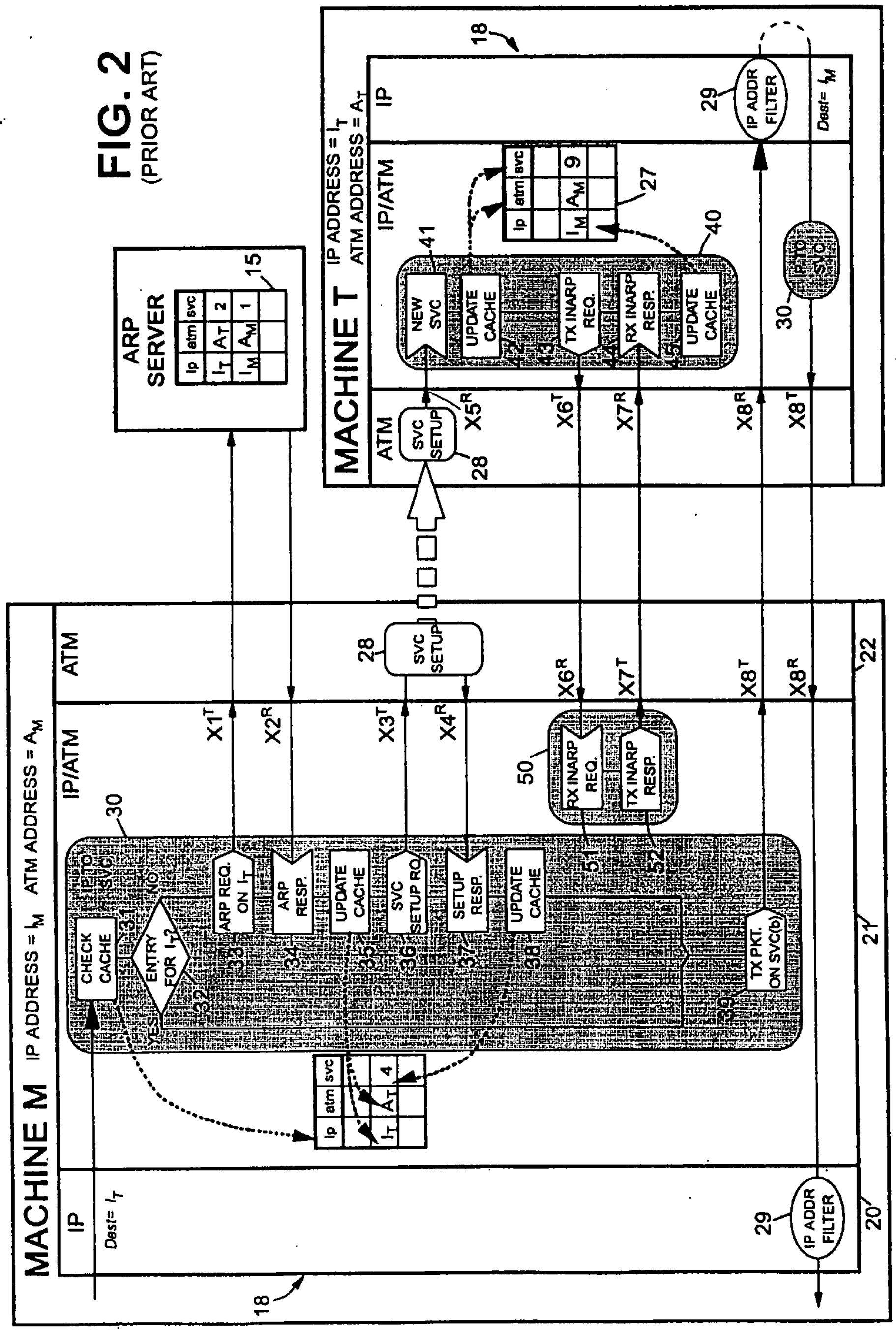


FIG. 2
(PRIOR ART)

FIG. 3
(PRIOR ART)

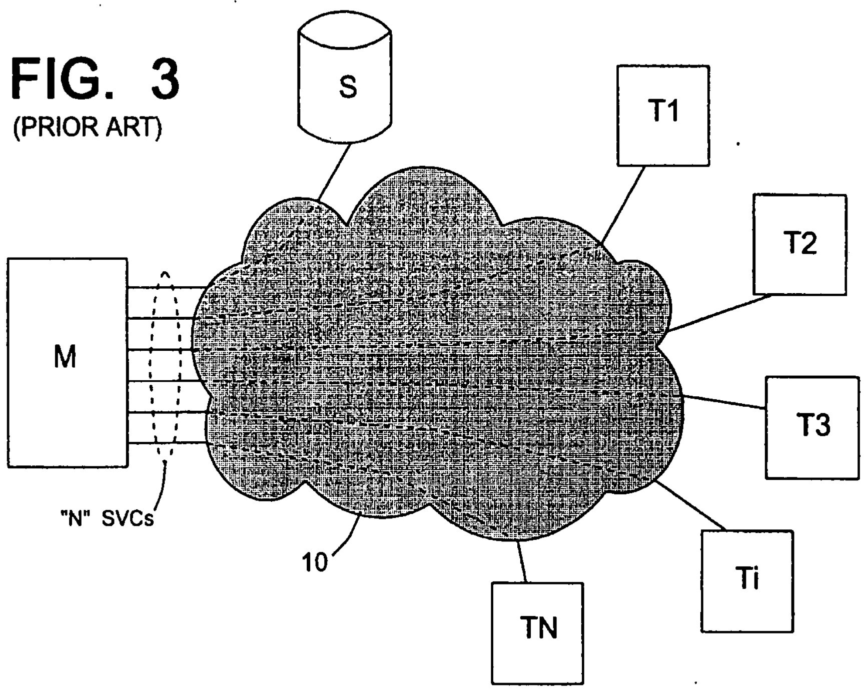


FIG. 4

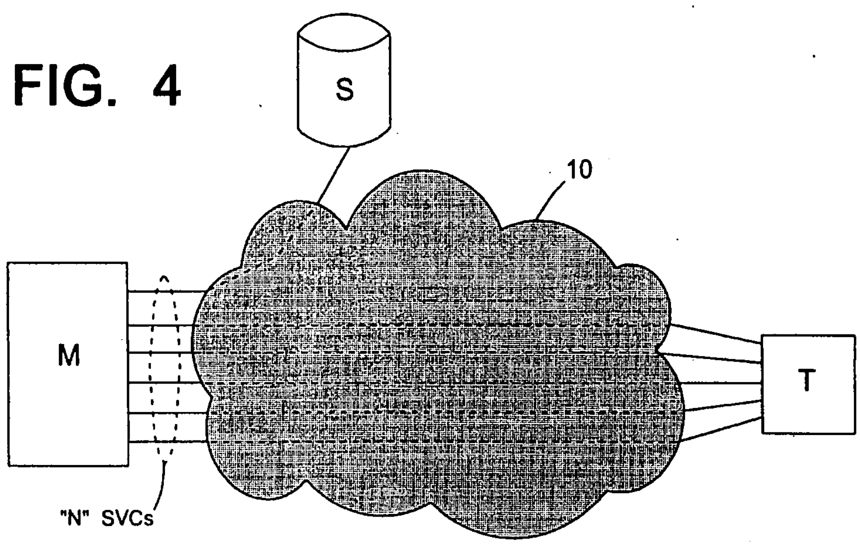
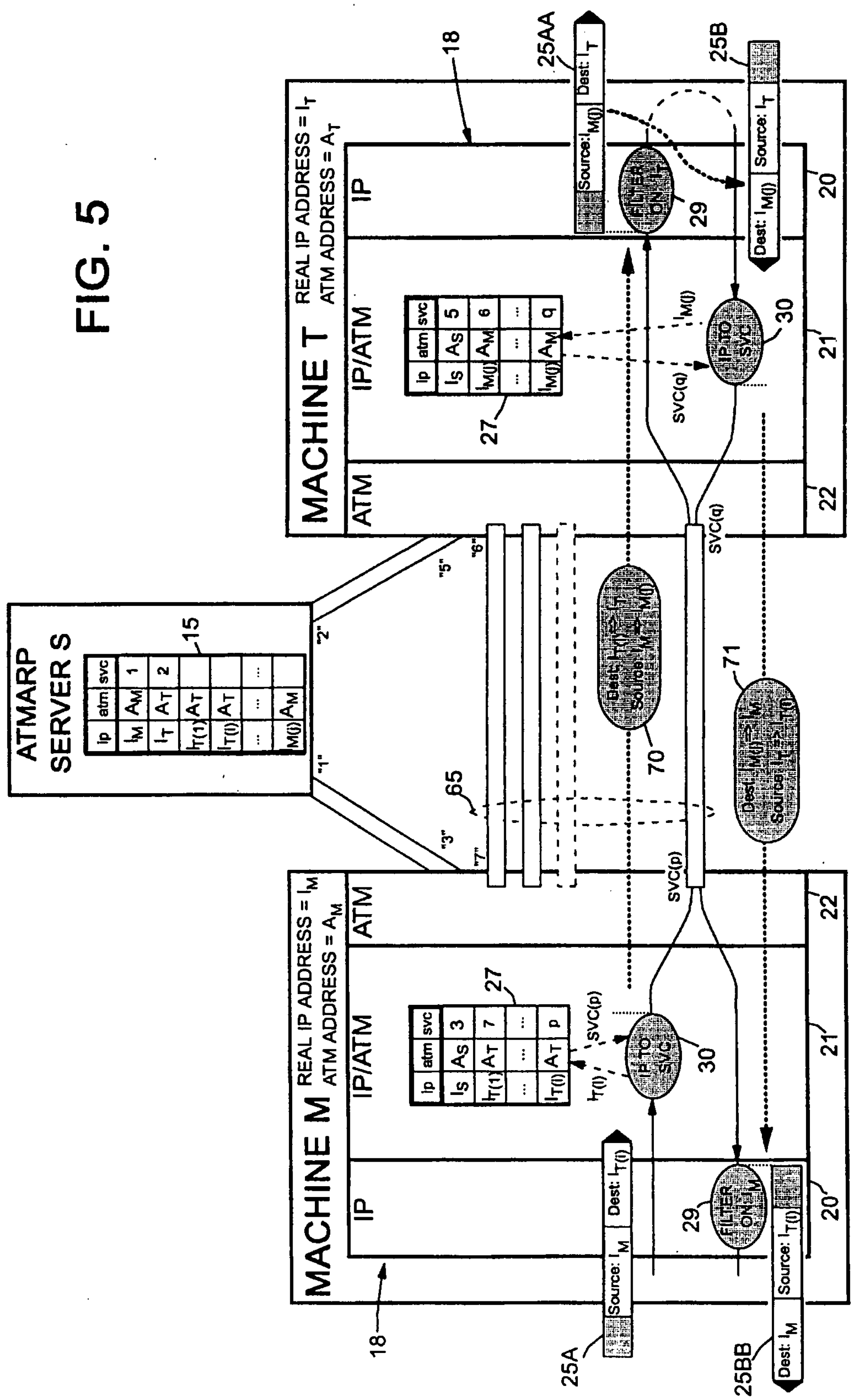
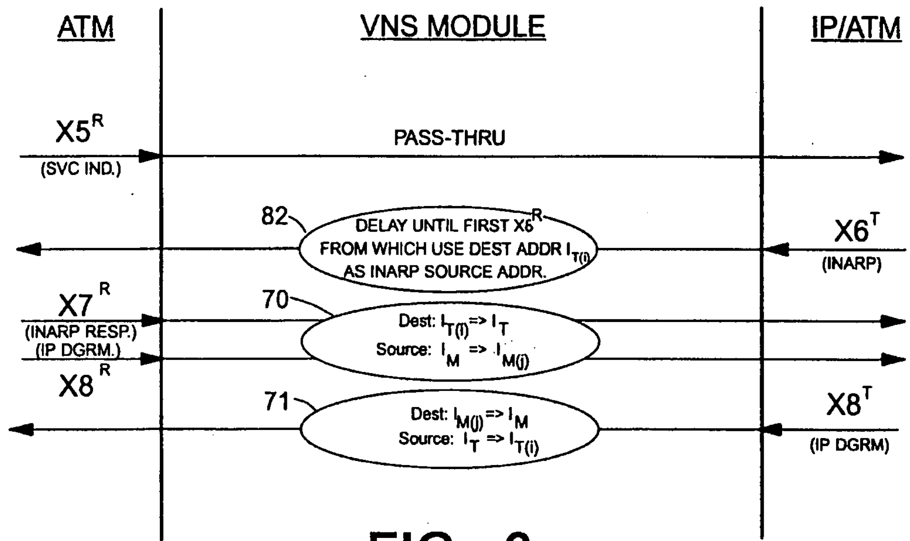


FIG. 5

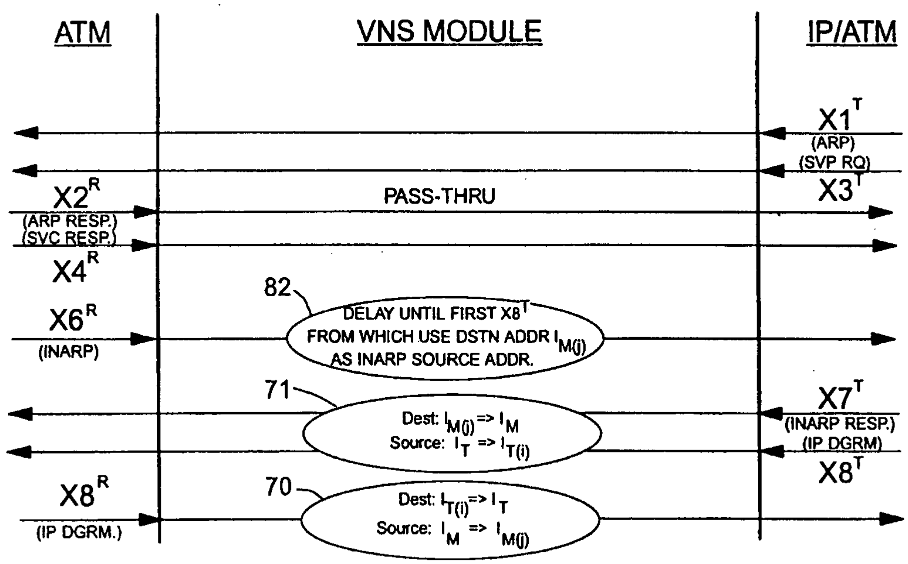


ATMARP SERVER S
MACHINE T
REAL IP ADDRESS = I_T
ATM ADDRESS = A_T
MACHINE M
REAL IP ADDRESS = I_M
ATM ADDRESS = A_M
IP
IP/ATM
ATM
FILTER ON I_T
FILTER ON I_M
IP TO SVC
SVC(p)
SVC(q)
Source: $I_{M()}$ Dest: I_T
Source: I_T Dest: $I_{M()}$
Source: I_M Dest: $I_{T()}$
Source: $I_{T()}$ Dest: I_M
18
20
21
22
27
29
30
15
65
70
71
25A
25AA
25B
25BB



ATM
VNS MODULE
IP/ATM
X5R
(SVC IND.)
PASS-THRU
82
DELAY UNTIL FIRST X6R
FROM WHICH USE DEST ADDR IT(i)
AS INARP SOURCE ADDR.
X6T
(INARP)
X7R
(INARP RESP.)
(IP DGRM.)
X8R
70
Dest: IT(i) => IT
Source: IM => IM(j)
71
Dest: IM(j) => IM
Source: IT => IT(i)
X8T
(IP DGRM)

FIG. 6



ATM
VNS MODULE
IP/ATM
X1T
(ARP)
(SVP RQ)
X3T
X2R
(ARP RESP.)
(SVC RESP.)
X4R
PASS-THRU
82
DELAY UNTIL FIRST X8T
FROM WHICH USE DSTN ADDR IM(j)
AS INARP SOURCE ADDR.
X6R
(INARP)
71
Dest: IM(j) => IM
Source: IT => IT(i)
X7T
(INARP RESP.)
(IP DGRM)
X8T
70
X8R
(IP DGRM.)
Dest: IT(i) => IT
Source: IM => IM(j)

FIG. 7

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 96 41 0106

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	COMPUTER COMMUNICATIONS REVIEW, vol. 25, no. 4, 1 October 1995, pages 49-58, XP000541650 PARULKAR G ET AL: "AITPM: A STRATEGY FOR INTEGRATING IP WITH ATM" * paragraph 2.1 *	1,10	H04L29/06 H04Q11/04
A	PROCEEDINGS OF THE ANNUAL SYMPOSIUM ON FOUNDATIONS OF COMPUTER SCIE, SANTA FE, NOV. 20 - 22, 1994, no. SYMP. 35, 20 November 1994, GOLDWASSER S (EDITOR), pages 424-434, XP000531950 LUND C ET AL: "IP OVER CONNECTION-ORIENTED NETWORKS AND DISTRIBUTIONAL PAGING" * paragraph 1 - paragraph 1.1 *	1,10	
A	DATA COMMUNICATIONS, vol. 24, no. 17, 1 December 1995, page 103/104, 106, 108, 110 XP000547618 MARSHALL G: "CLASSICAL IP OVER ARM:A STATUS REPORT" paragraph "Simple virtues""	1,10,11	TECHNICAL FIELDS SEARCHED (Int.Cl.6) H04L H04Q
A	IBM TECHNICAL DISCLOSURE BULLETIN, vol. 35, no. 4A, 1 September 1992, pages 28-31, XP000314666 "COORDINATED ADDRESS RESOLUTION PROTOCOL PROCESSING" * the whole document *	6,8,9	
A	EP 0 523 386 A (FUJITSU) * page 4, line 20 - page 5, line 14; figure 3 *	12	

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	13 March 1997	Staessen, B

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

B3

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04Q 11/04, H04L 12/22	A1	(11) International Publication Number: WO 98/27783 (43) International Publication Date: 25 June 1998 (25.06.98)

(21) International Application Number: PCT/IB97/01563

(22) International Filing Date: 12 December 1997 (12.12.97)

(30) Priority Data:
08/769,649 19 December 1996 (19.12.96) US

(71) Applicant (*for all designated States except US*): NORTHERN TELECOM LIMITED [CA/CA]; World Trade Center of Montreal, 8th floor, 380 St. Antoine Street West, Montreal, Quebec H2Y 3Y4 (CA).

(72) Inventors; and
(75) Inventors/Applicants (*for US only*): TELLO, Antonio, G. [US/US]; 114 Fountain Hills Drive, Garland, TX 75044 (US). HUI, Margaret [US/US]; 9920 Forest Lane #208, Dallas, TX 75243 (US). HOLMES, Kim [US/US]; 5409 Scenic Drive, Rowlett, TX 75088 (US).

(74) Agents: MCCOMBS, David et al.; Haynes and Boone, L.L.P., Suite 3100, 901 Main Street, Dallas, TX 75202-3789 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: VIRTUAL PRIVATE NETWORK SERVICE PROVIDER FOR ASYNCHRONOUS TRANSFER MODE NETWORK

(57) Abstract

A virtual private network service provider is used to transfer data over a data network to a final destination, with third-party billing. The method comprises the steps of: prompting the user at a data terminal to select a destination, password, and call type; sending a set-up message to the data network; selecting a virtual private network provider through the data network; the virtual private network provider giving an encryption key to the user, and then prompting the user for a password and a user identification; encrypting the password, and sending the user identification and the encrypted password to the virtual private network provider; the virtual private network provider decrypting the encrypted password, and verifying the password; the virtual private network provider providing an authorization code; and the data terminal transferring the data through the data network to the final destination, using the authorization code.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

VIRTUAL PRIVATE NETWORK SERVICE PROVIDER FOR ASYNCHRONOUS TRANSFER MODE NETWORK

Technical Field

The invention relates generally to asynchronous transfer mode ("ATM") networks and virtual private networks ("VPN"), such as those offered by MCI and Sprint, and, more particularly, to a method of using a VPN to transfer data over a data network, with third-party billing.

Background of the Invention

Telephone service providers offer third-party billing. For example, local and long distance telephone companies offer calling cards for third party billing.

VPNs exist to provide the sense of a private network among a company's locations. The lines/trunks of a VPN are actually shared among several companies, to reduce costs, yet to each company the VPN appears to be that company's own private network. However, a user at a remote data terminal, such as a portable computer in a hotel room, can not immediately charge his company for the access time to a data net, such as the Internet. Instead, his access time is charged to his hotel room, and so he must pay the inflated rates that hotels charge for phone service.

What is needed is a VPN service provider that offers remote access for users belonging to a VPN, user authorizations to prevent delinquent access into the VPN, and convenient third-party billing.

Summary of the Invention

The present invention, accordingly, provides a system and method for using a VPN service provider to transfer data over a data network to a final destination, with third-party billing. The method comprises the steps of: prompting the user at a data terminal to select a destination, password, and call type; selecting a VPN through the data network; giving an encryption key to the user, and then prompting the user for a password and a user identification; verifying the password, and providing an authorization code to the user; and allowing the user to transfer the data through the data network to the final destination, using the authorization code.

In another feature of the invention, the method further comprises negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

In another feature of the invention, the method further comprises encrypting the user's password, and sending the user identification and the encrypted password to the VPN service provider.

In another feature of the invention, the method further comprises a step of sending a set-up message to the data network.

In another feature of the invention, the method further comprises a step of the VPN service provider decrypting the encrypted password.

A technical advantage achieved with the invention is that it shifts or defers costs from an end user to a bulk purchaser of data network services. Another technical advantage achieved with the invention is that it permits end users mobility while attaining a virtual appearance on a corporate intranet.

Brief Description of the Drawings

Fig. 1 is a system block diagram of a VPN service provider of the present invention.

Fig. 2 is a flow chart depicting the method of the present invention, as implemented by application software on a user terminal.

Fig. 3 is the initial screen display of the user interface of the application software.

Figs. 4A and 4B are call flow diagrams, illustrating the preferred sequence of steps of the method of the present invention.

Figs. 5A, 5B, 5C, 5D, 5E, and 5F comprise a flow chart depicting the method of the present invention, as implemented by switching control point software.

Description of the Preferred Embodiment

In Fig. 1, the VPN service provider system of the present invention is designated generally by a reference numeral 10. The VPN service provider system 10 includes a VPN 12. The VPN 12 may be a corporate, government, association, or other organization's telephone/data line `network. The VPN service provider system 10 also includes access lines 13 from the VPN 12 to a data network 14, such as the Internet, or an ATM network. The VPN service provider system 10 also includes access lines 16 from the data network 14 to a long distance phone company 18, such as AT&T, MCI, or Sprint. The VPN service provider system 10 also includes access lines 20 from the data network 14 to a called party 22, such as, for example, American Express reservations service. The VPN service provider system 10 also includes access lines 24 from the data network 14 to a remote user terminal 26, such as a portable computer in a hotel room. The user terminal 26 includes user application software 28, which provides the interface for the user to enter the number to be called, the user identification number, and the user's authorization code. The VPN service provider system 10 also includes VPN service provider software 30, located in a switching control point (SCP) device 32, which, in the preferred embodiment may be physically located anywhere. The SCP 32 connects to the data network 14 via access lines 36. One possible physical location for the SCP 32 is on the premises of a local phone company central switch building 34. However, even when located within the building 34, the SCP 32 connects to the local phone company switches via the data network 14. The local phone company switches connect to the data network 14 via access lines 38.

In an alternate embodiment, the VPN service provider software 30 and the SCP device 32 may be located on the premises of an independent provider of local phone service, or on the premises of an independent VPN service provider.

Referring now to Fig. 2, the application software 28 begins the data transfer process in step 50. In step 52, the user is presented with a screen display.

Referring now to Fig. 3, a screen display 100 displays the following information requests: whether the call is a direct call 102 or a VPN call 104, the number the user desires to call 106, the VPN user ID 108, and the user password 110. The user is also presented with the option to make the call 112, or to quit 114.

Referring back to Fig. 2, in step 54 the user terminal sends to the SCP 32 the information captured through the graphical user interface ("GUI") in step 52 within a user network interface ("UNI") setup message. In step 56 the user terminal 26 waits for a connect message from the SCP 32. In step 58 the user terminal 26 determines if a connection was made. If no connection was made, then in step 60 the user application software 28 displays an error message to the user, and returns to step 50 to begin again the data transfer process.

If a connection was made, then in step 62 the user terminal 26 sends the VPN user ID to the SCP 32. In step 64 the user terminal 26 waits for an encryption key from the SCP 32. In step 66, having received the encryption key from the SCP 32, the user application software 28 encrypts the user's password, and sends it to the SCP 32. In step 68 the user terminal 26 waits for authentication of the user. In step 70 the user application software 28 determines if the SCP 32 authorizes the user to make the call.

If the user is not authorized, then in step 72 the user terminal 26 displays an error message, terminates the connection, blanks the screen display 100, and returns to step 50 to begin again the data transfer process. If the user is authorized, then in step 74 the VPN service provider software 30 sets up the billing, and authorizes it. In step 76 the user terminal 26 sends a "release", meaning to terminate or disconnect the connection, to the SCP 32. In step 78 the user terminal 26 sends a setup message to the number listed by the user as the "number to call", that is, to the final destination. In step 80 the user terminal 26 waits for a connection. In step 82 the user terminal 26 determines if a connection was made.

If a connection to the final destination was not made, then the user application software 28 returns to step 72, in which step the user terminal 26 displays an error message, terminates the connection, blanks the screen display 100, and returns to step 50 to begin again the data transfer process. If a connection to the final destination was made, then in step 84 the user terminal 26 exchanges user data, services, and/or value added or user specific applications with the computer at the address, that is, the telephone number, of the final destination. In step 86 the user selects the option presented to him to release, or terminate, the call. In step 88 the user terminal 26 sends a release message to the final destination. In step 90 the data network 14 sends billing information to the SCP 32. In step 92 the application software 28 ends the data transfer process.

Fig. 4A and Fig. 4B are call flow diagrams, showing the sequence of messages in the method of the preferred embodiment. These diagrams present the same method as the flow chart of Fig. 2. The horizontal arrows represent the messages sent and received. The vertical lines represent the various devices involved in sending and receiving the messages. For example, the top left arrow in Fig. 4A represents a message sent from the user terminal 26, labeled "Macintosh" in Fig.4A, to an interface with a public network. The user terminal 26 can be any brand of a work station computer, a desktop computer, a laptop computer, or even a notebook computer. The interface could be any interface, but in the example of Fig. 4A and Fig. 4B, the interface is imagined to be at a hotel, where a business traveler is using the method of the present invention. Thus, the interface is labeled "Hotel ATM Interface", which is not shown in Fig. 1. The vertical line labeled "Public ATM Network" is the same as the data network 14 in Fig. 1. The vertical line labeled "Moe's VPN Service" represents the VPN service provider software 30 within the SCP 32. The vertical line labeled "Travel ATM Interface" is not shown in Fig. 1, but is located between the called party 22 and the data network 14. The vertical line labeled "Travel Service" is one example of the called party 22 shown in Fig. 1. In the example of Fig. 4A and Fig. 4B, the business traveler is imagined to be using the method of the present invention to contact a travel service to make reservations for his next airline flight. In Figs. 4A and 4B the designation "Ack" represents "acknowledge", and the designation "Cmp" represents "complete".

Referring now to Fig. 5, the VPN service provider software 30 begins the data transfer process in step 300 by waiting for an event. The event it waits for is a setup message on a signaling port of the SCP 32, to be received from the user terminal 26. In step 302, having monitored the signaling ports, and the SCP 32 having received a setup message, the VPN service provider software 30 assigns a call condense block ("CCB") to the setup message, based on a call reference number. The CCB is a software data structure for tracking resources associated with the call. The call reference number is a number, internal to the SCP, for tracking calls. In step 304 the VPN service provider software 30 compiles the connect message. In step 306 the VPN service provider software 30 sends a connect message to the calling address, that is, the hotel room from which the user is calling. In step 308 the VPN service provider software 30 condenses, that is, it remains in a wait state for that call.

Referring now to Fig. 5B, in step 310 the VPN service provider software 30 waits for an event by monitoring the signaling ports of the SCP 32. After the SCP 32 receives a connect acknowledge message from the user terminal 26, then in step 312 the VPN service provider software 30 accesses the CCB, based on the call reference number. In step 314 the VPN service provider software 30 condenses.

Referring now to Fig. 5C, in step 316 the VPN service provider software 30 waits for dialog on a data port of the SCP 32. After the SCP 32 receives a

VPN ID on a data port, the VPN service provider software 30 verifies the VPN ID in step 318. In step 320 the VPN service provider software 30 determines if the VPN ID is valid. If the VPN ID is not valid, then in step 322 the SCP 32 sends a reject message over an assigned switch virtual circuit ("SVC"). The SVC is a channel over the data network 14. In step 324 the VPN service provider software 30 waits for dialog. In step 326, because the VPN ID is valid, the VPN service provider software 30 assigns an encryption key to the user terminal 26, in step 328 sends the encryption key over the assigned SVC to the user terminal 26, and in step 330 waits for dialog.

Referring now to Fig. 5D, in step 332 the VPN service provider software 30 waits for dialog. When the SCP 32 receives the encrypted password from the user terminal 26 at a data port, then in step 334 the VPN service provider software 30 verifies the password, and determines in step 336 if the password is valid. If the password is not valid, then in step 338 the SCP 32 sends a reject message over the assigned SVC to the user terminal, and in step 340 waits for dialog. If the password is valid, then in step 342 the VPN service provider software 30 assigns an authorization token to the user terminal 26, in step 344 sends the token over an assigned SVC to the user terminal 26, and in step 346 waits for dialog.

Referring now to Fig. 5E, in step 348 the VPN service provider software 30 waits for an event. When the VPN service provider software 30 senses that the SCP 32 has received on a signaling port a release message from the user terminal 26, then in step 350 the VPN service provider software 30 accesses the CCB, based on the call reference number of the user terminal 26, in step 352 compiles a release complete message, in step 354 sends a release complete message to the user terminal 26, and in step 356 condenses.

Referring now to Fig. 5F, in step 358 the VPN service provider software 30 waits for an event. When the VPN service provider software 30 senses that the SCP 32 has received on a signaling port a third-party billing setup message from the user terminal 26, then in step 360 the VPN service provider software 30 verifies the token just received from the user terminal 26, to determine, in step 362, if it is the same token that the VPN service provider software 30 sent to the user terminal 26 in step 344. If the token is not valid, then in step 364 the SCP 32 sends a release message to the terminal 26, and in step 366 condenses. If the token is valid, then in step 368 the SCP 32 sends a modified third-party billing setup message to the data network 14, and in step 370 condenses.

Although an illustrative embodiment of the invention has been shown and described, other modifications, changes, and substitutions are intended in the foregoing disclosure. Accordingly, it is appropriate that the appended claims be construed broadly and in a manner consistent with the scope of the invention.

WHAT IS CLAIMED IS:

1. A computerized method of a virtual private network service provider with third party billing, using a virtual private network to transfer data over a data network to a final destination, the method comprising the steps of:

   a. prompting the user at a data terminal to select a destination, password, and call type;
   b. selecting a virtual private network through the data network;
   c. giving an encryption key to the user, and then prompting the user for a password and a user identification;
   d. verifying the password, and providing an authorization code to the user; and
   e. allowing the user to transfer the data through the data network to the final destination, using the authorization code.

2. The method of claim 1, wherein step (d) further comprises negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

3. The method of claim 2, wherein step (c) further comprises encrypting the user's password, and sending the user identification and the encrypted password to the virtual private network service provider.

4. The method of claim 3, further comprising, after step (a), the step of sending a set-up message to the data network.

5. The method of claim 4, further comprising, after step (c), the step of the virtual private network service provider decrypting the encrypted password.

6. An apparatus for providing a datalink connection from a user terminal to a data network and to a virtual private network, with third party billing, comprising:

   a. an interface between the user terminal and the data network;

b. a switching control point device connected to the data network, the switching control point device connected to a computer; and

c. a computer-readable medium encoded with a method of using the virtual private network and the data network, with third party billing, the computer-readable medium accessible by the computer.

7. The apparatus of claim 6, wherein the method comprises negotiating for more bandwidth for the user, and including within an authorization code a grant of additional bandwidth.

8. The apparatus of claim 7, wherein the method further comprises encrypting a user's password, and temporarily storing the user identification and the encrypted password.

9. The apparatus of claim 8, wherein the method further comprises sending a set-up message to the data network.

10. The apparatus of claim 9, wherein the method further comprises decrypting the encrypted password.

11. A computer-readable medium encoded with a method of using a virtual private network, with third party billing, the method comprising the steps of:

a. prompting the user at a data terminal to select a destination, password, and call type;

b. selecting a virtual private network through the data network;

c. giving an encryption key to the user, and then prompting the user for a password and a user identification;

d. verifying the password, and providing an authorization code to the user; and

e. allowing the user to transfer the data through the data network to the final destination, using the authorization code.

12. The computer-readable medium of claim 11 wherein step (d) further comprises negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

13. The computer-readable medium of claim 12 wherein step (c) further comprises encrypting the user's password, and sending the user identification and the encrypted password to the virtual private network service provider.

14. The computer-readable medium of claim 13 further comprising, after step (a), the step of sending a set-up message to the data network.

15. The computer-readable medium of claim 14 further comprising, after step (c), the step of the virtual private network service provider decrypting the encrypted password.

16. An apparatus for providing a datalink connection from a user terminal to a data network and to a virtual private network, with third party billing, comprising:

a. means for prompting a user at the data terminal to select a destination, password, and call type;

b. means for selecting the virtual private network through the data network;

c. means for giving an encryption key to the user, and then prompting the user for a password and a user identification;

d. means for verifying the password, and providing an authorization code to the user; and

e. means for allowing the user to transfer data through the data network to a final destination, using the authorization code.

17. The apparatus of claim 16, further comprising means for negotiating for more bandwidth for the user, and including within the authorization code a grant of additional bandwidth.

18. The apparatus of claim 17, further comprising means for encrypting the user's password, and sending the user identification and the encrypted password to the virtual private network service provider.

19. The apparatus of claim 18, further comprising means for sending a set-up message to the data network.

20. The apparatus of claim 19, further comprising means for decrypting the encrypted password.

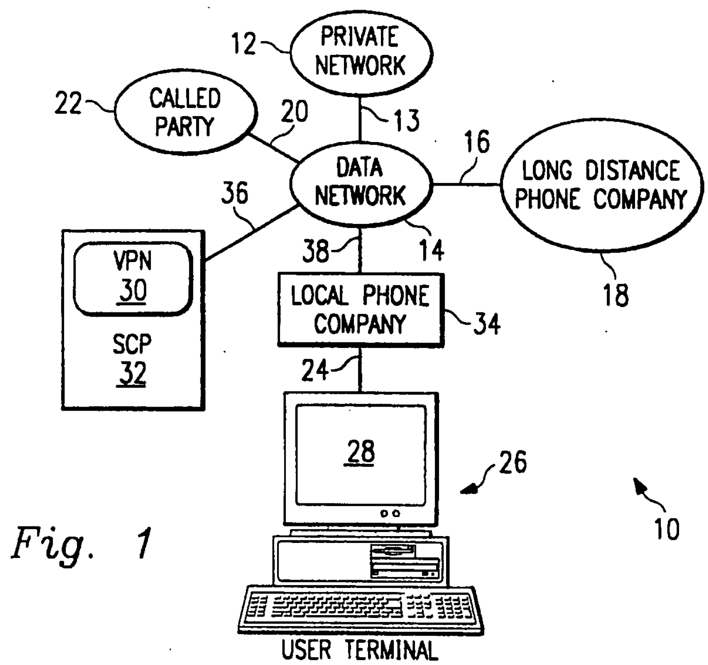

PRIVATE NETWORK
12
CALLED PARTY
22
20
13
16
DATA NETWORK
LONG DISTANCE PHONE COMPANY
36
38
14
VPN 30
SCP 32
LOCAL PHONE COMPANY
34
18
24
28
26
10
USER TERMINAL

Fig. 1

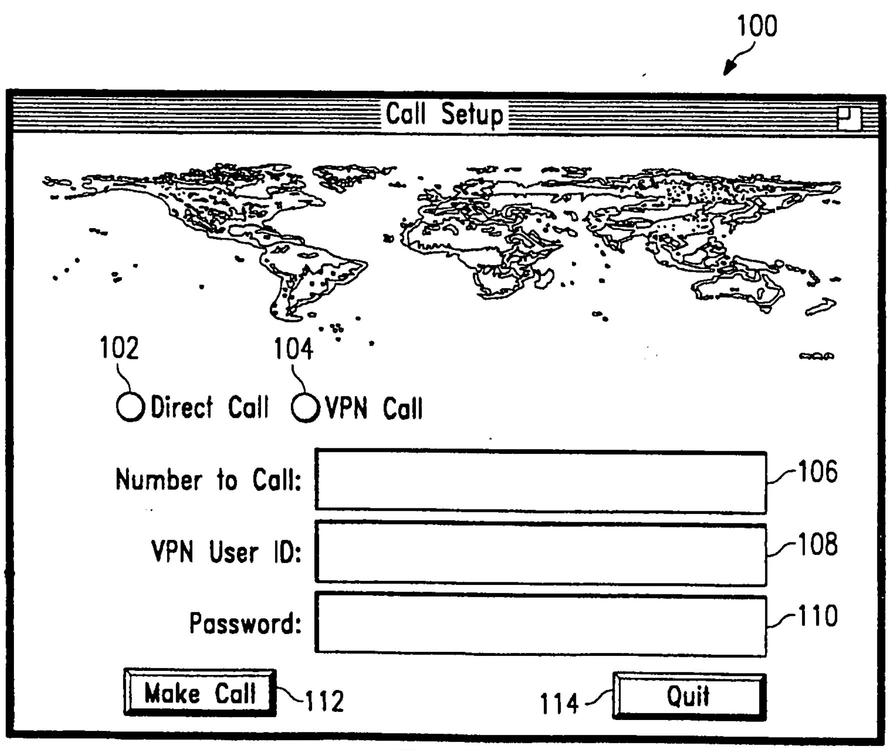
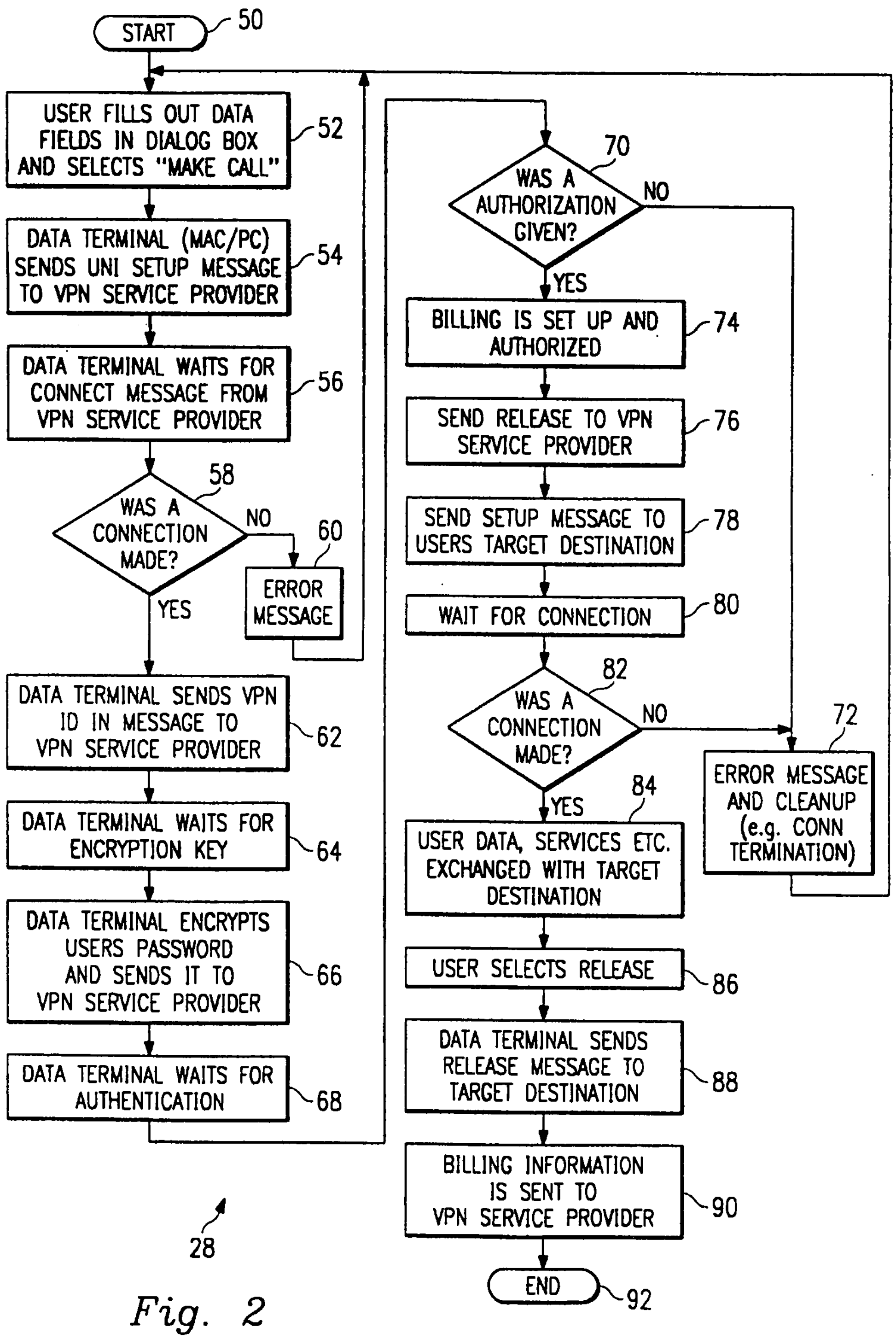

100
Call Setup
102
104
Direct Call
VPN Call
Number to Call:
106
VPN User ID:
108
Password:
110
Make Call
112
114
Quit

Fig. 3



START
50
USER FILLS OUT DATA FIELDS IN DIALOG BOX AND SELECTS "MAKE CALL"
52
DATA TERMINAL (MAC/PC) SENDS UNI SETUP MESSAGE TO VPN SERVICE PROVIDER
54
DATA TERMINAL WAITS FOR CONNECT MESSAGE FROM VPN SERVICE PROVIDER
56
WAS A CONNECTION MADE?
58
NO
YES
ERROR MESSAGE
60
DATA TERMINAL SENDS VPN ID IN MESSAGE TO VPN SERVICE PROVIDER
62
DATA TERMINAL WAITS FOR ENCRYPTION KEY
64
DATA TERMINAL ENCRYPTS USERS PASSWORD AND SENDS IT TO VPN SERVICE PROVIDER
66
DATA TERMINAL WAITS FOR AUTHENTICATION
68
WAS A AUTHORIZATION GIVEN?
70
NO
YES
BILLING IS SET UP AND AUTHORIZED
74
SEND RELEASE TO VPN SERVICE PROVIDER
76
SEND SETUP MESSAGE TO USERS TARGET DESTINATION
78
WAIT FOR CONNECTION
80
WAS A CONNECTION MADE?
82
NO
YES
ERROR MESSAGE AND CLEANUP (e.g. CONN TERMINATION)
72
USER DATA, SERVICES ETC. EXCHANGED WITH TARGET DESTINATION
84
USER SELECTS RELEASE
86
DATA TERMINAL SENDS RELEASE MESSAGE TO TARGET DESTINATION
88
BILLING INFORMATION IS SENT TO VPN SERVICE PROVIDER
90
END
92
28

Fig. 2

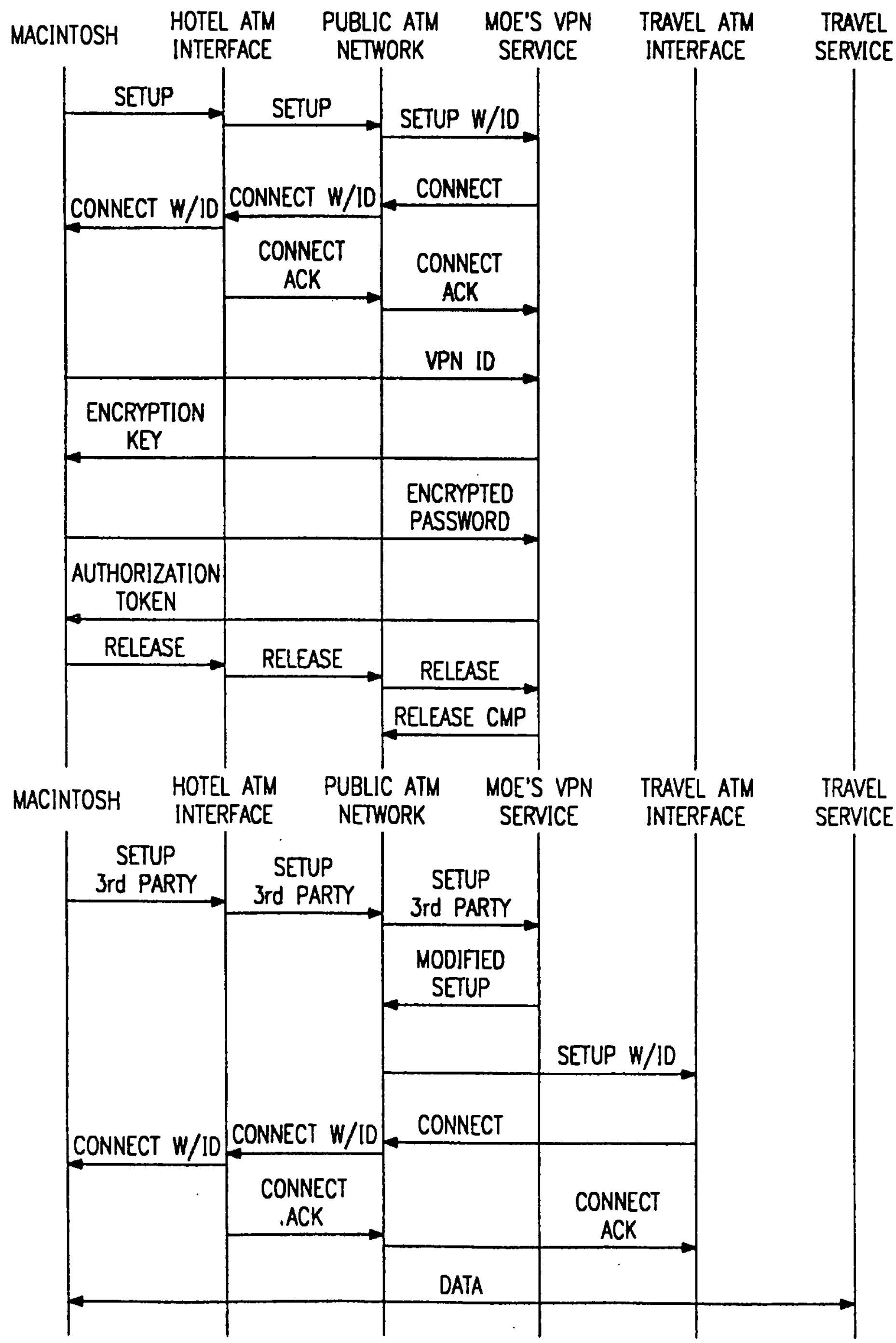

MACINTOSH
HOTEL ATM INTERFACE
PUBLIC ATM NETWORK
MOE'S VPN SERVICE
TRAVEL ATM INTERFACE
TRAVEL SERVICE
SETUP
SETUP
SETUP W/ID
CONNECT W/ID
CONNECT W/ID
CONNECT
CONNECT ACK
CONNECT ACK
VPN ID
ENCRYPTION KEY
ENCRYPTED PASSWORD
AUTHORIZATION TOKEN
RELEASE
RELEASE
RELEASE
RELEASE CMP
MACINTOSH
HOTEL ATM INTERFACE
PUBLIC ATM NETWORK
MOE'S VPN SERVICE
TRAVEL ATM INTERFACE
TRAVEL SERVICE
SETUP 3rd PARTY
SETUP 3rd PARTY
SETUP 3rd PARTY
MODIFIED SETUP
SETUP W/ID
CONNECT W/ID
CONNECT W/ID
CONNECT
CONNECT .ACK
CONNECT ACK
DATA

Fig. 4A

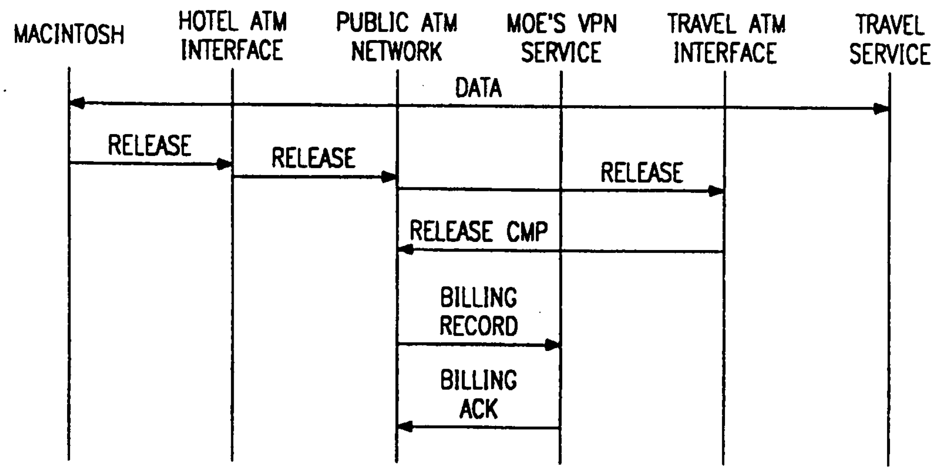


Fig. 4B

Fig. 5A

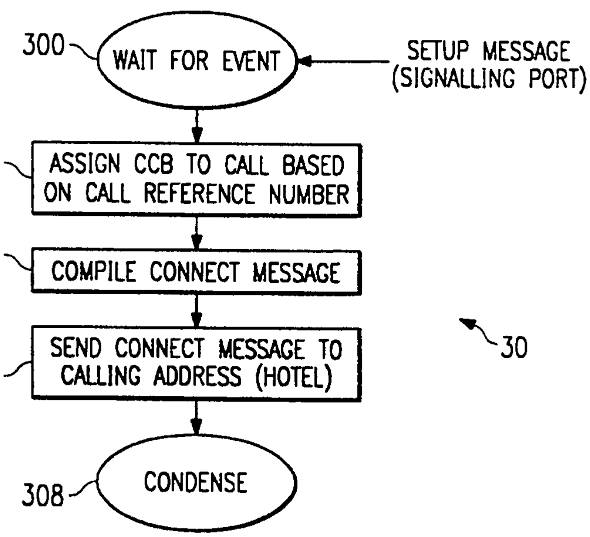
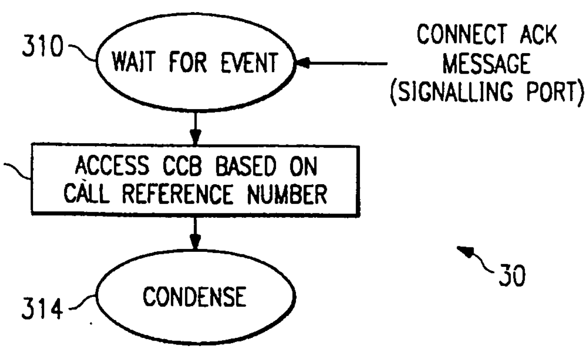


Fig. 5B



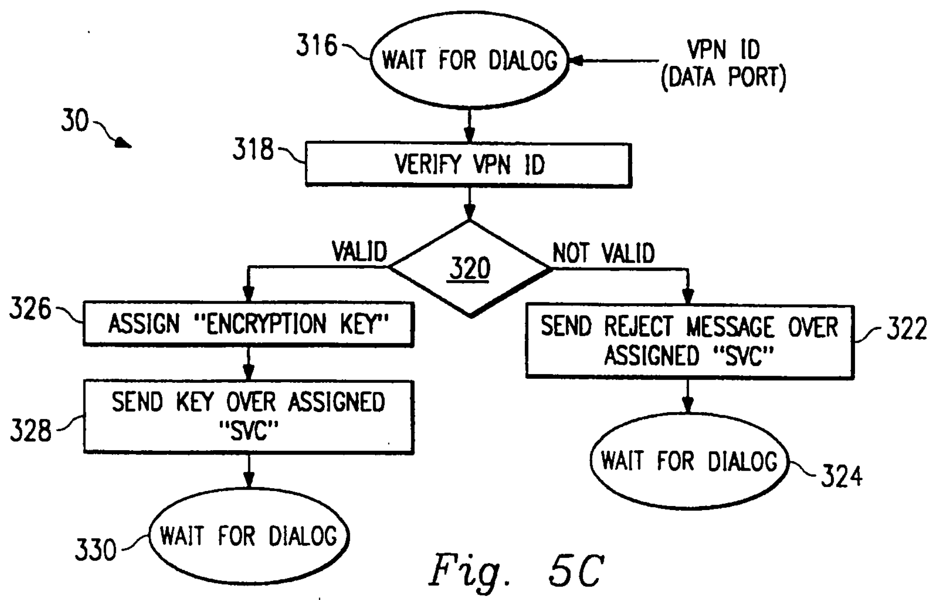

316
WAIT FOR DIALOG
VPN ID
(DATA PORT)
30
318
VERIFY VPN ID
VALID
320
NOT VALID
326
ASSIGN "ENCRYPTION KEY"
SEND REJECT MESSAGE OVER ASSIGNED "SVC"
322
SEND KEY OVER ASSIGNED "SVC"
328
WAIT FOR DIALOG
324
330
WAIT FOR DIALOG

Fig. 5C

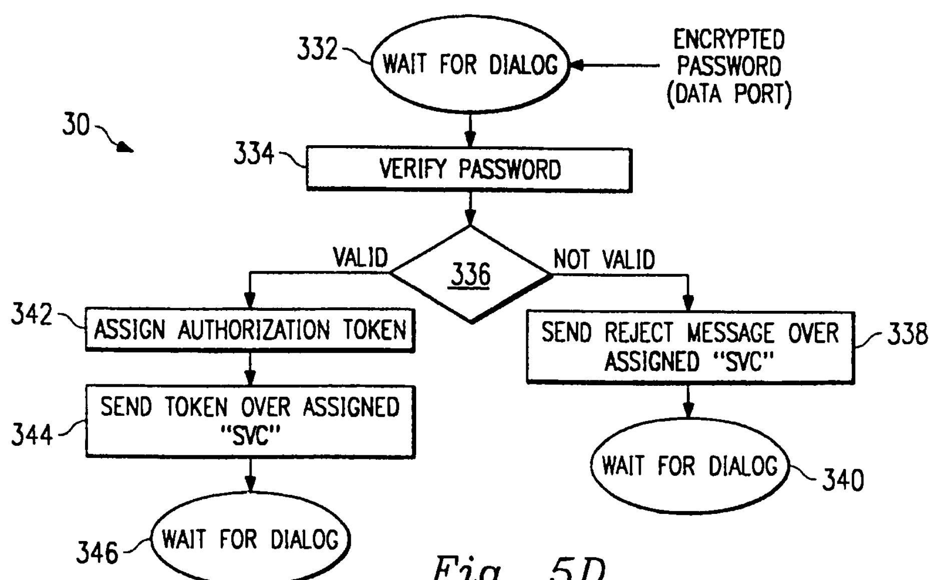
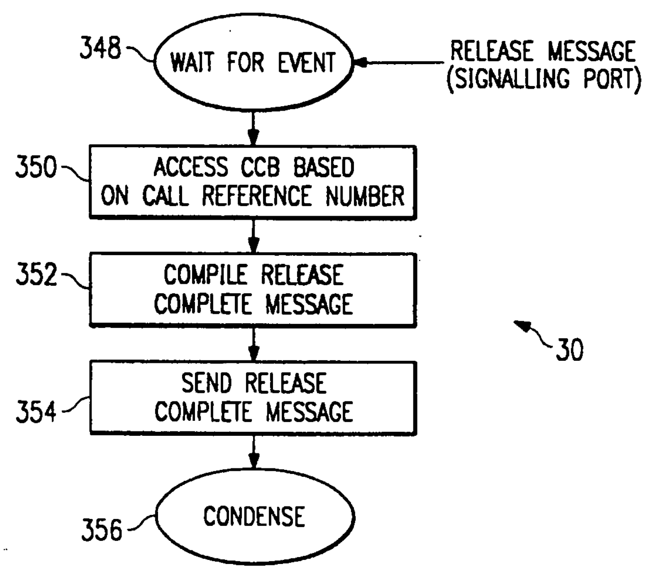

332
WAIT FOR DIALOG
ENCRYPTED
PASSWORD
(DATA PORT)
30
334
VERIFY PASSWORD
VALID
336
NOT VALID
342
ASSIGN AUTHORIZATION TOKEN
SEND REJECT MESSAGE OVER ASSIGNED "SVC"
338
SEND TOKEN OVER ASSIGNED "SVC"
344
WAIT FOR DIALOG
340
346
WAIT FOR DIALOG

Fig. 5D

Fig. 5E


348
WAIT FOR EVENT
RELEASE MESSAGE (SIGNALLING PORT)
350
ACCESS CCB BASED ON CALL REFERENCE NUMBER
352
COMPILE RELEASE COMPLETE MESSAGE
30
SEND RELEASE COMPLETE MESSAGE
354
CONDENSE
356

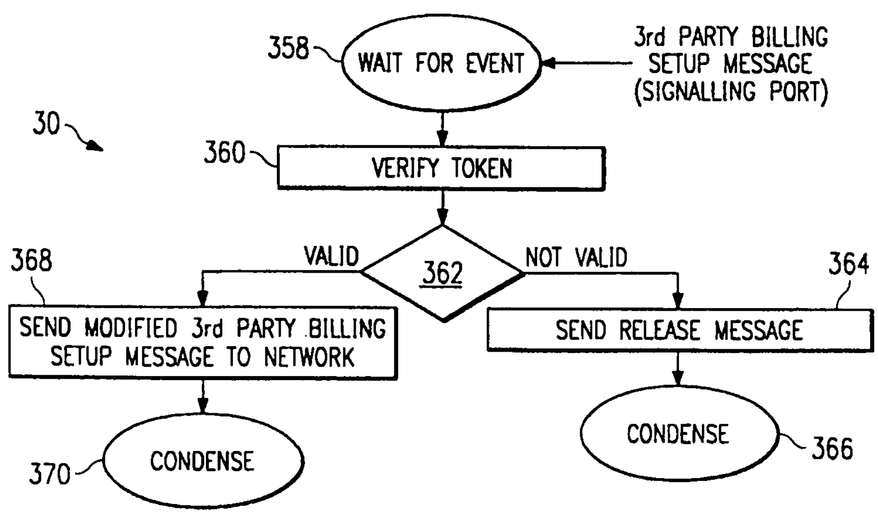

358
WAIT FOR EVENT
3rd PARTY BILLING SETUP MESSAGE (SIGNALLING PORT)
30
360
VERIFY TOKEN
368
VALID
362
NOT VALID
364
SEND MODIFIED 3rd PARTY BILLING SETUP MESSAGE TO NETWORK
SEND RELEASE MESSAGE
CONDENSE
366
370
CONDENSE

Fig. 5F

INTERNATIONAL SEARCH REPORT

Inte onal Application No
PCT/IB 97/01563

A. CLASSIFICATION OF SUBJECT MATTER
IPC 6 H04Q11/04 H04L12/22

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 6 H04Q H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	MUN CHOON CHAN ET AL: "AN ARCHITECTURE FOR BROADBAND VIRTUAL NETWORKS UNDER CUSTOMER CONTROL" NOMS '96 IEEE NETWORK OPERATIONS AND MANAGEMENT SYMPOSIUM , vol. 1, 15 April 1996, KYOTO, JP, pages 135-144, XP000641086 see abstract ---	1-20
A	BIC V: "VOICE PERIPHERALS IN THE INTELLIGENT NETWORK" TELECOMMUNICATIONS, vol. 28, no. 6, June 1994, page 29/30, 32, 34 XP000600293 see the whole document ---	1-20
	-/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
"&" document member of the same patent family

1

Date of the actual completion of the international search	Date of mailing of the international search report
19 March 1998	02/04/1998

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Staessen, B

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Inter onal Application No
PCT/IB 97/01563

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 729 256 A (NEDERLAND PTT) 28 August 1996 see abstract figures of pages 136 and 140 ---	1-20
A	CROCETTI P ET AL: "ATM VIRTUAL PRIVATE NETWORKS: ALTERNATIVES AND PERFORMANCES COMPARISONS" SUPERCOMM/ICC '94, 1 May 1994, NEW ORLEANS, US, pages 608-612, XP000438985 see abstract -----	1-20

1

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/IB 97/01563

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0729256 A	28-08-96	NL 9500339 A	01-10-96

Form PCT/ISA/210 (patent family annex) (July 1992)

B4

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 17/00 — A2

(11) International Publication Number: WO 00/17775

(43) International Publication Date: 30 March 2000 (30.03.00)

(21) International Application Number: PCT/US99/21934

(22) International Filing Date: 22 September 1999 (22.09.99)

(30) Priority Data:

60/101,431	22 September 1998 (22.09.98)	US
09/399,753	21 September 1999 (21.09.99)	US

(71) Applicant (*for all designated States except US*): SCIENCE APPLICATIONS INTERNATIONAL CORPORATION [US/US]; 10260 Campus Point Drive, San Diego, CA 92121 (US).

(72) Inventors; and

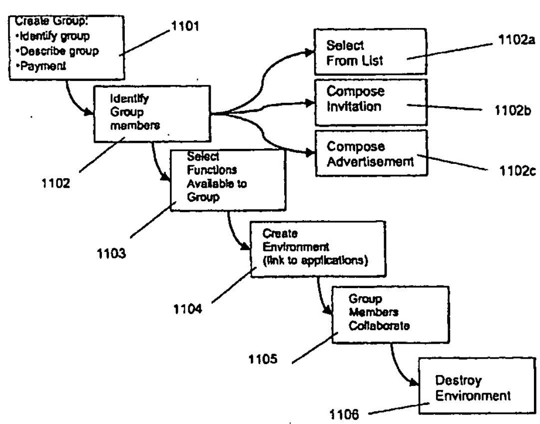
(75) Inventors/Applicants (*for US only*): MILLER, Craig [US/US]; Science Applications International Corporation, 10260 Campus Point Drive, San Diego, CA 92121 (US). MANGIS, Jeffrey, K. [US/US]; Science Applications International Corporation, 10260 Campus Point Drive, San Diego, CA 92121 (US). LESTER, Harold, D. [US/US]; Science Applications International Corporation, 10260 Campus Point Drive, San Diego, CA 92121 (US). NICHOLAS, John, M. [US/US]; Science Applications International Corporation, 10260 Campus Point Drive, San Diego, CA 92121 (US). WALLO, Andrew [US/US]; Science Applications International Corporation, 10260 Campus Point Drive, San Diego, CA 92121 (US). KRESS, Thomas, P. [US/US]; Science Applications International Corporation, 10260 Campus Point Drive, San Diego, CA 92121 (US). CHEAL, Linda, J. [US/US]; Science Applications International Corporation, 10260 Campus Point Drive, San Diego, CA 92121 (US). WEATHERBEE, James, E., Jr. [US/US]; Science Applications International Corporation, 10260 Campus Point Drive, San Diego, CA 92121 (US). DAVIES, Linda, M. [US/US]; Science Applications International Corporation, 10260 Campus Point Drive, San Diego, CA 92121 (US).

(74) Agents: WRIGHT, Bradley et al.; Banner & Witcoff, Ltd., Eleventh floor, 1001 G Street, N.W., Washington, DC 20001-4597 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: USER-DEFINED DYNAMIC COLLABORATIVE ENVIRONMENTS



(57) Abstract

A collaborative system and method allows members of a group to collaborate on a project such as a bid or proposal. According to a first embodiment, a complex instrument trading engine (CITE) facilitates negotiation between two or more parties. A set of tools and techniques are provided in order to facilitate negotiation and execution of complex instruments such as contracts between corporations and governments. According to a second embodiment, referred to as a dynamic collaborative environment, a user can define a group and a virtual private network environment including user-selected tools that facilitate communication, research, analysis, and electronic transactions within the group. The environment can be destroyed easily when it is no longer needed. Multiple environments can co-exist on the same physical network of computers.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

USER-DEFINED DYNAMIC COLLABORATIVE ENVIRONMENTS

This application is related in subject matter to and claims priority from provisional U.S. application serial number 60/101,431, filed on September 22, 1998. The contents of that application are bodily incorporated herein.

BACKGROUND OF THE INVENTION

1. Technical Field

This invention relates generally to computer systems and networks. More particularly, the invention relates to systems and methods for providing user-defined collaborative environments for transacting business or electronic commerce.

2. Related Information

Following hurricane Andrew, many insurance companies sought to limit their risk by withdrawing coverage from coastal areas. While this made good sense for the specific companies, it was not acceptable from a societal perspective. The cities, towns, homes and businesses built near the coasts could not afford to go without insurance, nor could the financial institutions that loaned money on these properties afford the risk. The problem facing the insurance companies was not the absolute magnitude of the risk, but the concentration of the risks in one area, leading to the possibility of very large losses resulting from a single event.

One law firm had conceived the idea of providing a mechanism for insurance companies to exchange risk. Companies with a high exposure in one area (e.g. Florida windstorms) could reduce their risk by ceding part of this to another company with non-coincident risk (e.g. California earthquakes) and assume part of the second company's risk in return. A company (CATEX) was formed to conduct such trading, but the trading rules had yet to be defined and the trading infrastructure had not yet been developed. CATEX postulated that the key barrier to insurance risk trading was determining the relative risk of different perils in different regions. One approach suggested by CATEX was to try to estimate these relative risks (termed relativities) for a broad set of perils and regions, to provide an initial basis for trading.

It was recognized, for various reasons, that this could not be done feasibly because: general estimates of risk, rather than the risk for specific locations, buildings, ships, etc. would be inadequate for commerce; there were many risks to evaluate given all of the permutations of location, perils, and structure; and companies would not be willing to trade risk based strictly on a third-party's analysis

An analysis of the problem, however, indicated that estimating the relativities was not essential to facilitate trading, or, in a broader sense, that trading was the only way to address the problem of insuring concentrated risk. The key difficulty was determining how to create greater efficiency in the reinsurance market, whether by introducing new instruments (like swaps), bringing new capital to the market, connecting more buyers to more traders, or reducing the cost of placing reinsurance. It was determined that the above concept could be implemented in an electronic trading system that could play an important role in promoting these factors, and could, in fact, transform the reinsurance market, which is not very automated. A system that allowed trading was developed and implemented. A more detailed description of this system, as enhanced in accordance with various inventive principles herein (referred to as "first-generation" complex instrument trading technology), are provided below. More generally, as electronic commerce (and business-to-business commerce, in particular) has grown, various companies have developed software tools and services to facilitate transactions on the Internet and over private networks. E-Bay, for example, hosts a well-known web site that operates a transaction model (a so-called "concurrent auction") that permits buyers to submit bids on items offered by individuals. Lotus Notes provides a network-oriented system that allows users within a company to collaborate on projects. Oracle Corporation hosts various transaction engines for clients that pay to host such services on a web site. DIGEX Corporation similarly hosts web-based application programs including various transaction engines. Other companies sell so-called "shrink wrap" software that allows individuals to set up web sites that provide catalog ordering facilities and the like.

Some Internet service providers, such as America Online, host "chat rooms" that permit members to hold private discussions with other members who enter various rooms associated with predetermined topics. A company known as blueonline.com hosts a web site that facilitates collaboration on construction projects. Various virtual private networks have been created to facilitate communication among computer users across the Internet and other networks, but these networks provided very limited functionality (e.g., e-mail services); are not user-defined (they must be created and installed by system administrators); and they cannot be easily destroyed when they are no longer needed.

The aforementioned products and services are generally not well suited to facilitating complex electronic transactions. As one example, most conventional services are predefined (not user-defined) and are centrally administered. Thus, for example, a group of companies desiring to collaborate on a project must fit their collaboration into one of the environment models provided by an existing service provider (or, alternatively, build a custom system at great expense).

Suppose, for example, that a group of high school students needs to collaborate on a research paper that requires soliciting volunteers for a survey on drug use, conducting the survey, brainstorming on the survey results, posing follow-up questions to survey participants anonymously, publishing a report summarizing the results, and advertising the report for sale to newspapers and radio stations. This project requires elements of communication among persons inside a defined group (those writing the paper) and outside the group (e.g., survey participants); conducting research (conducting the survey, compiling the results, comparing the results with other surveys published by news sources; and brainstorming on the meaning of the results); and conducting a commercial transaction (e.g., publishing the survey in electronic form and making it available at a price to those who might be interested in the results). No existing software product or service is available to meet the specific needs of this research team. Creating a user-defined environment including tools and communication facilities to perform such a task would be prohibitively expensive. Even if such a tailor-made environment could be created, it would be difficult to disassemble the environment (computers, networks, and software) after the project was completed.

In short, there is a need to provide a user-defined collaborative environment that is tailored to the needs of particular groups that conduct communication, research, electronic transactions, and deal-making.

SUMMARY OF THE INVENTION

A first embodiment of the invention, referred to as a complex instrument trading engine (CITE), facilitates negotiation between two or more parties. In this embodiment, a set of negotiation tools and techniques such as anonymous email, secure communication, document retention, and bid and proposal listing services are provided in order to facilitate the negotiation and execution of complex instruments such as contracts between corporations, governments, and individuals.

A second embodiment of the invention, referred to as a dynamic collaborative environment (DCE), allows members of a group to define a dynamic virtual private network (DVPN) environment including user-selected tools that facilitate communication, research, analysis, and electronic transactions both within the group and outside the group. The environment can be destroyed easily when it is no longer needed. Multiple environments can co-exist on the same physical network of computers.

Although the two embodiments are described separately for ease of comprehension, it should be understood that the two embodiments share many features and, in fact, the second embodiment could include some or all of the features of the first embodiment in a generalized collaborative system. Consequently, references to a specific embodiment in the following description should not be deemed to limit the scope of features or tools included in each embodiment. Moreover, references to specific applications, such as the reinsurance industry, should not be deemed to limit the application of the invention to any particular field.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1A shows a four-step model of deal making including meeting, analysis, negotiation, and closing the deal.

FIG. 1B shows contract formation among a group of parties to a contract.

FIG. 2 shows a listing display system showing all offers for contracts and responses thereto.

FIG. 3 shows details of a listing that has been selected by a user.

FIG. 4 shows one possible implementation of a reply card definition screen.

FIG. 5 shows one possible implementation of a document management screen.

FIG. 6 shows one possible implementation of a screen indicating persons having access to a shared folder.

FIG. 7 shows a list of consummated deals in the system.

FIG. 8A shows detailed information regarding a completed trade.

FIG. 8B shows a deal summary including structured and unstructured information concerning the deal.

FIG. 9 shows a "flip widget" in a first state.

FIG. 10 shows a "flip widget" in a second state.

FIG. 9A shows a more detailed example of a "flip widget" in a first state.

FIG. 10A shows a more detailed example of a "flip widget" in a second state.

FIG. 11 shows method steps that can be carried out to define, create, and destroy an environment according to a second embodiment of the invention.

FIG. 12 shows one possible system architecture in which various principles of the invention can be implemented.

FIGS. 13A through 13C show one possible user interface for creating a group and identifying group members.

FIG. 14A shows one possible user interface for selecting group members from one or more lists.

FIG. 14B shows one possible user interface for selecting group members by composing invitations.

FIG. 14C shows one possible user interface for selecting group members by composing an advertisement.

FIG. 15 shows a banner advertisement 1501 displayed on a web site, wherein the banner advertisement solicits participation in a group.

FIG. 16 shows one possible user interface for selecting communication tools to be made available to group members.

FIG. 17 shows one possible user interface for selecting research tools to be made available to group members.

FIG. 18 shows one possible user interface for selecting transaction engines to be made available to group members.

FIG. 19 shows one possible user interface for selecting participation engines to be made available to group members.

FIG. 20A shows an authentication screen for group members to gain access to a newly created environment.

FIG. 20B shows a web page generated for a specific user-defined environment, including tools available to group members having access to the environment.

FIG. 21 shows one possible method of generating environments in accordance with various aspects of the present invention.

FIG. 22 shows one possible data storage arrangement for storing and manipulating brain writing cards.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A. COMPLEX INSTRUMENT TRADING ENGINE EMBODIMENT

A first embodiment of the present invention provides a second-generation version of a complex instrument trading system. The second-generation system includes specialized tools that were not included in the first version of the prior art CATEX insurance trading system described above. These tools represent a substantial improvement over the first generation and incorporate new concepts of communications in a trading environment, and other capabilities that did not exist in the first generation technology. In addition, it is believed that many of these tools are also applicable to software systems other than the Complex Instrument Trading Engine or Negotiating System (CITE) described herein. Thus, the inventive principles are not limited to trading systems for complex instruments, nor even to trading systems in general.

Primarily, the tools described herein ameliorate certain difficulties associated with trading of complex instruments. Complex instruments are instruments where there is more than one dimension for negotiation. As compared to such instruments as securities, complex instrument transactions take longer to research and consummate and require more extensive documentation. For example, stock trading employs a simple instrument (a share) and negotiation focuses on one dimension (price) while insurance contracts have many dimensions (term, price, coverage, definitions of perils, etc.). The stock market is relatively simple to automate -- as soon as bid and asked prices match, the deal is concluded in an instant according to the rules of the exchange. Automation of complex trading is much more difficult, since the parties must negotiate and reach agreement on multiple dimensions and document that agreement using an instrument specific to the precise agreement. Automation of complex instrument trading is more difficult in every way than trading simple instruments.

The trading model behind the Complex Instrument Trading Engine or Negotiating System is built around a simple, four-step model of deal making. Referring to FIG. 1A, the steps are as follows:

1. Meeting: Potential buyers connect with potential sellers with reciprocal interests. This connection does not mean that a deal will necessarily be concluded but simply that the two parties have some basis for continuing discussion. In simple

instrument trading, it is typically only necessary to advertise quantity and price offered or sought. Offers for complex instruments must include substantially more detail and (frequently) extensive attachments or exhibits.

2. Research/Analysis: Each company considers its own position and/or offer and the counter party's position. Using information and analytic tools from various sources, including internal resources and resources provided by or through the trading system, each party does research and refines its position. The multiple dimensions of complex instruments increases the analytical complexity and limits the value of a simple market price. As indicated by the arrows in FIG. 1, this step is usually performed iteratively with the negotiation.

3. Negotiation: Parties to the negotiation speak directly and exchange whatever information is necessary to advance the deal. As indicated by the arrows in FIG. 1A, this step is usually performed iteratively with the research step.

4. Close: the companies negotiate and sign an instrument that documents the deal. This can be a complete and detailed contract, or it may be a simple memorandum. In simple instrument trading, the actual trade agreement is often standardized by the exchange. In complex instrument trading, the agreement must be more specific to the deal, though it is possible to use such tools and fill-in-the blank forms.

Within a system using these complex instrument tools, trading parties can place offers to buy, sell, or trade in a public area, and examine such offers ("listings") posted by others. Using advanced communications tools the parties can conduct initial discussions to determine if a placement is possible. Using tools described herein, the initial contact can be done anonymously.

If a deal seems possible, the system preferably provides access to the extensive information necessary to assess the possible deal. This can include static information (e.g. reports or data) maintained within the system, links to information providers outside the system, online analytical tools, and links to providers of analytical services.

For complex instruments, the process of negotiating a deal is contemplated to be an iterative one, with successive stages of analysis and discussion. The need for extensive communication is one of the critical distinctions between trading of simple instruments (e.g. retail sale) and complex instruments. Complex instrument trading requires dialog and more -- exchange of documents (often voluminous), consultation with counsel and intermediaries, conferencing, and working together on the final agreement. For electronic commerce to have an impact in complex instrument trading, it must support and facilitate this communication, and not force traders to fall back on methods and technology outside the electronic trading environment.

The final step is closing the deal. The companies can negotiate a contract online. Tools provide sample, fill-in the blank contracts and memoranda of understanding as a starting point. Negotiators can begin with these, or they can use one of their own. Collaborative software makes it possible to display text simultaneously on each negotiator's screen and to work on the language together. When the contract is final, the system allows for secure, online signature, though companies not comfortable with electronic signature for very large deals may print a hard copy and sign it conventionally.

By creating electronic exchanges for complex instrument trading, the CITE tools can have a fundamental and positive impact on many areas of commerce:

1. An electronic exchange makes it possible to put an offer in front of more people more quickly than could be informed through direct contact, even allowing for active intermediaries or brokers.

2. Traders can advertise and conclude deals without the need for an intermediary when they have adequate support or internal resources.

3. Through better communications, wider exposure for offers, and the first steps towards standard contract language, electronic trading of complex instruments can substantially reduces transaction costs.

4. With lower transaction costs, it is possible to conclude deals that were not possible with higher overhead.

5. Through the immediate posting of the results of trades, pricing is moved towards a market basis, reducing research and analysis costs enormously. This speeds placement.

6. Smaller exposure means lower risk, and market pricing is an adequate surrogate for analytically derived pricing in some circumstances. Together these factors make it possible for traders to participate in markets or market segments in which they would not normally do business.

7. By making it possible for all companies, large and small, to talk directly to each other, electronic trading of complex instruments can lead to the democratization of the marketplace increasing competition.

Overall, electronic trading of complex instruments has the potential to improve the efficiency of markets enormously, and to establish markets in areas of commerce that are currently done through intermediaries or on a one-on-one basis. The trading tools described herein are designed to facilitate electronic trading of complex instruments. The first-generation complex instrument trading tools broke new ground in the extension of electronic commerce into new and more complicated markets. The table below summarizes the areas of new and improved technology, organized into the four steps of the general complex instrument trading model.

Phase	First Generation Complex Instrument Trading Technology (PRIOR ART)	Advanced Complex Instrument Trading Technology
Meet	• Operates on private network only • Post a listing to board by filling out a form • Display listing summary in a table • Search listings by key word • Post response to listing on board • Establish communications with lister by following up on contact information in listings using unconnected communications tools	• Operates on private network or over the Internet • Post listing to a board by filling out a form • Listings and responses can have attachments and documents • Display listing summary in a table, with sorting by title, date, market type, buy/sell, or listing number. • Search listings by keyword • Register keywords with an electronic "agent" that monitors listings and sends notice of relevant new listings by Email • Post response to listing on board • Send private response (anonymously or with name attached). • Response can be through a "reply card" designed by the trader posting a listing, to structure responses • Direct connection between listings and communications tool

Analysis	• Internet access to research resources, on line and third-party analysis	• Internet access to research resources, on line and third-party analysis • Research resources searchable using the same search engine and display as used for listings. • Online dialogs / user groups
Negotiation	• Requires private network • Directory of contact information for all traders • Connection between directory and Email client. • Directory not linked to other components of the system • Anonymous mail application providing for communications between two individuals • Anonymous mail delivered to mail client • No attachments for anonymous mail • No system for central repository of documents	• Works on Internet or private network • Directory of contact information for all traders. • Direct connection between directory and Email client • Direct connection between directory and online conferencing software • Directory linked to listings and document management tool • Anonymous mail application providing for communications between individuals or groups of people working together • Anonymous mail does not require separate Email client software • Anonymous mail supports attachments • Internet-based system for distributions and sharing of documents. • Password and secure has protection for documents.
Closure	• Requires private network • Online signature of uploaded document	• Internet or private network • Online signature of uploaded document • Registration / closure of deal through a fill-in form • Provision for digital signature and archiving of all documents associated with a deal

Referring to FIG. 1B, one aspect of the system within the framework of the negotiation/analysis loop shown in FIG. 1, is the ability to define one or more contracts, for example, in the parlance of the reinsurance trade, "slip sheets." Various members of a group of authorities modify the contract causing it gradually to take a final form that is either rejected as untenable or accepted as a finalized deal. The system exposes various aspects of the contract and attendant documents to the appropriate participants in the transaction, also providing each with a level of authority to add, delete, or modify documents as well as the evolving contract or contracts (assuming there may be various contract templates being discussed). These filters (filter 1 through filter 4, for example), as shown in FIG. 1B, determine the authority of the party (Party 1-Party 4) to modify or see the data object, whether it is a document or a slip sheet. The system combines this system of filters with signature technology for closing the deal; that is, implementing signatures so that an enforceable contract is generated.

A deal is like any other data object and once it is defined and entered, it cannot be modified. Elements of the deal can be "signed" such as documents attached to a contract (for example, Contract 1 has documents D1 and D2 attached to (combined with) it. Together these elements, the contract and the attachments, define the deal. Also, the entire deal 245 can be signed using a signature device ("widget") S8. Other documents may relate to a deal but not be attached. These can be viewed using a document manager described further below.

Listing System

Referring to FIG. 2, a listing screen displays all offers for contracts, for example offer 314, as well as responses to them, for example, response 313. The parameters of the offers and responses to them are shown in columns, the heading of each of which may be selected to sort the listings by that heading, for example heading 315 if clicked would sort by the unique index number for the listing. Notice that the responses (for example, response 313) are shown indented to indicate a series of elements of a dialogue-thread. As indicated, the responses have a "daughter" relationship to the parent listings. That is, listing 314 is a parent and reply 313 is a daughter. The daughters remain in their hierarchical position beneath the parent despite sorting by the column headings. This makes the tabular sort scheme compatible with a threaded display, which is useful to show dialogues.

Referring now also to FIG. 3, when a user invokes a display of the details of a listing by clicking on an index hyperlink 312 to show the details of the listing, a user interface element displays the lister's defined parameters of the listing. As shown, various parameters are displayed, many of which are hyperlinked. For example, attachments 304 may be selected to display the corresponding attachments. A detailed description 301 may be provided as well as specific instructions for responding 302. A reply button 303 permits the user to reply. Activating the reply button 303 will either invoke a standard public reply screen which creates a new listing similar to the parent listing or a special reply defined by a reply card which is further described below.

A reply to a listing can take the form of a public reply that invokes a screen substantially the same as FIG. 3 but with blank spots for entry of reply information. A more useful kind of response element is a reply card that can be defined by the lister. This is because in negotiations on complex transactions such as reinsurance contracts and, for example, pollution emission allowances, the parties with whom a lister would be willing to trade are limited in terms of certain criteria. These criteria will vary from one type of transaction to another.

In an active trading system, the number of listings can quickly grow to a large number and quickly exceed the number which can conveniently be displayed in a single table. Several capabilities are built into the system to address this problem. First, by default, listings are presented in order from newest to oldest. Second, the sort capabilities previously described allow users to modify the standard order. Third, the total market may be divided into subcategories. In the area of insurance catastrophe risk, these could include categories for different lines of insurance (e.g. marine, aviation, commercial buildings). Fourth, users may enter search criteria to identify a subset of listings of particular interest.

Searching listings: A user may enter a keyword such as "hurricane" to identify all listings that contain that word in the title, description, and (optionally) attachments. To improve the reliability of the search, users are provided access to a standard lexicon when composing a listing. In the first embodiment, this capability is invoked by pressing the right mouse button while the cursor is any field of the listing. A list of common terms is displayed. The user can select the term of interest, which is then placed into the text of the listing at the insertion point marked by the cursor. For example, a listing for insurance risk would typically include a field for geographic scope (i.e. the location of the properties to be insured). When in this field, the lexicon displayed would include terms such as "California" and "Coastal Florida". Choosing a term from the lexicon insures uniformity of terminology across listings and between the search engine and the listings. "California" will be used rather than a mix of "Ca", "CA", "Calif", etc. The search is further improved by symantic indexing. Essentially, this means that synonymous terms are grouped, so that searches for one will find the other. A person who searches for "California" will get listings for "Los Angeles" that do not include the word "California".

The search engine can include an agent capability. This agent capability offers the user the option of saving a search, after the user reviews the results and deems them acceptable. This search is retained in a library of searches along with the email address of the owner of the agent. The search is retained in the library until is it either deleted by the user when it is no longer needed or automatically deleted in a cleanup of searches older than a certain date. Whenever a new listing is placed on the system, all of the saved searches are executed. If the new listing meets any of the search criteria, a message is sent to the owner of that criterion via email or instant messaging.

A model was developed to allow a lister to define a set of criteria and request a set of information from any respondents in the form of an anonymous reply "card." The card defines a set of requested information which may be packaged as a document object and placed in the document manager system and connected with each listing. A user would download the reply card and fill the card out and send it back to the posting party.

A document object, called a reply card, is made available to a respondent through the document manager. The respondent is permitted to retain his anonymity as is the lister. Each may communicate with the other through an Amail system described in more detail below. The respondent supplies the requested information and sends the data to the lister. A system in the listing manager allows a lister to define a reply card having any particular fields and instructions required of a respondent. Some of the information required may be obtained automatically from a set of default data stored on the respondent's computer.

Referring to FIG. 4, a reply card definition screen is invoked to define the parameters of a new listing. The new listing is defined using a user-interface element looking much like FIG. 3. While the details are not critical, the definition of reply card involves, in essence, the definition of a user-interface control such as a dialog with radio buttons, text boxes, etc. These are definable for server-side implementation through HTML and are well known so the details are not discussed here. The lister defines a set of controls that allow the entry by a replying party of the information that the lister requires. The reply card is stored as any other information object and may be organized and accessed through the document manager described below. FIG. 4 shows a simple example of a format of a reply card.

A reply card is created by a user when posting a new listing. The lister specifies the information that must be included in a response, and the type of information object to display for the data element (e.g. a text box, check box, radio button). The system then creates an HTML page to collect the requested information. When a respondent clicks "Reply Card" on the listing screen, the page is displayed. All of the responses are automatically entered into a database created automatically when the reply card is composed. As each respondent fills out a reply card, a new record is added to the database of the system and the lister is permitted to view it through an appropriate filter as discussed above.

<u>Signature System</u>

As business is increasingly done in an electronic environment, electronic signature and approval is becoming more critical. The typical electronic signature model has focused on two aspects:

1. Electronic validation of the user -- specifically determining that the person viewing a document on line is the authorized signatory; and

2. Validating the document being signed by a means that either prevents modification of a document or will reveal whether changes have been made.

Methods for validation of identity range from simple personal identification numbers or passwords, to electronic signature pads, and more advanced methods of biogenic validation such as fingerprint or retinal patterns. Methods for document validation range from simple archiving of one or more copies in a read-only model or inaccessible location to methods based on mathematical algorithms that create a characteristic number or alphanumeric string for a document. These strings are termed "electronic signatures." Changes to the document change the electronic signatures. Because the signatures are much shorter than the documents, very many documents have precisely the same signature, but the algorithms to calculate the signature are very difficult to invert, so that it is effectively impossible to deduce a meaningful change to a document that will preserve a specific signature.

These two aspects of electronic signature are highly developed, but there has been little analysis or development of the general process by which documents can be signed.

The invention allows for secure and reliable routing of documents, for which signatures are required, to a specified list of signatories. Unlike prior art systems, such as ordering or accounts payable systems which have highly structured signature procedures tailored to a specific process, the present invention provides a flexible method and system that allows a signature-type of authority/requirement to be attached any kind of information object. The method is sufficiently abstract, flexible, and general that it can be applied in many contexts aside from the CITE embodiment described in the present specification.

One signature method/device employs the following steps:

1. Registration of signatories – This process provides a register of identifiers indicating entities with signatory authority and correlates these identifiers with the information objects for which the signatory authority is applicable. The same register may also be used to identify other types of authority in the system in which the signature device is implemented. For example, document read authority, modification authority, exclusive access to documents, etc. may also be provided in the same register. Signature registration may be provided automatically in certain systems where registration of, for example, read/write authority is provided since any entity with signatory authority would in almost all instances, also be provided with some other kind of authority, most notably, read authority. Thus, where the signatory system is embedded in certain kinds of systems, it may be that no particular additional method or device is required to implement signatory registration since an existing register may already exist or be required for other purposes.

Registration information includes the general categories of information listed below. Definitions of specific fields within these categories are a function of the specific implementation of the signature system or the parent system. The following are exemplary:

1. Identity – unique identifier of the entity, the organization(s) with which the entity is affiliated, other relevant information.

2. Contact information – information indicating how the entity can be reached, how documents and mail messages can be routed to the entity.

3. Security Information – a password for each class of signature as described further below.

2. Classes of signatures – The device/method provides a variety of classes of signature, each associated with a unique level of approval or level of commitment. For example, a class of signature-authority can be defined that represents individuals, for example, with authority to sign contracts only below a set amount, or for expenses relating only to one department of an organization, or within certain time constraints, etc. The signatory system maintains this taxonomy of possible signature types in a database with a unique identifier for each level of authority defined. The system allows the creation and deletion of classes. Each class is preferably permitted to be named and a descriptive definition attached to each class.

3. Defining a Set of Signatures – Using an appropriate user interface element, the user of the system selects an information object (for example, a document, file, or collection of such objects) requiring signature(s). The entity originating the signature process then identifies the entity or entities required to sign the object. The specification of the signers can proceed either by the selection of individuals from a list supported by the above defined entity register. Alternatively, in an environment where individuals are strongly bound to organizations, for example, it can proceed by selecting the list of organizations that will sign and, within each organization, the person who will sign. The list is built by a series of selections. After each selection from the list, the user indicates his/her desire to add the selected individual to a list of required signatories. The user interfaces provides for entries in which all the selected signatories are required or only one of the selected signatories are required.

For example, if more than one entity is selected from the list prior to the selection (e.g., clicking an "Add" button), the system may require a signature from any of the people selected, but not all of them. To require signature from every member of the group, the initiator may select one person, then "add", select the second, then "add", and so on. Thus, adding a group with one "add" command would provide an "any signature will suffice" list and adding members individually would require a signature from that individual or entity. Note that this technique may also be used to define combinations of required and "any of" groups.

For each signer or group of signers selected in a single "add" command, the initiator of the signing sequence must specify the class of signature associated with the person for the document being signed. This may be selected from a list of signature classes (see item 2). If the specific implementation of the signature process only supports one class of signature, the selection of class may be omitted.

4. Random or Serial Order of Signature – After or concurrent with the creation of a signature list, the initiator specifies whether signatures must be in order or if a specific order is not required. For purposes of defining the order of signature, individuals who are selected as a group are considered as occupying a single place in the sequence.

5. Document Authentication – Upon initiating a signature sequence, the information object is authenticated by means of a secure hash algorithm. The specific hashing algorithm is a matter of design choice or may made dependent on a user's choice. There are several possible hash algorithms available in the public domain. The electronic signature produced by the secure hash algorithm is archived with the information object in a secure repository. If the information object is, for example, a record in a database, the contents of the record are copied to a file in delimited format for archival purposes. If the object is a table, the table is exported prior to archive.

6. Document Routing – Upon initiation of a signature sequence, the initiator specifies how the signatories are to be informed. The options are:

- No notification from the signature system
- Email message
- Email message with attachment of the information object.
- Posting on a signature web site

The system accepts and implements the chosen method, which may be connected to the signature or a single choice applied to all signatories. Alternatively, the method of notification may be stored with the signature class definitions. In a signature process with no required order, e-mail notice may be sent simultaneously to all of the designated individuals at the time of initiation. If the process is serial, only the first person may be notified. The electronic signature of the information object may be included in an e-mail message.

7. Accessing the signature system – The signature system can be implemented for access via a web browser or database client-server software across the Internet, an intranet, a LAN, or a WAN. Access to the system will typically require a password, but this may not be necessary on a secure network. Upon access to the system a user will have the option to display a list of all of the information objects which he or she has signed or is being asked to sign. For each object, the display can include the following information:

- Object name
- Description of object (text, mime, size, date)
- List of scheduled signatories
- Date each person signed
- Class of signature for each person
- Electronic signature produced by the secure hash algorithm

If the object is available (viewable) on line, the display may also include a link to display or download the object.

8. Validation of the Object at Time of Signature – If the user downloads or views the object, the system will execute the secure hash algorithm to calculate the electronic signature. This will be displayed so that the potential signer can compare it to the signature calculated at the time the process was initiated. If the user has previously downloaded the object or received it as an attachment to an Email, the user may access the secure hash code through the signature system and apply it to the version on the user's disk.

9. Signing a Document – After the user has determined that an information object is authentic and that the contents merit signature, he or she can affix a signature by authenticating his or her identity. Various means of authentication may be used. The means of authentication may be at the discretion of the manager of the signature system. Such means may include personal identification numbers, passwords, authentication based on computer address or information stored on the signer's computer, third party validation using a public key or other security infrastructure,

or biogenic (fingerprint-recognition, retina scan) methods.

After a document is signed, the date of signature is recorded in a database so that the display to other potential signers is updated. If the signature process is serial, the next person in the sequence is notified. E-mail notice can be sent to all signers when the last signature is collected.

10. Follow-up – At the time a signature process is initiated, the initiator can select a time (in hours, days, or a time or date-certain) for automated follow-up. If a document is not signed within the specified period after notice, a follow-up e-mail can be sent as a reminder. Additional reminders may be sent at the same interval if the object has not been signed. The reminders can be sent automatically by the system according to user-input specifications.

11. Cancellation – The initiator of a signature sequence can modify the sequence at any time, except that a signer can not be deleted from the list once they have signed an object.

12. Transfer of authority – The individual initiating a sequence can transfer the right to modify the list signature list to another individual in the system with appropriate validation of identity.

Document Manager

Successfully conducting commerce over an electronic network requires the exchange not only of messages, but of substantial blocks of information in the form of documents and data. Beyond simply transferring files from hand to hand, it is often necessary for multiple parties to work on a document simultaneously or serially, to track changes, and to maintain a record of versions. Two general architectures have emerged for document management, which can be termed a "mail model" and a "repository model." Under the mail model, documents are attached to messages and circulated person to person. Under the repository model, documents are placed in a central location. There are advantages and disadvantages to each. At a summary level:

	Mail Model	**Repository Model**

Advantages	Precise routing on a document specific basis. Push in the recipient is informed of a new document. Coupling between document flow and a messaging. Dating is automatic.	Compact storage -- only one version of a file need to be stored. Natural group of files on the basis of subject or access group. Supports good configuration management and version control.
Disadvantages	Creates multiple versions of a document, confounding configuration management and version control. Does not easily couple to online collaboration. Many mail servers limit size of attachment. Relatively high effort to prepare messages.	Not push in the sense that users are automatically informed of new documents. Security model is more complicated than for email. Prior arrangement is necessary to access a repository.

A browser-based document management model and tool combines the best features of repository model and the mail model, for document dissemination and sharing across the Internet or an intranet.

<u>General Architecture</u> – The general architecture of the system combines two basic components: (1) a database of directories and documents and (2) a directory of users. The directory of documents lists documents (of any type) contained in the system, and folders that can contain documents or other folders. The directory of users contains a list of individuals and organizations that can access the system, with passwords and/or other information necessary to validate identity and to establish authority.

<u>Representation of document</u> – The term "document" is used here in the broadest sense of any file that can be stored magnetically or electronically. Preferably, each file is given a unique name consisting of a string of no more than 256 characters. Preferably, the character set is limited to those members of the ASCII character set

which are displayable or printable. Thus, such codes as "escape" which have no visible representation, would be excluded. This is the file name that is displayed for purposes of identifying the document to the users. There is also an actual file name (which is not shown to users) to identify where copies of the file are stored in the central repository. Certain other information is kept in addition to the name of the file. This includes the following:

1. Data of creation
2. Date entered into repository
3. Person who entered the document into the repository
4. Description
5. Size of the document
6. Document type if known
7. Date of last update
8. Access password (optional) stored in encrypted form
9. File folder(s) where the document appears
10. Actual file name

In addition to the above information, data indicating whether the file is checked-out and to what entity, and the identities of entities that have checked the document out and returned it in the past are also stored. The term "checking out" is described further below. These functions related to file change control and configuration management, which are discussed later.

User database – A database contains information on all individuals who can currently access the system or who previously had access up to an administratively determined retention period. This database includes standard contact information including physical and electronic addresses. Security data such as passwords and/or encryption keys is also maintained. In a combined system such as the presently described system, the same database or registry of users can be employed for the document manager as for the signature system.

High level directories – The entire document management system can be divided into a number of high level directories that the user can display, one at a time. These include, at a minimum, a "Private" directory of files and folders visible only to the user, and a "Public" directory of files and folders visible to all users. Additional high-level directories can be created by the system administrator as needed. These could correspond to projects, business units, or any other logical basis. At any point in the use of the document management system, a user can see and select from the high level directories to which the user has access. The name of the currently open directory can be always displayed on the screen.

<u>Displaying the contents of a high-level directory</u> – When a user selects a high-level directory, the repository displays a series of file folders against the left margin of the active window. File folders whose contents are displayed are shown as open folders. File folders who contents are not displayed are shown as closed folders. A folder is opened or closed by clicking a single time. When a folder is opened, the contents are shown with an indent to indicate the parent/child relationship between the folder and its contents. Each folder can contain files, shown by an icon representing a printed page and other folders, represented by an image of a closed folder.

<u>Information about a folder</u> – Information about each folder is displayed on the same line, to the right of the folder icon. This information is as follows, from left to right:

1. Name of the folder
2. Number of files in the folder, or the word "empty"
3. Accessibility of the folder

Accessibility refers to user access rights to a folder which may private relative to the entity that created it, restricted (limited to a subset of people who can access the high level directory), or shared (available to everyone with access to the high-level directory). The level of access to a directory is indicated by the words "private", "restricted" or "shared."

If the directory is restricted, clicking on the word restricted displays a list of the entities that have access to the folder. This list is a series of hyperlinks. Clicking on the name of a person pulls up detailed contact information (discussed below). The objective is to facilitate communications between people with a shared interest in a file.

<u>Information about a file</u> – Information about a file is displayed to the right of the file icon. From left to right, the first item displayed is the name. This is followed by the word "details." Clicking on "details," causes the document management system to display complete information about the file (see Item 2, above), the person who placed the document in the file, (see Item 3, above), and the person who most recently modified the file.

Information about people/entities, and the link to communications – Information about people/entities with access to the system is displayable at several points in the document manager system:

1. by accessing the directory of users
2. when creating a new folder with "restricted" access
3. when displaying detailed information about a file (see #7)
4. when displaying information about a restricted directory (see #6)

Whenever such information is displayed, contact information from the database is rendered along with the name. Depending on the implementation, this can include complete contact info (multiple addresses, telephone and fax numbers, and email addresses), or some of the contact information may be restricted, in which case it is not displayed.

Creating a new top level folder – A new folder is created within a high-level directory, for example by clicking a button labeled "new folder." This can bring up a dialog in which the user assigns a name to the new folder and selects the type of access (private, shared, or restricted) rights to be assigned. If the document is restricted, the user specifies the entities (organizations and/or people) that can access the folder. If the creator of the folder specifies that an organization has access to a folder, all individuals associated with that organization may be granted access. Folders to which a user does not have access may remain hidden or not displayed. Alternatively, these folders can be shown with some indication that they are not accessible, for example, by ghosting.

Functions related to a folder – Once a folder is defined, a user can execute the following options.

1. Create a subfolder, using the same process described in 9
2. Add a document to the folder, using the process described in 11
3. Delete the folder, if it is empty
4. Modify access to the folder using the same tools used to specify access initially

The functions can be invoked by, for example, clicking on the appropriate label to the right of the name of the folder icon.

Adding a file – Users add a document using a dialog box that prompts for the following information:

1. Location of file - may be entered by user, or selected through a standard file browse dialog

2. Name to be used for the file in the repository

3. Version number or name (optional)

4. Password or encryption key (optional)

5. Description (optional)

6. Access rules (read only or read-write)

After entering the above information, the user either aborts or initiates upload. The information listed above is recorded along with the name of the person entering the document, and date and time.

<u>File options</u> – The following functions may be provided, preferably for every file in the system:

1. Delete (with confirmation)

2. Archive. The file is removed from main repository, but a copy is retained outside the repository. It may be restored though manual intervention.

3. View or download: a copy of the file is brought to the user's computer. This file can be modified there for the individual user's use. A modified version can be uploaded as a new file or different version of a current one, but a file in the repository can only be replaced if the user has it checked out.

4. Check out / check in (see below)

5. Forward (see below)

6. Change Password. The old password must be entered followed by a new password and confirmation.

7. Move: copy or more a document from one folder to another.

The functions may be invoked, for example by clicking on a label corresponding to the function, which can be displayed to the right of the name of the file. Not all options are shown to all users. If an entity does not have write-access to a file, the entity may not delete it, archive it, check it in or out, or change the password.

<u>Check in / Check Out</u> – All entities with write access to a file may check it out. By checking the file out, the entity reserves the exclusive write to save changes to a file. A person may not replace a file that is checked out. To check out a file, the user selects this option from the list of functions associated with the file. The user can

then enter an expected return date and a reason that the file is checked out or the changes to be made. This information is available to all others who can view the file. Each check in or check out is recorded in a permanent log. After a file is checked out, the "check out" button or link is changed to read "check in."

Each individual can check in only the files that he or she has checked out. This is done by clicking "check in." The user may then upload a new version of the file by specifying the location of the file on disk, or indicate that the version of the file currently in the repository is to be retained. After a file is checked in, the check button is changed back to "check out" and the file can be checked out by another user.

Forwarding – A file can be forwarded to any other user of the system. When the forward function is invoked, a list of users is displayed. The sender selects one or more users. Upon confirmation, a copy of the document is placed in folder labeled "in box" in each recipients private directory.

Referring to FIG. 5, a main screen for the document manager creates (using server-side scripting) a user-interface display with some of the features of a Windows Explorer® -type display. File and folder icons are shown along with an array features arranged next to each. The similarities with Windows Explorer® fairly well end there, however. Each of the properties shown next to each file/folder entry invokes a feature.

A parameter object W "Details" invokes a detailed display of the corresponding document object. The details can include contact information about the creator of poster of the document or other data as desired. This data can be hyperlinked and a return button can be provided to return the display back to the screen shown in FIG. 5. Clicking the "details" button to the right of any document brings up the display which can include the name, contact information, and other details about the person who loaded the document into the system, similar information about a person who has the document checked out, and, optionally, a description of the document and information on its change history.

A parameter object X "Forward" simply sends the document to a selected user. A selection screen can be invoked to allow selection of the recipient of the document from the user registry. Of course, since most correspondence can be handled on the server side, the user is, in reality, simply notified of the transfer and the recipient's action to view the document simply invokes a server side feature to display the document. The document is not actually transferred bodily to the recipient since the recipient, as a registrant logged in the user registry, can access it through the server by requesting to do so.

A parameter object U "Check-in" checks in a document that has been checked out. Other users may view the document, but not modify it when it is checked out. This button is not accessible to users that have not checked the document out and may be displayed ghosted or not displayed at all. A similar button can be displayed if a document that is not checked out may be checked out by the user authorized to see the document manager displayed shown in FIG. 5.

A parameter object T "Download" actually transfers a copy of the document to the client computer. Another object S "Delete" allows the document to be deleted. A new document can be added by clicking "New Document" Q. These are fairly conventional notions, except for their placement on the screen and the fact that each is filtered depending on the user's rights.

Note that when a folder is created, access to the folder can be restricted to the creator, shared with everyone (in which case the folder is created in the public directory), or shared with a select group of other users. The other users can be selected by company or organization (providing access to all individuals in the organization) or by individual within an organization. These are all selectable through a linked selection control where if one selects a company in one selection control, it shows employees in the linked selection control.

A parameter object P "Shared" displays a hyperlinked page that shows all users with access rights to the document. This page allows a user that places a document in the document manager or a user that has pertinent modify rights, to alter the parties that have access to the document. Also, it allows a user with read-only rights to see the list of users that can access that document. The names of the sharing parties are hyperlinked to invoke the user's email client to allow fast sending of email (which again may be performed server-side without actual transfer) or conventionally or selectively. If a folder is shared, the word "Shared" appears to the right of the folder. Clicking on "Shared" brings up the list of person who can access the folder, as shown in FIG. 6. Each name is a hyperlink to detailed contact information.

FIG. 7 shows a list of all deals that were completed through the system. The trade number (left column of the grid) is a hyper link to detailed information.

FIG. 8A shows detailed information about a completed trade. It shows the party to the trade, the price or rate, and a description of what was traded. The particular nomenclature is specific to a market. For insurance, for example, price is termed rate, and the summary of a deal is the slip sheet. A complete contract can be attached. Included documents can be downloaded to view on line. The intended signatories to a deal are shown (there can be more than two).

If a signatory has actually signed the document electronically, the date and time are shown. No date and time are shown for parties that have not yet signed. The amount of information displayed on the screen is dependent on the identity of the person viewing the screen. The viewer can be blocked from viewing any information about a deal, or certain fields, such as the contract details or the name of signatories.

Note that the detail screen of FIG. 8A would also show attached exhibits. The FIG. 8A display is the basic device for signing deals. A similar device would be used for signing documents.

Referring to FIG. 8B, all of the information necessary to document a deal is pulled together through the screen below. The deal summary includes highly structured information on parties, dates, terms, etc., as well as unstructured information in the form of attachments. The bottom part of the page allows the person registering the deal to designate the intended signatories. When the signers affix their electronic signature, they are doing so to all of the documents in the deal, including the attachments. These are archived and protected from tampering using secure hash technology. In this way it is possible to create a reliable, on line electronic signature to a complex deal, without risk of repudiation.

Note that any number of exhibits can be added to the UI device of FIG. 8B since the list scrolls from the bottom each time a second exhibit is added. The user interface has self-explanatory elements for defining information about the deal.

Anonymous Mail

For purposes of the following description, a "subscriber" is a person or entity that subscribes to an anonymous mail system to be described below. Certain types of negotiations and communications require anonymous initial contact, followed by some period of anonymous discourse, leading to eventual disclosure of the parties' identities. In the course of a typical sale or business deal, the initiating party begins either by contacting one or more targeted potential trading partners or advertising to a community of potential partners. While the identity of the initial offeror is usually clear in any direct contact, it need not be so in advertising. In certain cases it could be problematic for the initiating party to reveal his or her identity:

A party to a deal can have difficulty controlling the method of contact once the party's identity is known. If a company is known to be in the market for office space, for example, the party may be subjected to badgering by real estate firms outside the established bidding process. Executives of the company may be contacted directly in an effort to influence the decision.

Disclosure of intent may adversely affect the market. If a large company begins to acquire land in an area, the price can rise very quickly. Simple exploration of an option can make the option more costly or even impossible.

Disclosure of intent may adversely impact the reputation or standing of a company. An insurance company that determines that it is over exposed to a certain peril (e.g. hurricane losses in the Southeastern U.S.) would reveal that situation to their competitors and investors by a large public solicitation.

While anonymity can be crucial for the initiator of a deal, it can be equally important for the respondent for the same reasons. The need for controlled anonymity has been addressed by several methods that were initially developed for paper communications and have been extended to analogues in telephonic and computer communications.

- Numbered mail boxes, including government and private
- Communications through a mediator
- Anonymous voice mail drops
- The use of pseudonyms in computer e-mail and dialogs.

These methods have several serious shortcomings:

- The method may only allow anonymity from one side.
- There is no inherent mechanism to validate the credentials and intent on an anonymous party
- Use of a pseudonym may invalidate its future use by associating the name with a specific party

- Manually mediated communications are slow
- The creation and deletion of pseudonyms may not be completely within the control of the party, imposing an overhead cost (in cash or labor) and/or delay in creating a new name
- In most systems, a person with multiple pseudonymous mailboxes or e-mail addresses will receive communications in several different places (mailboxes or accounts), thus requiring multiple logons/passwords.
- Routing of messages received anonymously requires manual forwarding to all relevant parties by the individual with access to the anonymous mail box or email account.
- There is no mechanism to reveal actual identities in a secure and mutually acceptable way.

The present invention addresses these deficiencies by providing two-way anonymous communications, a central point of collection for messages sent to multiple pseudonymous addresses, connection of multiple parties to a single anonymous account, and a mechanism to reveal identities to all parties to a deal simultaneously, by mutual consent. In summary, the anonymous mail system is a server side system that allows clients to create anonymous handles on the fly. It also allows them to share anonymous handles among multiple recipients so that the group of recipients appears as a single recipient to the sender using the anonymous handle. It is like a transparent mailing group. When mail is sent to an anonymous handle, it is sent to all members of the group.

<u>Multiple Systems</u> – In contrast to the first-generation anonymous mail system, the present system allows for multiple anonymous mail (Amail) systems. Each Amail system operates in association with a conventional e-mail server, and uses the e-mail server for communications with non-subscribers, subscribers to Amail systems other than the local one, and for forwarding messages to the subscribers Email client software.

<u>Registration</u> – Subscribers to an anonymous mail system (Amail) each complete a registration that provides:

- Contact information (name, address, telephone number, fax, etc.)
- Information to determine whether they the party is qualified to

participate in the communications exchange. For example, if the system were to be used between and among real-estate agents, registrants to the system might be required to supply a real estate license number.

- Association with an organization (if appropriate)
- Additional information on the individual or organization that may be of use to others in the Amail system to determine the suitability of the party as a partner in negotiations.

The additional information can include such factors as credit ratings, assets, or the region in which the company does business. The specific information required depends on the application. Insurance, real estate, energy marketing, etc. would all have different data of interest.

Validation – Depending on the business model and role of the organization operating the Amail exchange, the organization can either accept the information provided by the subscriber, or verify the information and provide verification as part of the service. Upon acceptance of a subscription applications and validation of the background information if necessary, the use is assigned an Amail user ID and password.

In the first version of the Amail system, logon was automatic from the general application (CATEX); there was no separate user ID and password. In alternative versions, the Amail system can provide its own user ID and password, with the ability to bypass logon when it accessed from other applications with acceptable user validation. All of the actual contact information and validation information are maintained in a database. Validation information was not provided in the first version of CATEX.

Assignment of an Email address – Each subscriber must provide an Internet accessible Email address or be assigned an e-mail address in the Amail system. The first version of the Amail required that the user have an Email address on the system. The new version works directly with e-mail systems other than the Amail.

Logon – Subscribers access the Amail system by connecting an Amail web page provided either over the Internet or on an Intranet. The subscriber enters a user name and password. The first version of Amail was not browser-based and worked only over a LAN or WAN, not over the Internet or an intranet.

Available functions – After logon, the subscriber can access the following functions:

- Manage aliases
- Compose an anonymous message
- Read Amail messages. In the original CATEX system, the user could not access messages from within the Amail application.
- Log off

Managing Aliases – Aliases are directly under user control. After logon, a user can:

- Add a new aliases
- Delete an existing alias
- Create a free-form note associated with a new alias, or edit the note for an existing alias that will be accessible to recipients from the alias.
- Identify other subscribers to whom messages to alias should be forwarded
- Identify other subscribers with permission to generate messages from the alias

These last two features make it possible for a group of subscribers to share an alias, allowing them share communications and work together more effectively. The user will:

Compose an anonymous message – After logon, a user can create and send an anonymous message. After the option is selected, the system will display a message creation screen with the following features:

1. A list of aliases currently owned by the user (i.e. created by the user and not deleted), for the user to select the alias from which the message will originate.
2. A subject box for the mail.
3. A list of the e-mail and alias addresses to which messages can be sent for the user to select one or more. The original version could only send to one alias. The user can also supply an Internet e-mail address off system.
4. A list of the e-mail and alias addresses to which copies of the messages can be sent for the user to select one or more. The user may also supply an Internet e-mail address off system. The original version did not include a "CC" feature.
5. A space where the message can be typed, allowing for users to paste text copies form another system using the Windows-based clipboard utility.

6. A check box to select whether the sender is willing to reveal his identify to the recipient on mutual consent.

7. A check box to select whether the copies of the message should be sent to other subscribers who share the Alias. The original version allowed only one subscriber to access an alias.

Delivery of Messages – After an Amail message has been composed (see step 7), it is delivered as follows.

1. The body of the email message is modified by adding a header including routing information and an indication of whether the sender is willing to reveal identities if there is reciprocal concurrence. The message would appear as shown below. The items in italics are new since the original (prior art) version. The first generation of the anonymous mail system did not allow for communications between multiple Amail systems and, hence, did not list the Amail system name in the list of respondents. The first generation system also did not allow for multiple recipients.

> *This message was sent anonymously from alias: Amail system name: alias*
> The message was sent to:
> Amail system name: *alias*
> Amail system name: alias (cc)
> Amail system name: alias
> The sender is willing to reveal identities.
> [Original body of the message]

2. If the message is sent to a specific, non-anonymous e-mail address, Amail composes and transmits a standard Email message. The sender is listed as "amail.admin.alias@xxxxx" where "xxxxx" is the address of the standard mail server supporting the mail system. Off-system access was not a feature of the first version.

3. If a message is sent to an alias on the local or any other related Amail system, and the owner of the alias has an off system email address, a message is sent as in step 1, above. In addition, however, the message is stored in an Amail message database for access through the Amail system interface. The original version did not have an Amail message database.

4. If a message has been sent to an alias for which there is no associated conventional mail account, the message is stored in the Amail message database. The Amail message database contains a repository for all messages, listing the subscriber(s) associated with the alias to which the message was addressed. The database contains the message (including sender, addressees, and ccs), date and time of transmission, and the alias of the subscriber to which the message was sent. The original version did not have an Amail message database.

5. If the option was checked to send copies to other that share the alias (see above), copies of the message are placed in the message database for the subscribers associated with each of the aliases.

<u>Receipt of Messages</u> – Messages sent from the Amail system can be received in a standard e-mail client by Amail subscribers and non-subscribers.

Amail subscribers can also receive messages through an Amail reader interface. All messages received are placed in the Amail message database (see above). Since an alias can be associated with more than one subscriber, the Amail message database can list more than one subscriber as an "owner" of the message even if it was sent to only one alias. When a user logs on and selects the option to read Amail messages (see above) the messages are rendered as an HTML page through a browser. Messages to all of the aliases associated with the user are displayed. Each message has a hotlink to respond to send a message back to the sending alias. Each message also has a link to display the background and validation information and note associated with the alias (see above). The original version did not provide an Amail viewer nor did it provide for display of validation information.

Responding from off System from Amail – Individuals from off system can respond to Amail messages using the standard reply feature of their mail server. Messages will be returned to the reply address (see above). Messages received by the conventional e-mail server supporting the Amail system will forward the message to the Amail message repository for the alias listed in the return address. Responding from a standard Email client was not provided in the original version.

Flip Widget

Increasingly, computer applications are delivered through browsers over the Internet or an intranet. There are many design considerations in building a system for browser delivery in contrast to delivery as conventional client server application. Two related considerations are the graphic richness of a browser screen and the time lag to render a new screen. Partly because good web pages contain complex graphics and partly because the Internet can be a relatively slow network, it is important to design a web application to make few unnecessary wholesale screen changes. It is more economical from the perspective of data transmission and, hence, from response time, to create a "flat" rather than "deep" hierarchy of screens, and change only the part of a screen that is minimally necessary.

For example, it is better in a data query to provide a single screen that allows a user to specify a state and city within the state than to provide a first screen for the state, followed by a second screen for the city. As the function of screens becomes more complex, however, it becomes an increasingly difficult challenge to fit all of the options onto the screen (particularly when a user selects a lower screen resolution) and while maintaining a clean appearance. The invention described here provides a tool that allows the Internet application developer to display an effectively unlimited number of options in a very small space using a very familiar and intuitive display feature.

Appearance – The "Flip Widget" tool renders a graphical object representing two rows of file folders, overlapping. The labels on the front row are visible, the labels on the second row are obscured by the front row of tabs, but the edges of the apparent back tabs are visible. The number of the apparent tabs displayed in each row is a function of the screen resolution and the length of the longest label entered by the user.

The Flip Tab – In one embodiment, the rightmost tab on the front row is labeled "FLIP". When a user actuates this tab, the response is as described below.

Database of labels and links – In creating the display, the application programmer enters a set of paired values. Each pair consists of (1) text of the label to be displayed and a tab, and (2) the name of an HTML link, either within or external to the page to be rendered when the tab is selected.

Action – Upon rendering a page containing the flip widget, the two-row tab display shows the first "n" options from the list of labels and links. The value of "n" represents the maximum number that can be displayed while allowing room for the flip tab. Upon clicking any of these tabs, the corresponding link is executed. Upon clicking the flip tab, the two-row tab display is changed to reflect the next "n" options from the list of labels and links, retaining the flip tab on the right. If there are fewer than n options remaining, the flip widget will either display the last n options, or whatever number remain supplement by as many options are needed from the start of the list. Clicking the flip tab when the list has been completed starts the cycle over again with the first option.

Referring to FIGS. 9 and 10, a flip widget in a first state is shown in FIG. 9. In the first state, any of the tabs A through E can be selected and the corresponding set of controls displayed. For example, in FIG. 9, tab B has been selected and the controls 430-432 are displayed. If the flip tab 410 is selected, a next row of tabs is brought forward so that the display appears as in FIG. 10 with tabs F through J showing. In FIG. 10, tab G has been selected and the corresponding controls 435-437 are displayed.

FIGs. 9A and 10A show a more detailed example of how a flip widget can be used to organize functions available to a user. For example, suppose that one application is a commodity futures trading system that permits a user to execute trades, review prices, and obtain other information relating to various metals such as gold, silver, and platinum. As shown in FIG. 9A, for example, controls or functions

430, 431, and 432 (e.g., execute a trade, review current prices, and the like) are associated with a "gold" category and can be invoked easily when that category is at the forefront of the flip widget as shown. Clicking one of the other tabs (e.g., silver tab 400) would bring the functions associated with that category to the forefront while allowing the user to readily select other categories visible behind the front. Clicking "other markets" tab 410 would change the selection of front-row tabs to a different set of categories, as shown in FIG. 10A. The "other markets" tab 410 could be continually clicked to rotate through a plurality of groupings of markets, each having a set of functions or controls associated therewith.

A flip widget can be implemented in conjunction with the first or second embodiments of the present invention in order to permit many different functions to be displayed in a small screen space. The flip widget is a device to organize many different functions in a logical way, and can be used as a tool for building an interface to multiple applications. As one example, in a DCE (described in more detail below), there may exist n functions (e.g. bulletin boards, chat rooms, e-mail, a-mail, transaction engines, and the like) the specific availability of which can be defined by a user who creates the collaborative environment. This collection can change over time. Accordingly, the interface cannot be "hard coded" for a particular user.

One way to represent an indefinite (and potentially large) number of functions in a small space is with tabs resembling a file folder, with a graphic element representing hidden cards, implying that the user can reach the functionality on the cards by paging (i.e. flipping) to them. The flip widget makes it possible to provide a link to a list of applications maintained in a database rather than requiring that they be hard coded. Programming logic for storing folder labels in a database, linking those labels with associated functions and activating them using browser-type buttons, and for performing the display features described above, are conventional and no further elaboration is necessary. Although the "flip widget" provides one method of structuring a user interface to structure a user's view of application functions, other methods can of course be used.

B. <u>DYNAMIC COLLABORATIVE ENVIRONMENT EMBODIMENT</u>

In a second embodiment of the invention, a dynamic, user-defined collaborative environment can be created in accordance with a set of tools and method steps. As explained previously, this system differs significantly from conventional networked environments in that: (1) the environment (including access and features) is user-defined, rather than centrally defined by a system administrator; (2) each environment can be easily destroyed after completion of its intended purpose; (3) users can specify a group of participants entitled to use the environment and can define services available to those participants, including offering participation to unknown potential users; (4) the networked environment (including access features and facilities) can cross corporate and other physical boundaries; and (5) the environment offers a broad selection of tools that are oriented to communication, research, analysis, interaction, and deal-making among potential group members. Moreover, in a preferred embodiment, the environment is implemented using web browser technology, which allows functions to be provided with a minimum of programming and facilities communication over the Internet.

FIG. 11 shows various method steps that can be carried out to define, create, and destroy an environment according to a second embodiment of the invention. The term "environment" as used herein refers to a group of individuals (or computers, corporations, or similar entities) and a set of functions available for use by that group when they are operating within the environment. It is of course possible for one individual to have access to more than one environment, and for the same functions to be available to different groups of people in different environments.

The process of creating a collaborative environment involves the migration of tools and information resources available in the library of the environment generator into a specific collaborative environment. The collaborative environment can include / link to any application available to the environment generator. It can also include applications specific to the environment provided that theses are accessible through Internet protocols.

Underlying the environment is a directory of users, information about users, and their authorities. The core structure for the environment user database should conform to a directory standard – typically DAP (Directory Access Protocol) or LDAP (the lightweight directory access protocol). The environment generator has access to its own directory of users and to the user directories of the environments it has generated. The directory of an environment can be populated initially by selecting users from the environment generator's directories. These are added to the directory of the environment in one of two ways depending on the specific implementation. Directory records can be copies from the environment generators user database to a separate database for the environment or a flag can be added to the user data record in the environment generators users database to indicate that the user has access to the environment. The second, simple model is useful when all users in an environment have equal authority. A separate user database (directory) is necessary for an environment when the environment has its own security / authority model.

Additional members can be added through a set of standard application / subscription routines. These then become known to the environment generator (as well as the specific environment) providing the foundation for greater speed and efficiency in creating subsequent environment.

Beginning in step 1101, a new group is created by identifying it (i.e., giving it a name, such as "West High School Research Project," and describing it (e.g., providing a description of its purpose). The process of creating a group and defining functions to be associated with the group can be performed by a user having access to the system without the need for system administrator or other similar special privileges (e.g., file protection privileges, adding/deleting application program privileges, etc.). In this respect, environments are, according to preferred embodiments, completely user-defined according to an easy-to-use set of browser-driven user input screens. The principles described herein are thus quite different from conventional systems in which a central system administrator in a local area network can define "groups" of e-mail participants, and can install application programs such as spreadsheets, word processing packages, and the like on each computer connected to the network. Moreover, according to various preferred embodiments, the facilities provided to group members can be provided through a web-based interface, thus avoiding the need to install software packages on a user's computer.

It is also contemplated that various methods of obtaining payment for creating or joining groups can be provided. For example, when a new environment or group is created, the person or entity creating the group can be charged a fixed fee with payment made by credit card or other means. Alternatively, a service fee can be imposed based on the number of members that join, the specific functions made available to the group, or a combination of these. Moreover, fees could be charged to members that join the group. The amount of the fee could also be based on the length of time that the environment exists or is used.

Although not specifically shown in FIG. 11, step 1101 can include the step of creating a new entry in a database table (e.g., a relational or object-oriented database) to store information concerning the new group and the environment in which the group will operate. Database entries related to the group, including some or all of the information described below, can be created as the environment is defined. It is assumed that one or more computers are linked over a network as described in more detail below in order to permit the environment to be created, used, and destroyed, and that a database exists on one or more of these computers to store information concerning the environment.

In step 1102, the group members are identified. According to various embodiments, the group members can be identified in three different ways (or combinations thereof), as indicated by sub-steps 1102a, 1102b, and 1102c in FIG. 11. It is contemplated that group members can span physical networks and computer systems, such as the Internet. Consequently, group members can include employees of different corporations, government agencies, and the like. In contrast to conventional virtual private networks, both the group members and the functions made available to those group members are entirely user-selected, thus permitting a broad range of persons to easily create, use, and destroy virtual private networks and associated functionality.

First, in step 1102a, group members can be identified by selecting them from a list of known users that are to be included in the group. For example, within a corporation or similar entity, a list of internal e-mail addresses can be provided, or an electronic version of a phone list or other employee list can be provided. If the hosting computer system is associated with a school, then a list of students having accounts on the computer (or those in other schools that are known or connected to the host) can be provided. From outside a corporate entity, users can be selected based on their e-mail addresses (e.g., by specifying e-mail addresses that are accessible over the Internet or a private or virtually private network). In this step, the environment creator specifies or compels group members to belong to the group.

Second, in step 1102b, group members can be invited to join the group by composing an invitation that accomplishes that purpose. For example, a group creator may choose to send an invitation via e-mail to all members of the corporation, or all members of a particular department within the corporation, all students in a school or region, or members of a previously defined group (e.g., the accounting department, or all students in a particular teacher's class). The invitation would typically identify the purpose of the group and provide a button, hyperlink, or other facility that allows those receiving the invitation to accept or decline participation in the group. As those invited to join the group accept participation, their responses can be stored in a database to add to those members already in the group. Invitations could have an expiration date or time after which they would no longer be accepted. As invitees join the group, the group creator can be automatically notified via e-mail of their participation.

Third, in step 1102c, group members can be solicited by way of an advertisement that is sent via e-mail, banner advertisement on a web site, or the like. Persons that see the advertisement can click on it to join the group. It is also possible for advertisements to have a time limit, such that after a predetermined time period no more responses will be accepted. The primary difference between advertising participation in a group and inviting participation in a group is that invitations are sent to known entities or groups, while advertisements are displayed to potentially unknown persons or groups.

It will be appreciated that group members can be selected using combinations of steps 1102a, 1102b, and 1102c. For example, some group members can be directly selected from a list, while others are solicited by way of invitation to specifically identified invitees, and yet others are solicited by way of an advertisement made available to unknown entities.

In step 1103, the functions to be made available to the group are selected. For example, the group can be provided with access to an auction transaction engine; a survey tool; research tools; newswires or news reports; publication tools; blackboard facilities; videoconferencing facilities; and bid-and-proposal packages. Further details of these facilities and tools are provided herein. The group creator selects from among these functions, preferably by way of an easy-to-use web browser interface, and these choices are stored in a database and associated with the group members. Additionally, the group creator can specify links to other web-based or network-based applications that are not included in the list by specifying a web site address, executable file location, or the like. The group creator can also define shared data libraries that will be accessible to group members.

In step 1104, the environment is created (which can include the step of generating a web page corresponding to the group and providing user interface selection facilities such as buttons, pull-down menus or the like) to permit group members to activate the functions selected for the group. In some embodiments, access to the group may require authentication, such as a user identifier and password that acts as a gateway to a web page on which the environment is provided. Other techniques for ensuring that only group members access the group functions and shared information can also be provided. A web page can be hosted on a central computer at an address that is then broadcast to all members of the group, allowing them to easily find the environment.

In step 1105, group members collaborate and communicate with one another using the facilities and resources (e.g., shared data) available to group members. In the example provided above, for example, a group of high school students collaborating on a school research project could advertise for survey participants; conduct an on-line survey; compile the results; communicate the results among the group members; brainstorm about the results using various brainstorming tools; conduct a videoconference including group members at various physical locations; compile a report summarizing the results and exchange drafts of the report; and publish the report on a web site, where it could optionally be offered for sale through the use of an on-line catalog transaction engine. The group could even contact a book publisher and negotiate a contract to publish the report in book form using bid and proposal tools as described herein.

In step 1106, after the environment is no longer needed, it can be destroyed by the person or entity that created the group. Again, in contrast to conventional systems, the destruction of the environment is preferably controlled entirely by the user that created the environment, not a system administrator or other person that has special system privileges. Destruction of the environment would typically entail deleting group entries from the database so that they are no longer accessible.

FIG. 12 shows one possible system architecture for implementing the steps described above. As shown in FIG. 12, an Internet Protocol-accessible web server 1201 is coupled through a firewall 1202 to the Internet 1203. The web server includes an environment generator 1201a which can comprise a computer program that generates user-defined environments as described above. Further details of this computer program are provided herein with reference to FIG. 21.

Web server 1201 can include an associated system administrator terminal 1204, one or more CD-ROM archives 1205 for retaining permanent copies of files; disk drives 1206 for storing files; a database server 1207 for storing relational or object-oriented databases, including databases that define a plurality of user-controlled environments; a mail server 1208; and one or more application servers 1209 that can host application programs that implement the tools in each environment. Web server 1201 can also be coupled to an intranet 1210 using IP-compatible interfaces. Intranet 1210 can in turn be coupled to other application servers 1211 and one or more user computers 1212 from which users can create, participate in, and destroy environments as described herein, preferably using standard web browsers and IP interfaces. Web server 1201 can also be coupled to other user computers 1217 through the Internet 1203; to additional application servers 1215 through another firewall 1216; and to another IP-accessible web server 1213 through a firewall 1214.

It will be appreciated that the system architecture shown in FIG. 12 is only one possible approach for providing a physically networked system in which user-defined network environments can be created and destroyed in accordance with the principles of the present invention. It is contemplated that application programs that provide tools used in a particular user-defined environment can be located on web server 1201, on user computers 1217, on application servers 1215, on application servers 1209, on application servers 1211, or on any other computer that provides communication facilities for communicating with web server 1201. It will also be appreciated that web pages that provide access to each user-defined environment need not physically reside on web server 1201, but could instead be hosted on any of various computers shown in FIG. 12, or elsewhere.

Reference will now be made to exemplary steps and user interfaces that can be used to carry out various principles of the invention, including steps of creating a group, selecting group members, and defining functions to be made available to group members in the environment.

FIGS. 13A through 13C show one possible user interface for creating a group and identifying group members. In FIG. 13A, a user gains access to an environment creation tool by way of an authentication process. This may be a simple username and password device as shown in FIG. 13A, or it could be some other mechanism intended to verify that the user has access to the environment creation tool. In the case of a corporation, school, or other entity that already provides a log-in procedure to access the entity's network, such log-in procedure could serve to authenticate the user for the purpose of creating a new environment. It should be appreciated that user authentication is not essential to carrying out the inventive principles. Moreover, although it is contemplated that for ease of use (and to minimize programming) web browsers and web pages be used to receive user-defined information to create each environment, other approaches are of course possible.

In FIG. 13B, the user is prompted to create a new group by supplying a group name (e.g., "Joe's Homework") and a brief description of the group. This information is preferably stored in a database file and associated with group members and functions available to those group members.

In FIG. 13C, the user is prompted to identify group members. As described previously, group members are preferably identified in one of three ways (or combinations of these): (1) selection from a list of known group members; (2) inviting known candidates to join the group; or (3) advertising for new members. When the user clicks one of the options in FIG. 13C, he or she is prompted to supply additional information as shown in FIGS. 14A through 14C.

Beginning with FIG. 14A, for example, group members can be individually specified by entering an e-mail address (e.g., an internal or external e-mail address) in a text form data entry region and/or by selecting from a previously known list. This screen permits the user to compel attendance in the group by specifying names and/or e-mail addresses to which group messages will be sent. All those added to the group in this manner will be provided with access to the environment corresponding to the group. Aliases and pre-defined groups could also be specified as the basis for membership (e.g., all those in the accounting department of a corporation, or all students in a high school).

Each member of a group might have a group email account, or they may use an off-system email account. Off-system email addresses can be maintained in a database of users. Mail sent to the group email address is preferably forwarded off-system, protecting the actual email address of the person unless that person wishes to give out that address. New members can be added until the group is completed. Although not explicitly shown in FIG. 14A, it is contemplated that new members can be added to a previously defined group after the environment has already been created.

When group members are selected or specified, the user creating the environment can also create a password for each user in the group in order to enable those in the group to access the environment. Alternatively, when a user visits the environment, the environment can retrieve a "cookie" from the user's computer to determine whether the user is authorized to access the environment. If no cookie is available, the user could be prompted to supply certain authentication information (e.g., the company for whom he or she works, etc.) In yet another approach, authentication could occur by way of e-mail address (i.e., when the user first visits the environment, he or she is prompted to enter an e-mail address). If the e-mail address does not match one of those selected for the group, access to the environment would be denied.

Turning to FIG. 14B, prospective group members can also be "invited" to join the group. The user creating the environment can specify one or more e-mail addresses to which an invitation will be sent. The invitation can be a simple text message, or it could be a more sophisticated video or audio message. An expiration date can also be associated with the invitation, such that responses to the invitation received after the date will not be accepted. Software resident in web server 1201 (FIG. 12) receives responses to the invitations and adds members to the appropriate group or drops them if the expiration date has passed or the prospective group member declines participation. Prospective members can join the group by sending a reply with a certain word in the message (e.g., "OK" or "I join"); by clicking on a button in an e-mail message; or by visiting a web site identified in the invitation.

Turning to FIG. 14C, group members can also be solicited by creating an advertisement directed primarily at potential group members that are unknown. The advertisement could include, for example, a banner ad comprising text, video, and/or audio clips. The graphic should conform to the size designated for the ad on the web page. The ad could be posted on a web site by uploading the graphic through a web interface and, optionally providing a URL on the screen of FIG. 14C to link to if the advertisement is clicked. Software on the group page can render advertisements on a page either (a) every time the page is displayed, (b) in rotation with other ads; or (c) when characteristics of the user match criteria specified for the ad.

The advertisement can include an expiration date after which responses would no longer be accepted. Advertisements could range from the very specific (e.g., an advertisement posted on a school's home page advertising participation in Joe's research project on drug use at the school) to more general (e.g., an advertisement that says "we're looking for minority contractors looking to establish a long-term relationship with us" that is posted on web sites that cater to the construction industry.

A qualification option can also be provided to screen prospective group members. For example, if an advertisement seeks minority contractors to participate on a particular construction project, selecting the "qualify" option would screen responses by routing them to the user that created the group (or some other authority) before the member is added to the group. Those responding to the advertisement could be notified that they did not pass the qualifications for membership in the group, or that further information is required (e.g., documents evidencing qualifications) before participation in the group will be permitted. Alternatively, an automatic qualification process can be provided to allow a prospective member to join if the person fills in certain information on the response (e.g., e-mail address, birthdate that meets certain criteria, or the like).

As shown in FIG. 15, a banner ad displayed on a web site invites minority contractors to join a group that bids on information technology contracts. Those interested in the advertisement click a button, which leads them to another site (not shown) requiring that they provide certain information (qualification information, name, age, company registration information, etc.) This information is then forwarded to web server 1201 which either pre-screens the information according to pre-established criteria, or notifies the user creating the group that a prospective member has requested access to the group. In the latter case, the user could screen the applicant and grant access to the group.

FIG. 16 shows one possible user interface for selecting communication tools to be made available to group members. This screen can be presented to the user creating the environment after the group has been identified and its members selected. It is contemplated that a variety of communication tools can be provided, including a bulletin board service; advertisements; white pages (e.g., a listing of members, their e-mail addresses, telephone numbers, and the like); yellow pages (e.g., a listing of services or companies represented by group members, with promotional and contact information); document security (e.g., shared access secure document storage services); anonymous e-mail (described above with respect to the first embodiment); threaded dialogs; a group newsletter creation tool; videoconferencing; and even other user-provided applications that can be specified by name and location (e.g., URL). Details of these services are provided below.

According to various preferred embodiments, dynamic collaborative environments are designed to integrate tools from multiple sources provided that they are web-accessible (i.e., they operate according to Internet Protocol and/or HTML-type standards). The categories listed above provide a reasonable taxonomy of the tools necessary for collaboration, but this list can be extended to include virtually every class of software such as computer-assisted design, engineering and financial analysis tools and models, office applications (such as word processing and spreadsheets), access to public or proprietary databases, multimedia processing and editing tools, and geographic information systems. The following describes some of the communication tools that can be provided:

Bulletin boards. A bulletin board (see, e.g., FIG. 2) lists notices posted by group members, which may be offers to buy or sell, but need not be limited to such offers. Many types of bulletin board services are of course conventional and no further discussion is necessary in order to implement one of these services. Nevertheless, in one embodiment the following data items (attributes) can be provided for each notice appearing on the bulletin board: an item number, a title, the date posted, and one or more special attributes defined by the user. The attributes may include a field to indicate whether a listing is a "buy" or "sell " offer. The board can be provided with an integrated sorting capability. By clicking on the heading of each column, the user can sort the entries in, alternately, ascending or descending order. Thus, it is possible to organize the records from oldest to newest or newest to oldest, or to separate buy and sell offers. To limit the values on a board, a search capability can also be provided, such that only those entries that meet the search criteria are displayed.

Advertisements. In a typical environment of a dynamically created network there are a number of fixed places for advertisements – the top of a page for a banner, the bottom of a page for a banner, and space on the side for small ads. The creator of the environment may choose to use none, any, or all of these spaces for advertisements. Once a space is designated for advertising, group members may place adds by completing a template that provides payment information (if required), the text for the ad (any standard image format), and a link to be executed if the ad is clicked by someone viewing the ad.

Each user is responsible for providing functionality behind the link. The ad may be displayed persistently (every time a page is displayed), in rotation with other ads for the same place, or may be triggered on the basis of user characteristics including purchasing history. Revenue can be collected for placement (fixed price regardless of how many times an ad is displayed), per time that the ad is displayed, or per click on the ad. The virtual private network provides the front-end to facilitate online placement of the ad. Display can be done by linking pages to standard ad display code, available off the shelf from several sources. This code provides for rotation of the ads. Software for customization (i.e. choosing the ad based on user characteristics) is available commercially from several sources.

White pages. White pages provide a comprehensive listing or directory of members with information about them and information regarding how to contact them. Various types of commercially available software can be used to manage such directories, and it is elementary to code typical directories that have fixed contents for each member.

A web-accessible directory can be used in accordance with various embodiments of the invention. One type of directory that can be provided differs from directories having fixed structures. The key differences are as follows:

(a) User control over information Users enter and maintain their own information directly, rather than through a central organization. This provides more immediate update of data and reduces transcription errors. It makes it simple, for example, for people to change their phone number when they are temporarily working at another location.

(b) Multiple points for quality control. The data regarding each user can be displayed to the user periodically (e.g.30, 60, and 90 days), and the user prompted to update and verify the data. A feedback capability can be provided for members of a group to report errors they find. Email addresses can be "pinged" periodically to determine if they still exist. In addition, server management staff can periodically review accounts that have had recent activity.

(c) Object structure. A directory entry consists of a collection of data elements. These elements include such things as name for addressing (Dr. John D. Smith), sort name (Smith, John D), or primary work telephone (800-555-1212). Traditional mail systems have a fixed number of rigidly formatted elements. In one embodiment, a more flexible approach can be used in that individuals identify which elements they wish to add to the collection comprising their directory entry. For example, a person can add 3, 4, 5 or more telephone numbers attaching a note to each explaining its use (e.g. "for emergencies after 8PM").

(d) Direct link to communications tools. Where a directory refers to a contact method (e.g. a telephone number), the method can be invoked directly from an entry if the necessary software is available. For example, phone number can be dialed, email messages initiated, or a word processing session initiated with letter and envelope templates, preloaded with address information.

(e) Descriptive information. In addition to contact information, each directory can contain information describing the entry (individual or business). The description can be different in each group or it can be the same. The descriptive is free form, with the exception that the user may drop in terms from a group-specific lexicon. This lexicon can include terms specific to the industry (e.g. "fuel system") for the automotive industry, or preferred forms of standard terms (e.g. "California" rather than "CA", "Ca", or "Calif."). Standardization of terms in this way makes search the directory more reliable.

Yellow pages. Conventional "yellow pages" products provide a one level classification of directory entries designed to facilitate identification of and access to an individual or organization with specific interests and capabilities. Within industries, and particularly online, multi-level hierarchical directories are common, with the multiple levels providing more precise classification. There are numerous commercial products for maintaining online yellow page type classification systems.

Any web-accessible directory can be connected to a DVPN group. A preferred method offered with the system integrates the classification system with the descriptive field in a directory entry. Every time a standard term pertaining to a classification is pulled from the lexicon, the entry is added to that classification in the hierarchical sort. In addition to hierarchical access, this correspondence between the traditional hierarchical sort and the free-form description with standardized terms makes it possible to access records via search rather than browsing the hierarchy. Searching makes it possible to identify an organization with multiple capabilities (e.g. "brake repair" and "frame straightening"). This search capability is much like a general web-search using a tool like AltaVista's or Inktomi's search engine and can use the same search engine, but differs in that material being search is in a precisely defined domain (group members), the information being searched is limited and highly quality controlled (i.e. group directory entries), and has a precision rooted in a precise vocabulary (the lexicon used in preparing the description).

Document repository. Any commercial web-enabled document repository can be integrated into a group. Examples are Documentum and PC DOCs. An improved version offered specifically with the DVPN package was described above.

Document security. Within the document repository various tools can be provided to protect the security of documents. These include (1) limiting access to a document to certain people or groups; (2) only displaying the directory entry for documents to people who can access it; (3) password protection; (4) encryption; (5) secure archive in read only mode on a third-party machine; (6) time-limited access and (7) a secure hash calculation.

All of the above are conventional except for time-limited access and the secure hash calculation. Software for limiting access to a document to a certain period is available from Intertrust, among others. A secure hash is a number that is characteristic of the document calculated according to a precisely defined mathematical algorithm. There are several secure hash algorithms, and implementers can develop their won. They are "trap door" in nature. That is, the calculation can be performed with reasonable effort, but the inverse of the function is computationally intractable. The classic example of a trap door function is multiplication of very large prime number (on the scale of hundreds of digits). The product can be calculated with relative ease, but factoring the product (the inverse function) is very time consuming, making if effectively impossible with generally available hardware. This method is used in public key encryption, but can be applied equally well in secure hash, though other trap door functions are preferred, in particular, the one specified by the U.S. Department of Commerce as FIPS standard 180. Code to implement this standard can be developed from published algorithms.

Anonymous e-mail (described above with respect to the first embodiment);

Threaded dialogs. Threaded dialogs are a collection of messages addressing a specific topic, added serially, not in real time. They are threaded in the sense that new topics can branch off from a single topic, and topics can merge. According to one embodiment, threaded dialogs differ from conventional news group functionality in that (1) users can initiate new topics; (2) users can post a message to one topic, then indicate that the message pertains to other topic as well; (3) browsers reading a message may continue down the original thread or one of the alternates if other topics are suggested.

Group newsletter creation tool. A newsletter creation tool can be used to link columns provided by multiple users (and maintained as separate web documents) into a whole through an integrating outline maintained by an "editor". The purpose of the tool is to provide the look and feel of an attractive single document to a disparate collection. To create the newsletter the editor generates an outline identifying an author for each component and a layout. Art for the first page can be provided. Through messaging, the authors are provided a link to upload their content. Content is templated to include a title, date, a by line, one or more graphic elements, a summary for the index, and text. The editor may allow documents to go directly to "publication" or require impose a review and editing step.

Chat groups. Real time chat room software is widely available from many sources including freeware and shareware.

Audio and videoconferencing. Commercially available tools for web-based audio and video conferencing can be included in the group functionality. Examples are Net Meeting and Picture Tel software.

FIG. 17 shows one possible user interface for selecting research tools to be made available to group members. As shown in FIG. 17, various tools such as a mortgage calculator, LEXIS/NEXIS access, news services, Valueline, and other research tools can be provided by checking the appropriate box on the display. All of these research tools are conventional and commercially available (via web-based links and the like).

FIG. 18 shows one possible user interface for selecting transaction engines to be made available to group members. As shown in FIG. 18, many different types of transaction engines can be provided to group members, including electronic data interchange (EDI) ordering; online catalog ordering; various types of auctions; sealed bids; bid and proposal tools; two-party negotiated contracts; brain writing (moderated online discussion) and online Delphi (collaborative estimation of a numerical parameter). The following describes various types of transaction engines in more detail. Enhanced features (i.e., those that differ from conventional products) are highlighted in gray text.

A. Order placement (online catalog) transaction engine

An order placement or online catalog engine allows the buyer to place an order for a quantity of items at a stated fixed price, essentially ordering from an online catalog. The catalog contains the description and specification of the offerings. The catalog may be publicly accessible (Subtype 1a) or provided for a specific customer (Subtype 1b). Prices are included in the catalog but may be customer specific, may vary with quantity purchased, terms of delivery and performance (e.g. cheaper if not required immediately). The catalog can represent a single company's offering or an aggregate of the offerings from several companies. The catalog can range from a sales-oriented web site designed for viewing by customers, to a engine designed only accept orders sent via electronic data interchange (EDI). Note that the catalog can be shopper oriented (i.e. designed to sell) or a simple, machine-readable list of available items and prices. The following describes in more detail steps that can be executed to create an online catalog:

1. Enter and maintain a framework for catalog
    1.1. Enter / delete / edit categories. Categories are titles for groups of items, such as "furniture" or "solvents"
    1.2. Enter / delete / edit subcategories. Subcategories are categories within categories, effectively establishing a hierarchy of products. Example: furniture/dining room/tables.
    1.3. Create groups of categories and subcategories (e.g. "see also...."). The grouping allows a person browsing items to be referred to another category that may contains items of interest. For example, someone may reach the furniture/dining room/tables and then be referred to

furniture/office/conference room tables where other suitable tables may be listed, or to furniture/dining room/chairs to buy chairs that make the table. This cross-referencing transforms the hierarchical arrangement of categories into a web.

2. Enter / edit / delete items in catalog by entering and updating the information listed below. The system allows users to enter this information and provides basic quality assurance.
   2.1. Catalog item number
   2.2. Supplier part number(s)
   2.3. Name of item
   2.4. Description
   2.5. Photos and drawings
   2.6. Specifications (depends on item type). Different items have different specifications. For example, a computer printer can have color vs. black and white, dots per inch resolution, paper size, etc. In contrast to a fixed, hard coded catalog, the specification section of the general purpose catalog engine the user prepares the specification section by selecting parameters from a list and then specifying a value for that parameter. The parameter list contains values such as length, width, height, voltage, color, resolution etc. It is can be extended by the manager of the auction environment. A lister selects a necessary parameter (e.g. length, then enter the value, such as 14"). The specification section is a concatenation of individual specifications.
   2.7. First available date
   2.8. Last available date
   2.9. Category (categories) into which the item fits
   2.10. Alternate suggestion(s) if product not available
   2.11. Related and associated products (e.g. printer supplies for a printer or other household items with the same pattern.
   2.12. Additional information at the option of the individual or organization listing the item.
3. Enter / update pricing information
   3.1. Simple price. The fixed prices is per item or per unit. The price must

specify the

3.2. Pricing algorithm -- link to code for pricing algorithm

4. Take Orders

There are two variants: 4a: manual purchase in which a person browses a catalog and selects and item for purchase and 4b: automated order in which a purchase is initiated by an electronic message.

Variant 4a: Manual Purchase

4.1. Potential buyers access the catalog by drilling down through the category / subcategory tree or

4.2. Buyers search fields in catalog to identify the appropriate item. The search may examine the title, description, or any of the specification fields.

4.3. Display general information for item(s) meetings specifications

4.4. Allow user to modify search or to select specific item if the items displayed to do not meet his requirements

4.5. Display detailed information for selected item

4.6. Display the fixed price or calculate price if prices is based on an algorithm. The pricing algorithm may include parameter such as characteristics or affiliation of the users (e.g. affiliated with a pre-negotiated discount program) , delivery date and mode, and quantity.

4.7. Offer the option to purchase or search again if they choose not to purchase.

4.8. If the buyer opts to proceed with the purchase, then check the availability of the item by linking to the sellers inventory system

4.8.1. If the item is available then execute an 'add to basket'. That is, place it on a list of items designated for purchase.

4.8.2. If the item is not available, then execute the contingent response:

4.8.2.1. Offer delivery at predicted date

4.8.2.2. Terminate the sale, but offer to deliver or notify when next the item is next available.

4.8.2.3. Suggest alternate items

4.8.2.4. Report 'sorry' and abort transaction

4.9. Offer option to purchase addition options

4.9.1. If offer is accepted, execute from step 4.1

4.9.2. If offer is not accepted, proceed with step 4.10

4.10. Conclude the transaction

4.10.1. Collect shipping information, offer options

4.10.2. Collect payment information

4.10.3. Validate payment

4.10.4. Summarize order

4.10.5. Obtain final authorization

4.10.6. Generate receipt

Variant 4b: automated order, done using an EDI (electronic data interchange) message

4.1 Accept requests for item

4.2 Return price and confirmation of availability

Note that users may conduct transactions without employing EDI. It is possible, however, for members to agree on a transaction EDI format either by completing a template within the system or selecting a pre-established EDI format from a library. This library can include formats developed by recognized standards organizations (e.g. UNEDIFACT or ANSI) or formats developed specifically for an industry or a trading environment. Once there is agreement on a format, transactions can be initiated, concluded, and confirmed through the exchange of appropriate EDI messages. As many commercial ordering, accounts payable, accounts receivable and enterprise resource planning systems have an EDI interface the collaborative environment should have the capability to forward the message to the order fulfillment system.

B. English Auction Transaction Engine

In an English Auction, a single item is offered for sale to many buyers. The auction can be open or limited to pre-qualified bidders. The buyers offer bids in turn, each succeeding all prior bids. The highest bid received at any point in the auction is visible to all buyers. The identity of the highest bidder may or may not be visible to traders. Buyers may increase their bids in response to this information. Award is to the highest bidder at the end of trading. The end of trading is reached when there are no higher bids during an interval that may be formally defined or determined by the manager of the auction at the time of execution.

There are two models for the access to the transactions. In the first model, all buyers and sellers are members of the group. In the second model, all sellers are members of the group, but buyers can include members and non-members. If non-members are allowed to buy, the creator the transaction must enter a new URL for buyers. This is a sub-URL of the main group URL. A registration process may be established for the buyer URL.

In live auctions (as opposed to online) all traders are connected at the same time, and the duration of the auction is brief – typically only a few minutes. In online trading, it is not necessary for all of the bidders to be present (i.e. connected at the same time). To distinguish between these two options they are designated (a) concurrent (everyone bidding at the same time) and (b) batch (not everyone connected simultaneously. The manager of the auction can set the minimum bid and the minimum increment.

1. The first step in conducting an auction is to collect information on the items being offered for sale. This is done online. The information collected includes:
    1.1. Identity of seller. Note that the business rules of the auction may require advance registration of sellers to verify their identity.
    1.2. Descriptions, optionally including attachments and photographs, independent certifications or appraisals, and anything else in digital form necessary or useful in determining the value of the item.
    1.3. Reserve price
    1.4. Minimum increment
    1.5. Time offered for sale
    1.6. Time bidding is scheduled to end
    1.7. Verify the seller's consent to the rules of the auction house regarding delivery, payment, responsibility for non payment, etc.
2. If the business rule of the auction house is to require payment up front, collect payment either by:
    2.1. Debiting a deposit account
    2.2. Charging to account for billing
    2.3. Collecting online payment such as through a credit card.
3. Post information about auction, including:
    3.1. Description of items to be auction
    3.2. Auctions rules:
        3.2.1. Qualification process for bidders

3.2.2. Time of bidding

3.2.3. Criterion for ending bidding – time between bids

3.2.4. Legal statement – responsibilities of buyer and seller, limitation of liability

4. Execute qualification process (optional)

4.1. Admit bidders who are qualified based on past participation

4.2. Provide fill-in-the blank qualification form new bidders

4.3. Collect information

4.4. Conduct automated review or manual review

4.5. Inform prospective bidder of qualification or not

Variant (a): concurrent auction

5. Conduct Auction

5.1. Fifteen minutes prior to appointed time for auction, display "Welcome" screen with space for qualified bidder to enter an alias or handle to be used in the auction. Screen should have a description of the object. Show time until auction starts. Auto refresh at 15 second intervals.

5.2. At appointed time, display the main auction page with the following information:

5.2.1. Description / picture of item for auction stored in a separate, static frame of the PC so that it does not need to be downloaded each cycle.

5.2.2. Current bid (initially the reserve price)

5.2.3. Suggested next bid (e.g. current + 3 * increment)

5.2.4. Button to accept suggested next bid

5.2.5. Field to enter bid higher than suggested next

5.2.6. Handle of the highest bidder

5.3. Refresh main auction page at 15 second intervals

5.4. Collect bids, either

5.4.1. Notice that the suggested bid was accepted

5.4.2. Bid higher than accepted bid

5.4.3. If new bid is lower than current highest, disgard

5.4.4. If higher than current highest then

5.4.4.1. Log identity of highest bidder

5.4.4.2. Update highest bid

5.4.4.3. Update next suggested bid

6. If nobody accepts the suggested bid, then
   6.1. Reduce suggested next bid
   6.2. If accepted, resume normal sequence
   6.3. If not accepted, reduce suggested next bid
   6.4. If accepted, resume normal sequence
   6.5. If not, begin close
   6.6. "Going once ...", if response, resume normal sequence, else
   6.7. "Going twice ..." if response, resume normal sequence, else
   6.8. Done. Display closing screen
7. Settle with winning bidder, two models
   7.1. Connect buyer to seller for direct settlement
   7.2. Collect money from buyer, deduct fee, convey amount to seller

<u>Variant (b): batch (i.e. time limited) auction</u>

Conventional on-line batch (time limited) auctions are common. E-bay is the most prominent example. This process description continues from step 4 of the English auction description as the startup of the concurrent and batch auctions are the same.

5. Conduct auction: Until closing time for an item:
   5.1. On entry to system display the following for the potential buyer:
      5.1.1. Latest listing
      5.1.2. Categories
      5.1.3. Search screen
   5.2. On selection of categories:
      5.2.1. Execute dill down
      5.2.2. Retrieve count of items that meet criteria
      5.2.3. If more count is less than 25 (or other small number (n) consistent with the layout of the screen) retrieve all items that meet criterion
      5.2.4. If count is more than n, retrieve n auctions with nearest expiration time
      5.2.5. Display link list to all items in list, sort order should be auction with nearest deadline to most distant

5.2.5.1. Item name

5.2.5.2. Time till end of auction

5.2.5.3. Highest current bid

5.2.6. On user selection of the item, display same information as above plus

5.2.6.1. Description

5.2.6.2. Photo (if any)

5.2.6.3. Attachments (if any)

5.2.7. If count is more than n, display further drill-down options as well as item information above

5.3. Accept new bid through the display screen

5.3.1. Log bids in order, reject if bid is not higher than last high bid by increment.

5.3.2. If bid is rejected, tell bidder that their bid is not sufficient

5.3.3. Update database recording highest bid, bidder, time of bid

5.3.4. Display screen to user to confirm that their bid is the highest

6. When the time limit is reached, determine if a new bid has been received in the last 3 minutes (or other short time period). If so, extent the bidding time by 3 minutes (or other short time period) and execute step 5 with a new closing time.

7. When the time limit is reached, including all extensions under step 6, then

7.1. Email message to highest bidder that they won

7.2. Add transaction to completed deals

7.3. Update splash and add screens

7.4. Settle with winning bidder-- two models:

7.4.1. Connect buyer to seller for direct settlement

7.4.2. Collect money from buyer, deduct fee, convey amount to seller

C. Dutch Auction Transaction Engine

A Dutch auction, like a standard auction, involves the sale of a single item or batch with fixed specifications. There is one seller, and many potential buyers. The seller sets the prices, ideally higher than any buyer's maximum bid price. The offered price is reduced by a fixed increment at fixed intervals until a buyer accepts the price. The purchase goes to the first buyer in to accept the price. In the physical world (as opposed to the online world), Dutch auctions are rarely if ever run concurrently. In a live trading room, it could be difficult to determine which buyers was first to commit to a price when several are willing to pay the same amount. The Dutch auction is relatively simple to implement in an electronic environment. There are, at present, no online Ducth Auctions of which the inventors are aware.

1. Enter and maintain a framework for catalog
    1.1. Enter / delete / edit categories. Categories are titles for groups of items, such as "furniture" or "solvents"
    1.2. Enter / delete / edit subcategories. Subcategories are categories within categories, effectively establishing a hierarchy of products. Example: furniture/dining room/tables.
    1.3. Create groups of categories and subcategories (e.g. see also....). The grouping allows a person browsing items to be referred to another category that may contains items of interest. For example, someone may reach the furniture/dining room/tables and then be referred to furniture/office/conference room tables where other suitable tables may be listed, or to furniture/dining room/chairs to buy chairs that make the table. This cross referencing makes transforms the hierarchical arrangement of categories into a web.
2. Execute qualification process (optional)
    2.1. Admit bidders who are qualified based on past participation
    2.2. Provide fill-in-the blank qualification form new bidders
    2.3. Collect information
    2.4. Conduct automated review or manual review
    2.5. Inform prospective bidder of qualification or not
3. Collect information on items to be auctioned and owners, including
    3.1. Identity of seller
    3.2. Descriptions, optionally including attachments and photographs, independent certifications or appraisals, or other information necessary to establish the value of the tiem
    3.3. Categorization
    3.4. Starting price
    3.5. Increment, Interval for reduction
    3.6. Minimum price
    3.7. Obtain consent to rules (possibly as part of registration/qualification process)

3.8. Collect to conduct auction if item is

3.9. Calculate time to take item off auction by determining the number of steps (intervals) necessary to reduce price from the starting price to the minimum

3.10. Record all of the above information in the Dutch auction database

4. Cull expired options

4.1. Search database periodically for items where current time is later than time to take item off auction (2.9)

4.2. Inform owner that item was not sold

4.3. Delete entry from database

4.4. Prompt for revised terms start of another auction, create new entry if user takes option

5. When the buyer enters the system display a list of high level categories, a prompt for search criteria, and/or a link to a search page. Allow user to drill down through categories or enter search parameters.

5.1. Retrieve count of items that meet criteria

5.2. If more count is less than 25 (or other small number (n) consistent with the layout of the screen) retrieve all items that meet criterion

5.3. If count is more than n, retrieve n auctions with nearest expiration time

5.4. Display link list to all items in list, sort order should be auction with nearest deadline to most distant

5.4.1. Item name

5.4.2. Time till end of auction

5.4.3. Current price:

5.4.3.1. Retrieve starting price (SP) and increment (I$)

5.4.3.2. Calculate number of intervals since start of auction (INT)

5.4.3.3. Determine price = SP – (INT * $)

5.5. On click, display same information as above plus

5.6. Description

5.7. Photo (if any)

5.8. Attachments (if any)

5.9. The display screen should include a button that allows the buyer to purchase the item at the selected price.

6. When the user clicks the "buy" button
   6.1. Email message to highest bidder that they won
   6.2. Add transaction to completed deals database
   6.3. Settle with winning bidder-- two models:
      6.3.1. Connect buyer to seller for direct settlement
      6.3.2. Collect money from buyer, deduct fee if any for auction and payment services, convey the remainder to seller.

D. Reverse English Auction Transaction Engine

In a reverse auction, there are multiple buyers to one seller. Prices come down rather than up. There are many variants of a reverse auction. The variant discussed here is a reverse English auction. Reverse auctions have been implemented on line in Open Markets.

The process for posting an item for bid and for qualifying bidders is the same as for other auctions. The difference here is that the buyer may optionally set a maximum price.

1. Accessing the list of items sought
   Potential bidders access items sought by working through a hierarchy of categories and subcategories or entering search criteria, as for other auctions. A list of items within the category/subcategory and/or meeting the search criteria is displayed. The user may then
   1.1. Terminate the session on finding no suitable items
   1.2. Revise the search criteria
   1.3. Select an item on which to bid
2. If the user selects an item on which that may wish to bid, detailed information about the items is displayed. This item may include the following information:
   2.1. Name
   2.2. Seller
   2.3. Description
   2.4. Detailed specifications for items
   2.5. Delivery requirements
   2.6. Proposed terms
   2.7. Current low bid
3. If the user determines that they should bid, he accesses the bid entry screen from

the detailed description in Step 2 above. Making a bid consists of entering the following information:

3.1. New, lower bid

3.2. Comments pertaining to any special terms, features, or conditions

3.3. Attachments containing relevant additional information and any certifications required by the buyer

4. On receipt of bid, there are two options – either all bids are accepted, or bids are accepted only after review of information by the buyer.

4.1. Case 1: all bids are accepted

4.1.1. New bid is checked to determine if it is lower than prior bid

4.1.2. If so, then

4.1.2.1. bidder is notified that their bid is currently the lowest

4.1.2.2. seller is notified of new low bid

4.1.2.3. bid database is updated

4.1.3. If not, then

4.1.3.1. Bidder is notified that their bid is not the lowest

4.1.3.2. Bid screen is displayed so that bidder may lower bid

4.2. Case 2: bids are accepted after review by buyer

4.2.1. Buyer is notified of bid via email or online message

4.2.2. Buyer accesses complete information on the proposed bid through the system

4.2.3. Buyer select accept bid or reject bid.

4.2.4. If bid is accepted, then

4.2.4.1. Bidder is notified that their bid is currently the lowest

4.2.4.2. Bid database is updated

4.2.5. If bid is not accepted, then

4.2.5.1. Buyer enters reason for not accepting bid

4.2.5.2. Bidder is informed that bid is rejected with reason stated above

4.2.5.3. Bidder may access the bid screen to revise offer

5. When time period has expired and there have been no bids within a short specified interval, then

5.1. If at least one bid less than the maximum has been received, then:

5.1.1. Notify low bidder that their offer was successful

5.1.2. Add transaction to completed deals database

5.1.3. Settle with winning bidder-- two models:

5.1.3.1. Connect or introduce buyer to seller for direct settlement

5.1.3.2. Collect money from buyer, deduct fee if any for auction and payment services, and convey the remainder to seller.

5.2. If no bid less than the maximum has been received, the

5.2.1. Notify buyer

5.2.2. Allow buyer to revise bid criteria

E. Sealed Bid Transaction Engine

In a sealed bid system, the buyer publishes or distributes detailed, fixed specification to a number of potential bidders (who may or may not be prequalified). Bidders submit binding bids by a specified deadline, in a specific format that allows ready comparison. The competitive bidding process is distinguished from the bid and proposal process by the complexity of the specifications and the bids. In a simple competitive bid, competition among the bidders is along one or two readily quantified dimensions (always including price) and there is little or no room for variation in the form or specifications of the offering. Comparison of the bids is elementary.

The process for posting an item for bid and for qualifying bidders is the same as for other transactions as is the method to identify items on which to bid either using the hierarchy of categories and subcategories or a search engine.

1. If the user selects an item on which he may wish to bid, detailed information about the items is displayed. This item may include the following information:

1.1. Name

1.2. Seller

1.3. Description

1.4. Detailed specifications for items including all information necessary to prepare a bid

1.5. Bid instruction including specification for any documentation the buyer may required with a bid (e.g. proof of bonding or license)

1.6. Notice of any fees for bid registration

1.7. Delivery requirements

1.8. Proposed terms

2. After review of the bid requirements, the user may choose not to bid or may enter a bid. The process for entering a bid consists of preparing a bid package, including the price offered and any necessary supporting documentation. This is done by completing an online form, with provision for attachments. The bid is submitted through the system where it goes into a database of bids that are not opened to the closing time for the bidding process.

3. At the closing time, all bid packages are conveyed to the buyer.

   3.1. If there are no bids, the buyer is offered the opportunity to revise the request for bids.

   3.2. If there are multiple bids, the buyer reviews the bids and selects the lowest priced qualifying bid. They buyer informs the seller and arranges payment and delivery in accord with the terms stated in the bid package.

F. Order Matching Transaction Engine

In an order-matching system there are many potential buyers. Each posts binding offer to buy (bid amount) or sell (asked amount). The process proceeds in real time. The order matching system constantly compares bid and asked and, when a match is found within a specified spread, the deal is concluded. No accepted offer can be repudiated, but offers may be withdrawn before a deal is consummated. The strike price is posted so that buyers and sellers can modify their offerings in real time. The items traded are fungible so that price is the only decision. For the market to operate efficiently the items traded must be tightly defined and the terms of sale must be fixed and determined in advance. This is typically done by the operation or an exchange, with the order-matching engine operating in the background. To insure that the items traded are well defined, and the terms of sale are rigid example of an order matching process in stock trading on an exchange.

Users of an order-matching engine are all potential buyers and seller. They are qualified in advance using a process like that outlined by for auction with the extension that deposit accounts are frequently required given the speed of transactions in exchange environments.

1. Establish and maintain items to be traded. All functions in this category are reserved to the manager of the exchange or a designee.

   To add (i.e. "list" and idem), enter

1.1. Unique item number or symbol

1.2. Description of item (e.g. Sears Class A Common Stock)

1.3. Terms and conditions ownership (e.g. who can own) if any

1.4. Trading units (e.g. shares, blocks, etc.)

1.5. Additional information as required by the rules of the exchange

To delete (i.e. "delist" and item)

1.6. Select the item to be deleted

1.7. Confirm deletion

2. On entry to the system, potential buyers and sellers can review the price of the last transaction of any item, either through a list or a search by item name or symbol. The current highest asked and lowest bid price are also shown.
3. An offer to sell is posted by entering the following information:

   3.1. Item number or symbol

   3.2. Quantity offered

   3.3. Proposed price ("asked")

   3.4. Seller

   3.5. Offers may be revise at any time prior to consummation of a deal
4. An offer to buy is posted by entering the following information

   4.1. Item number or symbol

   4.2. Quantity offered

   4.3. Proposed price ("asked")

   4.4. Buyer

   4.5. Offers may be revised at any time prior to consummation of a deal
5. Offers to buy and sell are constantly reviewed by the software. When there is an offer to buy and sell at a price within a preset difference. When prices match, buyers and sellers are notified of the transaction, and the transaction is recorded. The display of the last transaction price, the highest bid and the lowest asked price is updated.
6. The transaction is conveyed to the backend accounting system of the exchange.

G. Bid and Proposal

The bid and proposal process is typically used for procurement of large or complex products or services, in which cost is not the only factor. Cost must be weighed against the buyer's assessment of the quality and suitability of an offering and the ability of the bidder to deliver the product or perform the specified services.

The bid and proposal process is conducted between one buyer (possibly representing a consortium) and many potential sellers, sometimes organized into teams. The buyer issues specifications that may be general or highly specific, brief or very lengthy. The specifications may be distributed freely or to a list of qualified buyers.

With physical RFPs, the size and the associated cost of distribution make it common practice to advertise the availability of the RFP first, sending copies only to those that request it. Frequently, the requestors are required to supply information to establish their qualifications to bid. While cost is not an issue in electronic dissemination of RFPs, the model of advertising prior to distribution is still useful in managing the qualification process. This is addressed as variant (a) is this description. Variant (b) requires no prequalification.

In a competitive bid on fixed requirements (sealed bid or auction), there is typically very little communication between buyer and seller between publication of the request and submission of the bids. The requirements are comparatively simple, clear, and unambiguous. In contrast, the bid and proposal process may involve considerable communication between buyer and seller. The process may begin with a bidders' conference to answer questions about the requirements. Additional questions from bidders may be accepted, though not all need be answered. Questions and answers may be made available to all bidders or the response may be in private. This dialog is crucial for two reasons. First, it helps the bidders understand the requirements and to be responsive in their bids. Second, it is not unusual for the bidders' questions to identify some point of ambiguity, error, or contradiction in the specifications, leading to a modification of the RFP. The diverse perspectives of the bidders, and the close attention required on their part to prepare a bid inherently provides an excellent review of the RFP.

The initial phase of the RFP process concludes with submission of the bids, but this is far from the conclusion of the process. Commonly, questions arise from the review of the proposals. These may relate to a specific submission or have broader implications, leading to modification of the requirements. The list of bidders can be culled to the best candidates. These are asked to answer questions about their proposals and to provide additional and clarifying information.

The process described here is built around the document repository described elsewhere in this application. Through this process of refinement, the list of bidders is narrowed to one or two with whom a contract is negotiated. The process of negotiation is addressed as a separate transaction type (Negotiation Engine) as it may be conducted without the bid and proposal process.

Variant (A): with pre-qualification

1. Software supports the user in creating a web site for the proposal process. Initially this site manages the process for requesting the request for proposal (RFP), qualifying bidders, and disseminating the RFP.
2. Supported by the system software, the bidder creates and RFP advertisement by
   2.1. entering a summary of the RFP.
   2.2. entering a summary of the information needed to qualify as a bidder or
   2.3. attaching a form (HTML web page or template for paper form) for entering qualifying information
3. The RFP advertisement includes file transfer software for uploading qualifying information to the repository.
4. Disseminate RFP advertising
   4.1. Post on public bulletin board or
   4.2. Disseminate via mail to selected users
5. When users access the system, issue them an encryption key and PIN to be used for subsequent uploads and communications to verify their identity.
6. Receive requests for RFP in repository
   6.1. Prompt for key
   6.2. Encrypt submission
   6.3. Upload
   6.4. Generate receipt – should include an authentication number
7. Disseminate RFP to selected user, either:
   7.1. Attach to return Email or
   7.2. Post the RFP in a repository from which qualified prospective bidders may download the file. If the repository model is used, provide notice of the posting via email including any necessary PINs and codes to access the repository
   7.3. When a prospective bidder downloads an RFP, issue an encryption key to be

used in submitting proposal

8. The RFP site also includes a page through which prospective bidders can submit questions. Questions and answers are posted to the site.
9. Updates to the schedule and amendments to the RFP are posted to the site
10. All access to the site is recorded to verify that prospective bidders have received critical information. Direct contact may be used when it is determined that a bidder had not accesses the site since critical new information was posted.
11. Bidders prepare their proposal and then upload them to a repository for proposals using software built into the proposal site.
    11.1. Prompt for key
    11.2. Encrypt submission
    11.3. Upload
    11.4. Generate secure hash number to prevent tampering with the submission
    11.5. Generate receipt including secure hash number and authentication code
12. After initial proposals are received, the process moves into a phase commonly termed the "best and final process" in which the proposals are reviewed, the list narrowed, and the proposals refined.
    12.1. Create separate secure environment (i.e. web site with repository) for each respondent
    12.2. Exchange materials through repository (described elsewhere in this filing)
    12.3. Records and receipt each access
    12.4. Generate key for revised proposal
    12.5. Receive proposal using process in 11
    12.6. Repeat from step 11 as many times as necessary

The remainder of the process is completed as a negotiated deal, described below.

<u>Variant B: no pre-qualification:</u>

Proceed as above, beginning with Step 6 and not requiring a key for download of the RFP.

H. Negotiation Deal Engine

An engine for negotiating a deal can be built around the capability of the system to create a temporary virtual private network through the web. A temporary network is created for the negotiation. Access to the network is limited to the parties of the negotiation, their advisors and counsel, and, potentially, arbitrators and regulators. The members of the negotiating environment have access to the complete set of tools described in this filing including those for communications (email, anonymous mail, online chat, threaded dialogs, and audio and video collaboration), the library of standard contract instruments, the tools for document signature and authentication, and the document repository. Using these tools in a secure environment they can negotiate, close, and register a deal.

FIG. 19 shows one possible user interface for selecting participation engines to be made available to group members. The term "participation engine" refers generally to collaboration tools that provide features beyond merely communicating among group members. Various services such as an on-line survey tool, a DELPHI model tool; brain writing tool; and real-time polling can be provided.

A. Online Survey

In online polling or surveying, the person creating the poll uses and automated tool (new to this application) to build simultaneously an online questionnaire and a database to collect the results. The user builds the questionnaire by entering a series of questions and an associated data collection widget for each. The polling tool builds the database and the data entry screen. The data entry screen consists of two columns. The left column is a series of questions. The right column is the data entry tool appropriate to the question. Various data entry tools can be provided to respond to the query, including such things as:

1. yes / no radio buttons
2. true / false radio buttons
3. slider with scale from 1-5, 1-10, etc.
4. fill-in-the-blank text box
5. numeric field
6. multiple check boxes (e.g. strongly disagree, disagree, agree, strongly agree)

Other data entry types may be added.

As each question / data collection widget is added, the polling tool creates the database. The database includes one record per data collection form. Creating the database structure simply means adding one new field to each record definition for each question. The type of data collection widget defines the format of the field, as follows:

1. yes / no radio buttons: one character field, limited to "y" or "n"
2. true / false radio buttons: one character field, limited to "y" or "n"
3. slider: real number field, with appropriate range check
4. fill-in-the-blank text box: text box
5. numeric field: real number or integer
6. multiple check boxes: integer field with range check from 1 to number of boxes

Every data entry screen provides a "save" and "cancel" button. Save writes to the database. Cancel exits the entry screen without saving.

The survey, once composed as described above exists as a web page. This page can be embedded in web applications. It can be made available on a site available to the entire Internet, on an Intranet, or in a dynamically created environment. Alternatively, it can be distributed via e-mail. When the form is completed, the submit button transmits the value entered to the database that is created at the time the form is generated. Access to the database is controlled by the rules of the database system. It may be limited to the individual who creates the survey form and database, but it may be accessible other users in the survey developers organization, as determined by the database administrator. Distribution of the result of the analysis is at the discretion and control of the individual managing the survey. This manager may be the individual who creates the survey, but the actual creator may be acting on behalf of the survey manager. Results may be kept private, posted to the Internet, and intranet, or a collaborative environment, distributed via e-mail within an organization, or, if the information is available, sent via e-mail to the participants in the survey.

B. Online Delphi Engine

The online Delphi engine allows real-time collaboration in estimating or predicting an outcome that can be expressed numerically. For example, the method can be used to develop a consensus forecast of grain prices. The method has been in used since the 1970s, but has not previously been adapted to online processes. One possible method is as follows:

1. Establish the session
    1.1. Within an online community, the moderator of the session creates the brain writing session by entering the following information:
        1.1.1. Name of moderator
        1.1.2. Title of the session
        1.1.3. Description of the session
        1.1.4. Background reading as references or attachments
        1.1.5. Start date for the session
        1.1.6. Scheduled end for the session
        1.1.7. Access to the session:
            1.1.7.1. URL for access
            1.1.7.2. Open to all or invitees only for observation
            1.1.7.3. Open to all or invitees only for participation
        1.1.8. Payment information if required
2. Optionally, the session may be advertised on line
3. If the session is private, invitations with logon keys must be distributed via email, actual mail, or download.
4. Optionally, the moderator may run on online applications and qualification process
5. Prior to the start of the session, the moderator must describe precisely the value to be estimated. The definition must be completely unambiguous.
6. Each participant connects at the start of the session. On connecting, they question is posed (e.g. "What will be the price of West Texas intermediate oil in December?")
7. Each participant enters a number a brief (1 paragraph maximum) explanation of their reasoning.
8. When the participant is done entering their estimate, they click "Done".
9. Each participant's estimate and explanation is recorded.
10. Each participant then sees the summary screen.

11. Estimates are arrayed graphically from top to bottom of the screen, from lowest to highest. The value is stated as is the associated comment, but the source of the comment is not revealed.
12. Participants can review the estimates and comments, send an anonymous message to the author or any comment, or amend their answers.
13. The session terminates when the time expires, or when the moderator determines that there it is no longer appropriate to continue. The operator may determine this is based on declining participation or, if participation is high, the moderator may extend the deadline.
14. Participants and observers may access the final display of estimates, again arrayed from top to bottom, lowest to highest.

C. Brain Writing

Brain writing is a variant of a method for facilitated group discussion termed brainstorming. The objective of brainstorming is to maintain the focus of the discussion while encouraging creative input and recognizing the contributions of all members of the group. It seeks to avoid problems with a few individuals dominating the discussion, with junior staff deferring to senior staff, and with new ideas being abandoned before than can be developed fully. Brain storming has been commonly used since the late 1960s. Brain writing is a more intense method that relies on joint writing rather than discussion. What is presented here is adaptation of that method to an online environment. It is believed to be the first such adaptation.

1. Establish the session
   1.1. Within an online community, the moderator of the session creates the brain writing session by entering the following information:
      1.1.1. Name of moderator
      1.1.2. Title of the session
      1.1.3. Description of the session
      1.1.4. Background reading as references or attachments
      1.1.5. Start date for the session
      1.1.6. Scheduled end for the session
      1.1.7. Access to the session:
         1.1.7.1. URL for access
         1.1.7.2. Open to all or invitees only for observation

1.1.7.3. Open to all or invitees only for participation

1.1.8. Payment information if required

2. Optionally, the session may be advertised on line
3. If the session is private, invitations with logon keys must be distributed via email, actual mail, or download.
4. Optionally, the moderator may run on online applications and qualification process
5. Prior to the start of the session, the moderator must list some number (typically 5-10) of questions or hypotheses to be explored. (e.g. " Our company should create a spinoff to develop and commercialize the new breast cancer vaccine") This may be done by the moderator alone, in consultation with the participants, or with other outside the session.
6. Each question or hypothesis becomes a "Card".
7. Participants may enter the session any time after the start. A password may be required if the session is not open.
8. On entry into the system, a user if given a card at random. The card consists of the initial question or hypothesis plus all comments entered on the card by other participants.
9. After reviewing the card, the participant may add his or her own comments to the bottom. After entering comments, the participant clicks "Done" to return the card to the pile.
10. When a participant returns a card to the pile, they received another card, chosen at random (preferably) or selected by the user. This process continues until the opt to exit. They may reenter at any time up to the conclusion of the session.
11. When a card is returned to the pile, it is become available for assignment to the next participant. The card includes the additions of the most recent participant.
12. A participant may opt to return the card without addition if he or she has nothing to add.
13. Participants may create new cards when new ideas come to mind. These are treated in exactly the same way as original cards.
14. Observers may view any card but may not add to them.
15. The moderator may limit participation to a set number at any time so that there is a sufficient number of cards to keep the participants fully occupied.

16. The session terminates when the time expires, or when the moderator determines. that there it is no longer appropriate to continue. The operator can determine this based on declining participation or, if participation is high, the moderator may extend the deadline.
17. The raw cards are distributed at the conclusion to all participants. The moderator or another individual is charged preparing a summary and arranging follow-up.

FIG. 22 shows one possible scheme for storing brain card writing data elements. In accordance with one embodiment, each brain writing card comprises a data structure including the following elements:

1. Brain writing session number: Serially assigned number to differentiate brainwriting sessions. A session is the set of all cards pertaining to a particular topic.
2. Card number: A Serially assigned sequence number
3. Initial Comment : The question or comment used to initiate the discussion (e.g. "SAIC should purchase a company that produces Internet server software"
4. Date and time card started
5. Date and time card closed
6. Comments: A collection (i.e. a set of unlimited length) containing the comments added by participants in the brainwriting session..
7. Date of additional comment: Date and time that each additional comment was added.
8. Commenter: Name or user ID of the person adding each additional comment. Ideally, brainwriting should be anonymous to encourage open dialog. Accordingly, this field may be omitted from an implementation. Some organizations, however, may wish to track this information without making it visible to users, or in some cases to attribute comments.

When the user has finished defining the group and specifying its functions, environment generator 1201a (FIG. 12) creates an environment accessible to the group members and including the functions specified during the environment definition process. As shown in FIG. 20A, for example, a web page can be created for the newly created environment, including those functions that were selected by the user that created the group. All group members are notified of the existence and location of the environment, and each group member can use the functions provided in the environment to collaborate on a project or conduct business.

FIG. 20B shows what an environment might look like to a group member after entering the environment. As shown in FIG. 20B, for example, a news banner announces the latest news for the group. Additionally, specific communication tools, research tools, transaction engines, and participation engines are made available to group members, which can be executed by appropriate mouse clicks in accordance with the inventive principles. According to various inventive principles, each tool shown on the web page is accessible through a hyperlink to a web-based program that performs predefined functions as set forth above. For example, clicking on "online catalog" would link the group member to a web page that implements an online ordering engine as described previously. Users can navigate through the various tools using conventional web browser features (i.e., forward, backward, etc.). It may be desirable to implement some or all of such software using server-side scripting or other similar means consistent with the system configuration of FIG. 12.

FIG. 21 shows how environment generator 1201a can create multiple environments including virtual private facilities, which can be implemented through web pages that contain hyperlinks to functions available to members of each group or environment. An environment definition software component 2106 implements steps 1101 through 1103 of FIG. 11 in order to create one or more environments 2107. (In one embodiment, each group can also be provided with a copy of an environment generator 2106 in order to create sub-groups that draw on the applications and directory structure created for the group). As a user identifies group members and selects functions to be provided for the environment in which the group will collaborate, environment definition component 2106 stores information relating to the selected members and functions in databases. Each environment can include a web page (not shown in FIG. 21) and directories, tools and other applications specific for each created group.

Based on user selections of the type illustrated in FIGS. 13 through 19, environment generator 2106 creates an environment 2107 containing one or more web pages with links to the selected tools. Environment generator 2106 retrieves information from various information sources including a directory of communication tools 2101 (e.g., including descriptions of tools and URL/IP addresses of web applications to set up each communication tool); directory of transaction engines 2102 (e.g., including descriptions of transaction engines and the URL/IP addresses of web-based applications to set up each transaction engine); directory of research tools 2103 (similar to above); list of global data objects 2104 (e.g., a dictionary of data elements from which the directory of each group can be composed); and a directory of applications 2105 (e.g., a description of available applications and URL/IP addresses of pages to set up access to applications).

WE CLAIM:

1. A method of negotiating a deal over a network of computers, the network including at least one or more computers connected to the Internet, the method comprising the steps of:

(1) posting, on an electronic list that can be viewed over the Internet, information regarding one or more offers to form a contract;

(2) posting on the electronic list one or more responses to the one or more offers;

(3) researching the one or more responses to determine whether they satisfy one or more contract criteria;

(4) negotiating over the network between at least two parties to accept or modify one or more of the responses; and

(5) electronically signing a document to consummate the contract.

2. The method of claim 1, wherein step (1) comprises the step of displaying offers and responses in a parent-daughter spatial relationship on a computer display.

3. The method of claim 1, further comprising the step of sorting the one or more offers and one or more responses according to a user-selected sort order.

4. The method of claim 1, wherein steps (1) and (2) are done anonymously, such that each party to the contract cannot determine the identity of the other party to the contract.

5. The method of claim 4, further comprising the step of simultaneous revealing the identity of each party prior to step (5).

6. The method of claim 4, wherein steps (1) and (4) comprise the step of sharing a single anonymous e-mail alias among a plurality of users.

7. The method of claim 1, further comprising the steps of:

(6) registering keywords with an electronic agent that monitors the one or more offers and providing an e-mail address to be notified upon a keyword match; and

(7) in response to the electronic agent detecting the keyword match, transmitting a message to the e-mail address provided in step (6).

8. The method of claim 1, wherein step (2) comprises the step of clicking on a hyperlink linking the information posted in step (1) to a reply card.

9. The method of claim 7, wherein step (2) comprises the step of requiring

the submission of certain information before the reply card will be accepted.

10. The method of claim 1, wherein steps (3) and (4) are performed a plurality of times for a single contract, such that modifications are made to the one or more responses.

11. The method of claim 1, further comprising the step of electronically registering a plurality of entities that have signatory authority and correlating the registered entities with one or more documents to which signatures can be affixed.

12. A method of displaying information on a computer display, comprising the steps of:

(1) displaying a first plurality of graphical objects each having a shape of a file folder comprising a folder face and a labeled tab, wherein the first plurality of graphical objects are stacked in a cascading arrangement; and

(2) in response to user activation of a "flip" tab, changing the graphical objects displayed in step (1) to show a second plurality of graphical objects each having a shape of a file folder comprising a folder face and a labeled tab,

wherein each of the first and second plurality of graphical objects can be brought to a foreground position in front of other graphical objects by clicking on a corresponding labeled tab.

13. The method of claim 12, wherein each of the first and second plurality of graphical objects has associated therewith one or more functions displayed on the folder face thereof, wherein user can activate the one or more functions by clicking thereon.

14. A method of creating a user-defined networked environment across a plurality of computers without requiring system administrator-level privileges, comprising the steps of:

(1) creating a group by providing a group identifier, a group description, and by specifying a plurality of group members entitled to use the user-defined networked environment;

(2) selecting a plurality of web-based communication, collaboration, and transaction tools from a list of available tools, wherein the selected tools are to be made available to the plurality of group members specified in claim 1; and

(3) through the use of computer software, automatically creating the user-defined networked environment by creating a web page accessible to the plurality

of group members selected in step (1), wherein the web page provides access to the plurality of tools selected in step (2).

15. The method of claim 14, wherein step (1) comprises the step of inviting a plurality of individuals to join the group by transmitting an invitation to prospective group members.

16. The method of claim 14, wherein step (1) comprises the step of advertising an invitation to join the group by posting an advertisement for prospective group members, wherein at least some of the prospective group members are unknown to the user creating the networked environment.

17. The method of claim 14, further comprising the step of screening prospective members that respond to the advertisement in order to determine whether they should be added to the group.

18. The method of claim 14, further comprising the steps of electronically collaborating among group members using the user-defined networked environment.

19. The method of claim 14, further comprising the step of destroying the user-defined networked environment when it is no longer needed.

20. The method of claim 14, wherein step (2) comprises the step of selecting a transaction engine that implements an auction to members of the group.

21. The method of claim 14, wherein step (2) comprises the step of selecting a transaction engine that implements an on-line electronic survey comprising survey questions that are to be answered electronically by survey participants.

22. The method of claim 14, wherein step (2) comprises the step of selecting a transaction engine that implements a bid-and-proposal tool that permits group members to electronically submit bids on one or more proposals.

23. The method of claim 14, wherein step (2) comprises the step of selecting an online ordering engine that permits group members to electronically order goods or services in the user-defined networked environment.

24. The method of claim 14, wherein step (2) comprises the step of selecting an Electronic Data Interchange (EDI) compatible interface that executes electronic commercial transactions between two or more group members.

25. The method of claim 14, wherein step (2) comprises the step of a selecting an electronic brain-writing tool that permits participants to brainstorm using electronic idea cards.

26. A system for implementing a user-defined networked environment that can be created without the need for system administrator-level privileges, comprising:

a plurality of networked computers that communicate using Internet Protocol;

a plurality of web browsers executing on the plurality of networked computers;

a database that stores information concerning the user-defined networked environment; and

a computer program executing on one or more of the plurality of networked computers, wherein the computer program performs the steps of:

(1) permitting a user to create a group comprising a plurality of group members;

(2) permitting the user to select a plurality of web-based communication, collaboration, and transaction tools from a list of available tools, wherein the selected tools are to be made available to the plurality of group members; and

(3) automatically generating a web page accessible to the plurality of group members, wherein the web page provides access to the plurality of tools selected in step (2) to the plurality of group members.

FIG. 1A

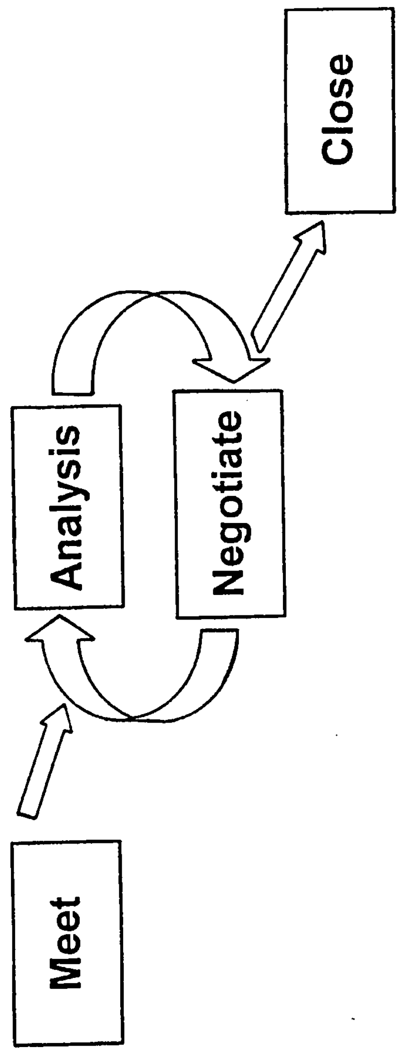


FIG. IB

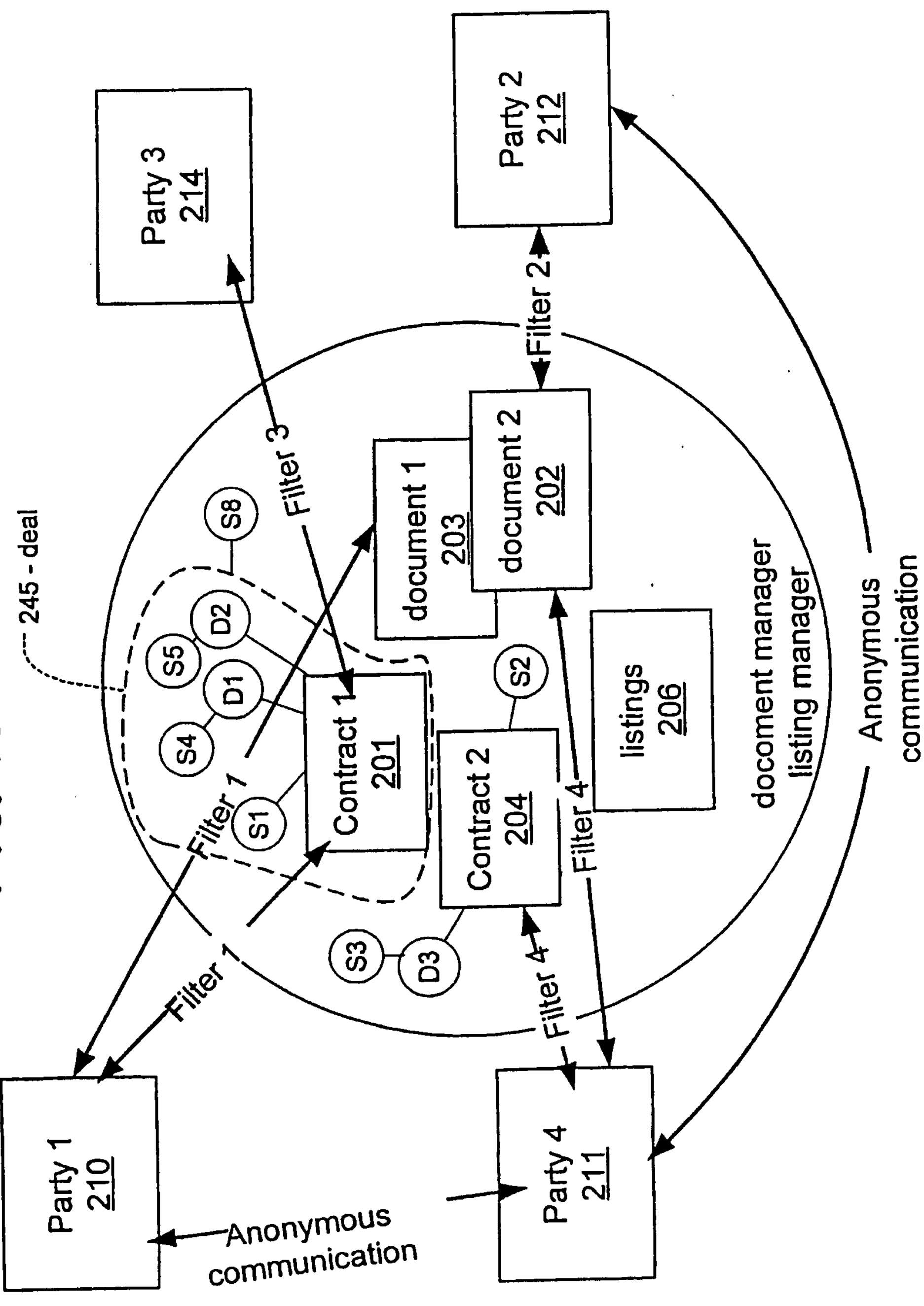
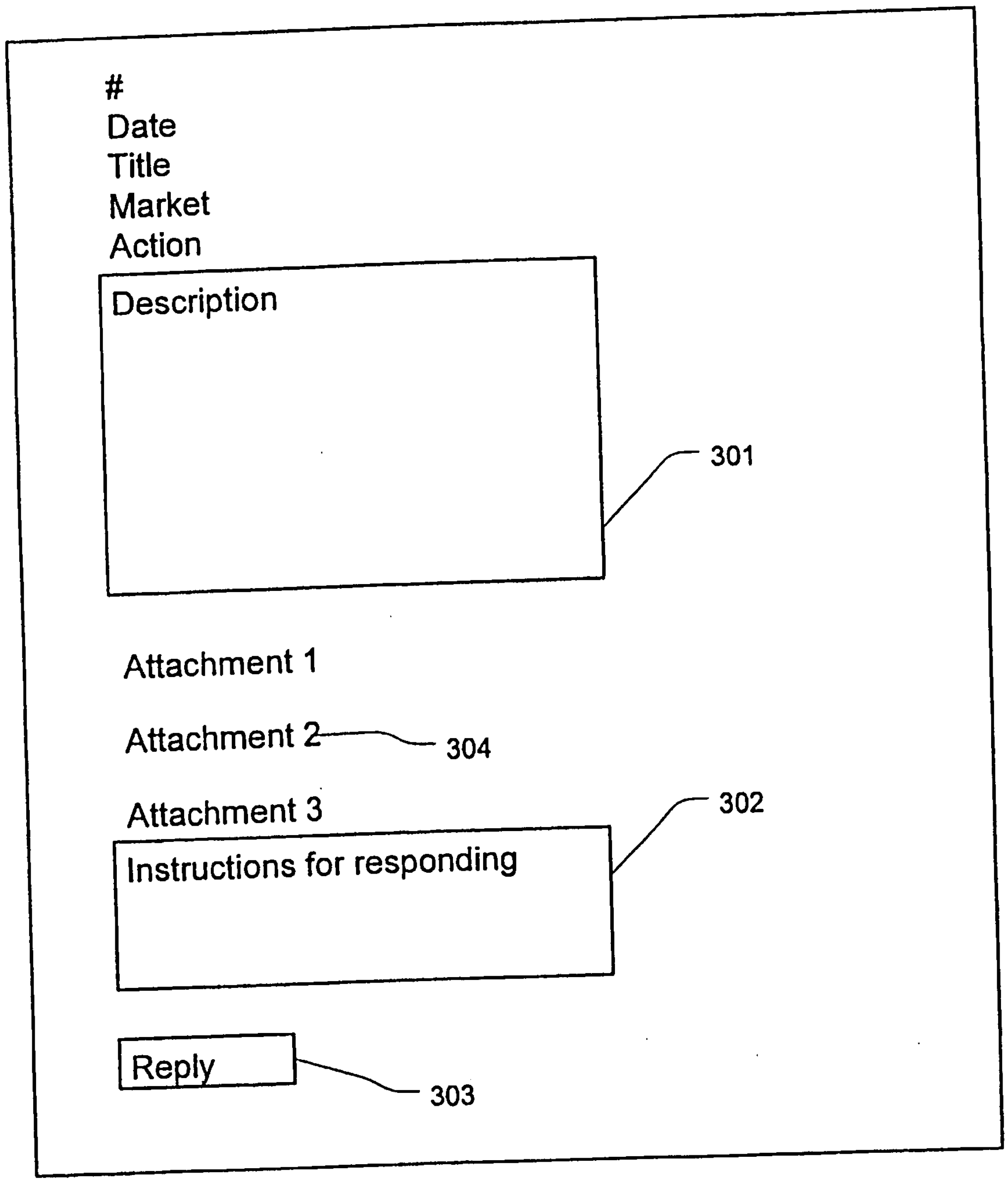

Party 1
210
Party 2
212
Party 3
214
Party 4
211
Anonymous communication
Anonymous communication
Filter 1
Filter 1
Filter 2
Filter 3
Filter 4
Filter 4
245 - deal
Contract 1
201
Contract 2
204
document 1
203
document 2
202
listings
206
docoment manager
listing manager
S1
S2
S3
S4
S5
S8
D1
D2
D3

FIG. 2

Number	Date	Market	Action	Title
123	7/10/87	Steel	Buy	Title
234	7/12/87	Alum.	Sell	Title
236	7/12/87			Title
239	7/10/88	Gold	Action	Title
240	8/12/87	Flood	Action	Title
243	9/01/87		Action	Title

315 314 313 312

FIG. 3


#
Date
Title
Market
Action
Description
301
Attachment 1
Attachment 2
304
Attachment 3
302
Instructions for responding
Reply
303

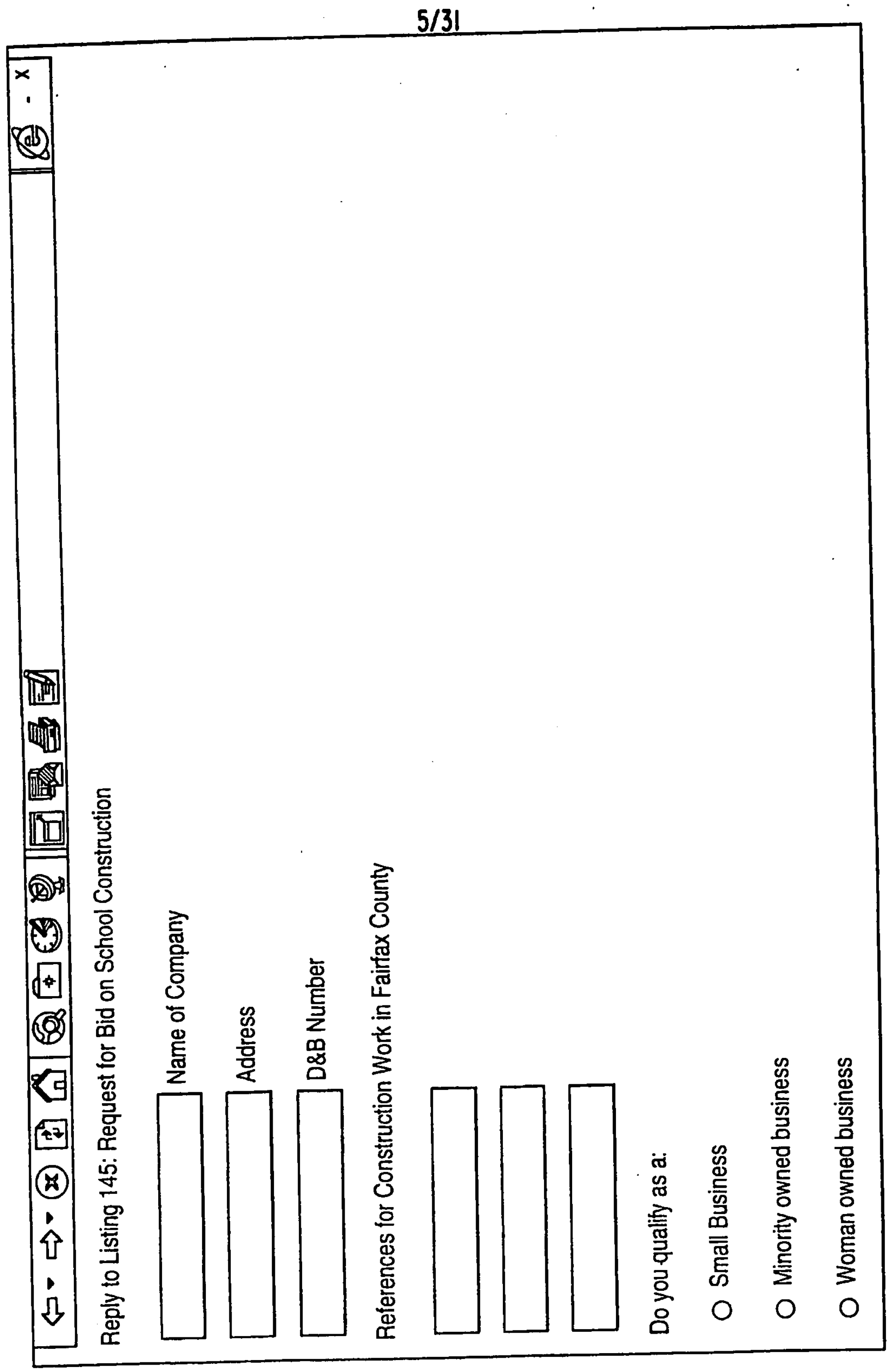

Reply to Listing 145: Request for Bid on School Construction
Name of Company
Address
D&B Number
References for Construction Work in Fairfax County
Do you qualify as a:
Small Business
Minority owned business
Woman owned business

FIG. 4

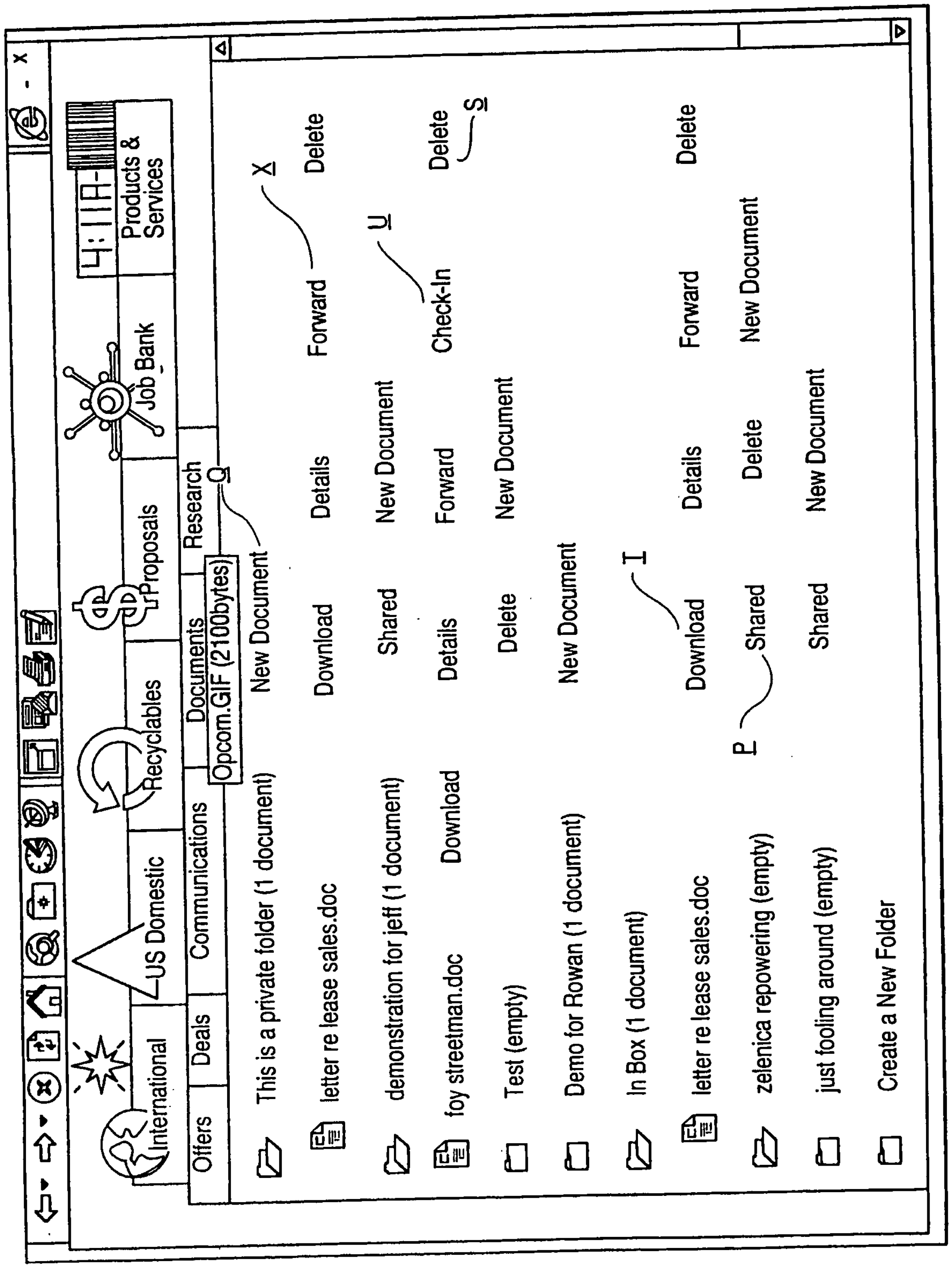


FIG. 5

SUBSTITUTE SHEET (RULE 26)

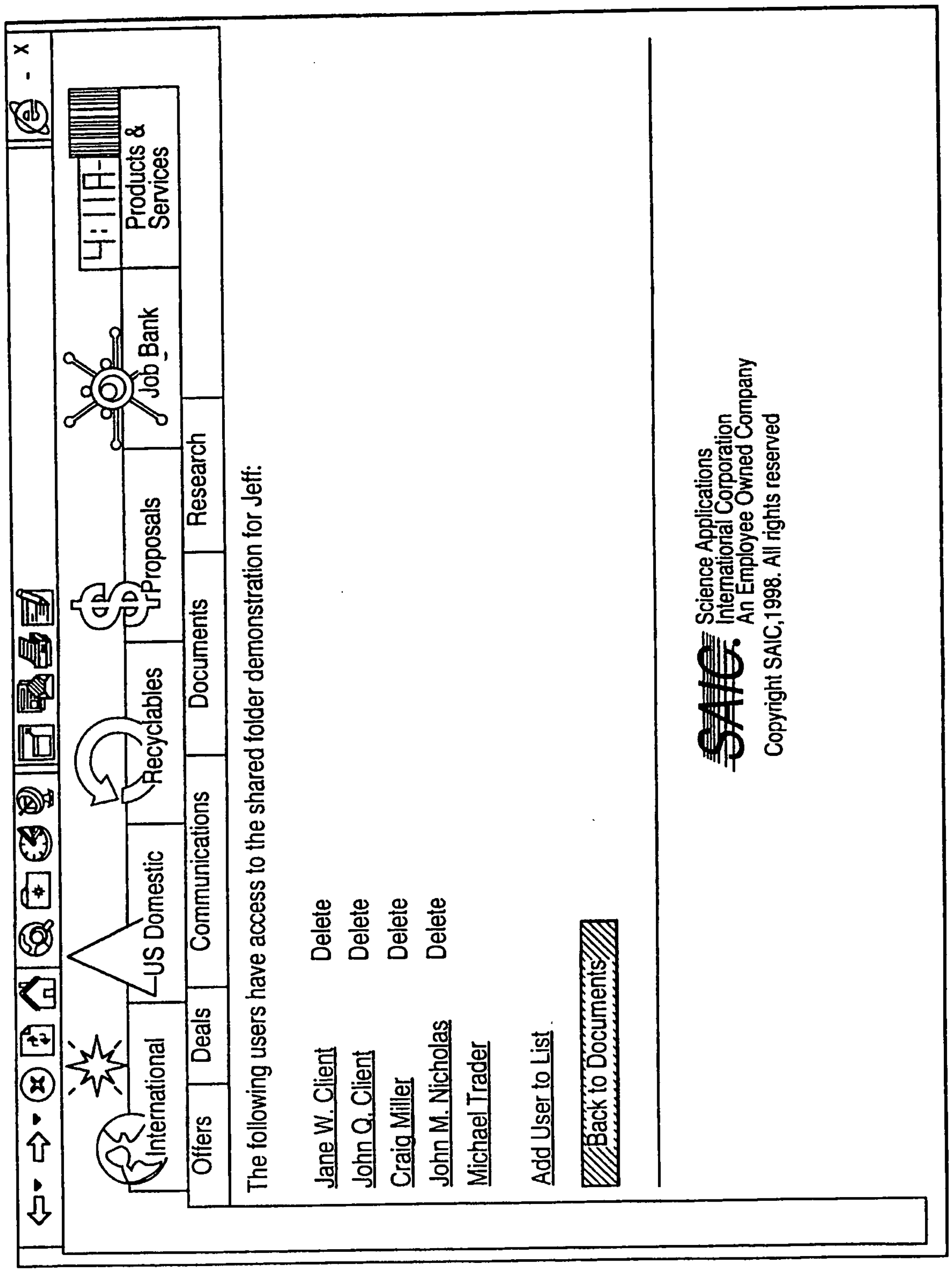

International
US Domestic
Recyclables
Proposals
Job_Bank
Products & Services
Offers
Deals
Communications
Documents
Research
The following users have access to the shared folder demonstration for Jeff:
Jane W. Client Delete
John Q. Client Delete
Craig Miller Delete
John M. Nicholas Delete
Michael Trader
Add User to List
Back to Documents
SAIC. Science Applications International Corporation An Employee Owned Company
Copyright SAIC,1998. All rights reserved

FIG. 6

FIG. 7

Trade Number	Date Created	Trade Title
357	Sep 22 1998	mmmmmmmmmmmm
356	Sep 22 1998	World Trade Center
353	Sep 22 1998	Florida Windstorm Coverage
352	Sep 22 1998	vvvv
342	Sep 1 1998	Bigger deal
341	Sep 1 1998	big deal
330	Aug 18 1998	Another trade
297	Jun 23 1998	xgssgs
295	Jun 23 1998	lkjdalskjd
293	Jun 23 1998	Test again
292	Jun 23 1998	Test of compiled tm
6	Feb 11 1998	Get in shape
5	Feb 10 1998	Big Deal
4	Feb 8 1998	Master Contract

SAIC. Science Applications International Corporation An Employee Owned Company

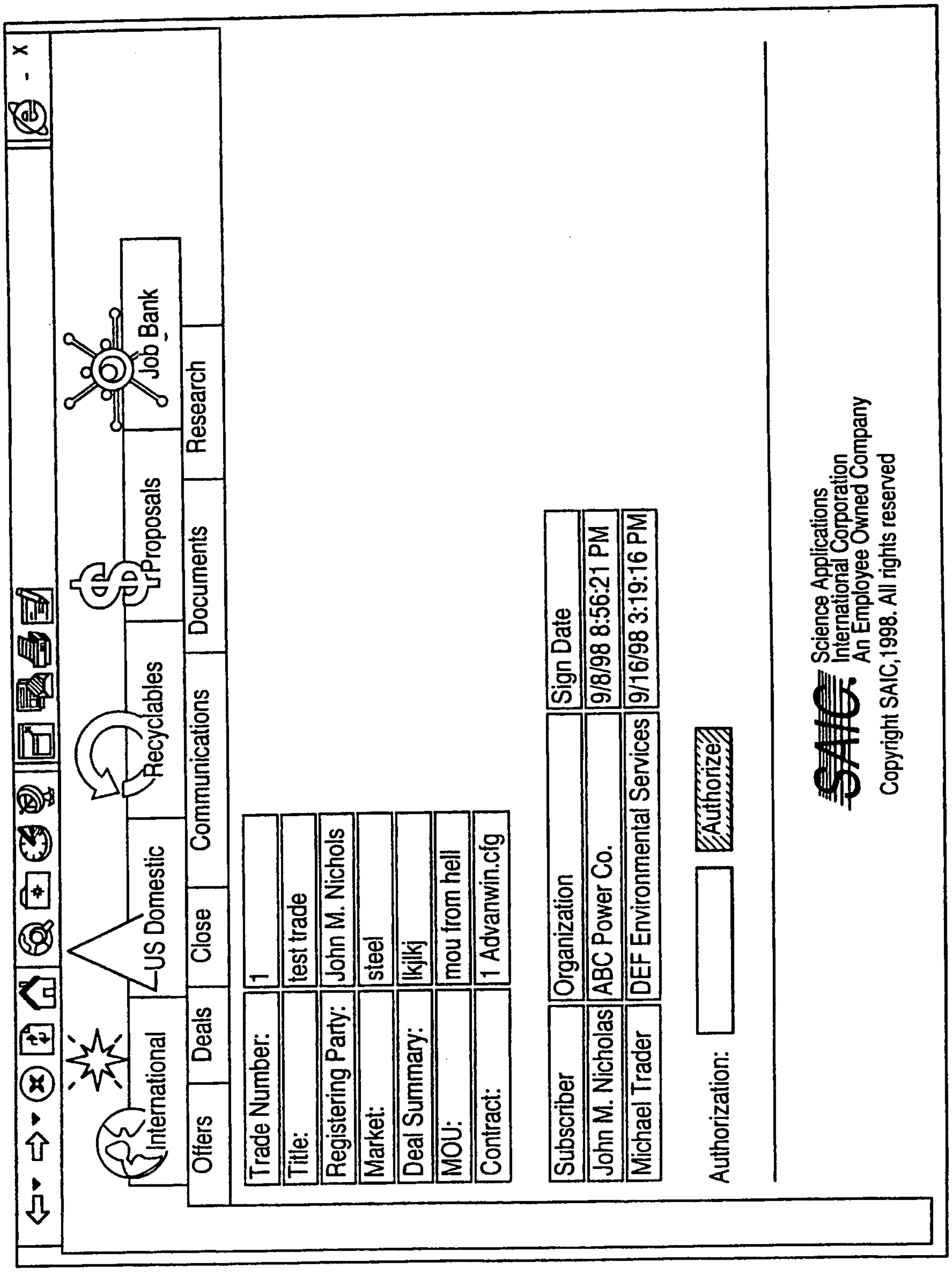

International
US Domestic
Recyclables
Proposals
Job_Bank
Offers
Deals
Close
Communications
Documents
Research
Trade Number:
1
Title:
test trade
Registering Party:
John M. Nichols
Market:
steel
Deal Summary:
lkjlkj
MOU:
mou from hell
Contract:
1 Advanwin.cfg
Subscriber
Organization
Sign Date
John M. Nicholas
ABC Power Co.
9/8/98 8:56:21 PM
Michael Trader
DEF Environmental Services
9/16/98 3:19:16 PM
Authorization:
Authorize
SAIC
Science Applications International Corporation
An Employee Owned Company
Copyright SAIC,1998. All rights reserved

FIG. 8A

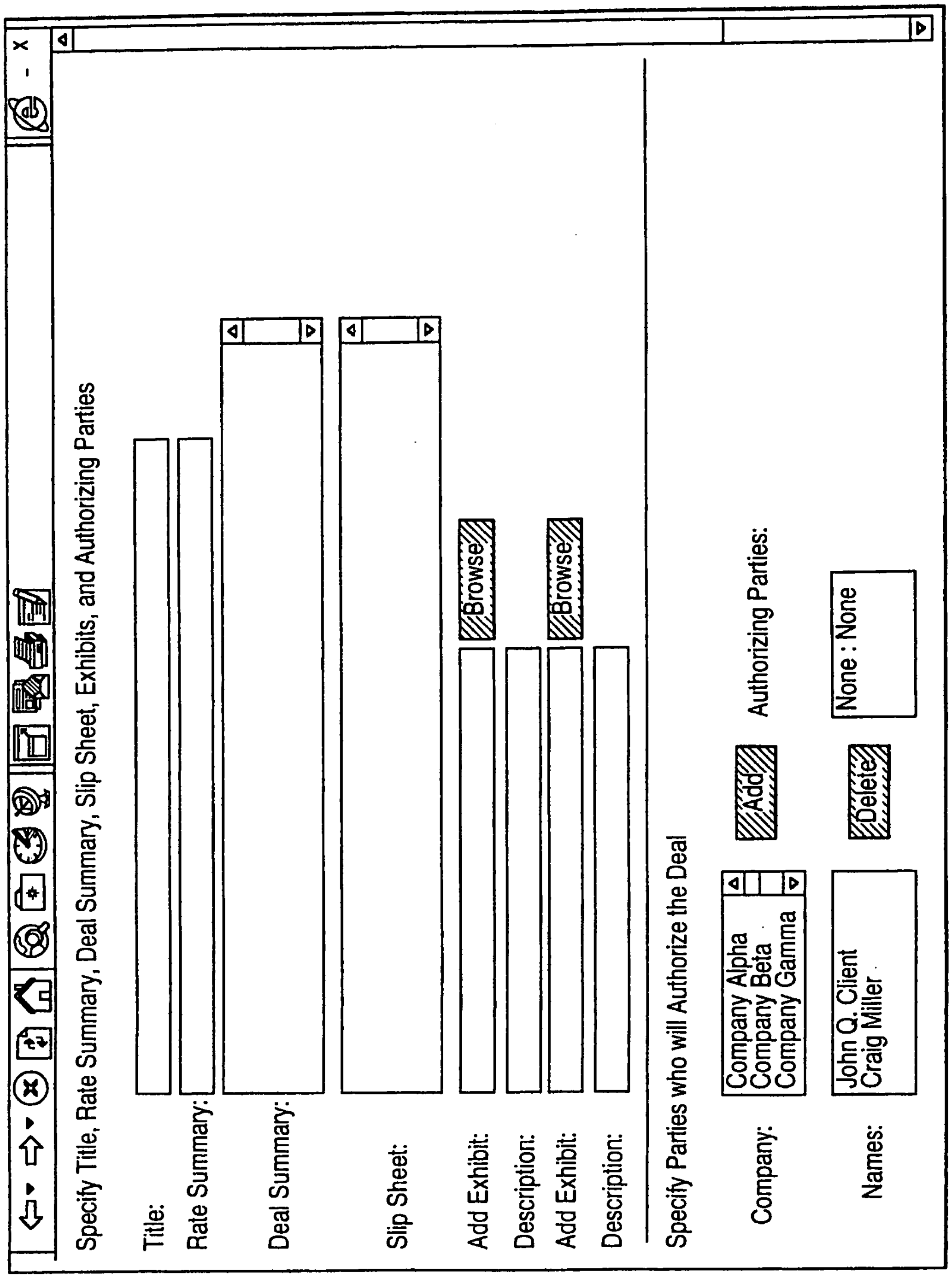

Specify Title, Rate Summary, Deal Summary, Slip Sheet, Exhibits, and Authorizing Parties
Title:
Rate Summary:
Deal Summary:
Slip Sheet:
Add Exhibit:
Browse
Description:
Add Exhibit:
Browse
Description:
Specify Parties who will Authorize the Deal
Company:
Company Alpha
Company Beta
Company Gamma
Add
Authorizing Parties:
None : None
Names:
John Q. Client
Craig Miller
Delete

FIG. 8B

FIG. 9

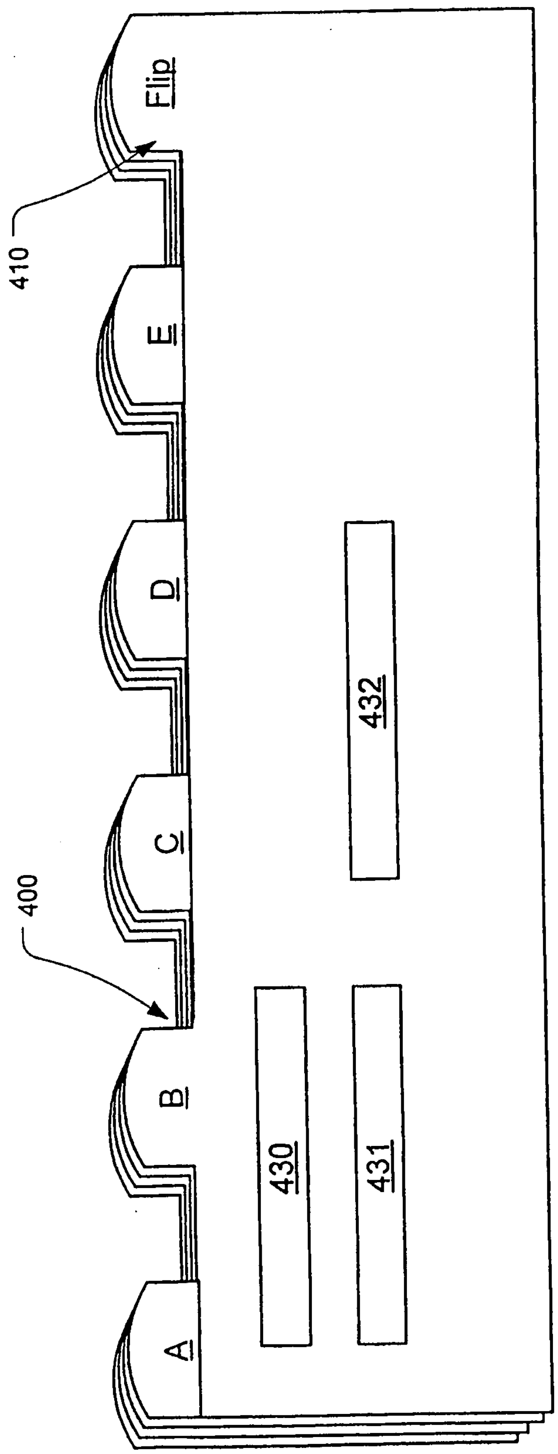
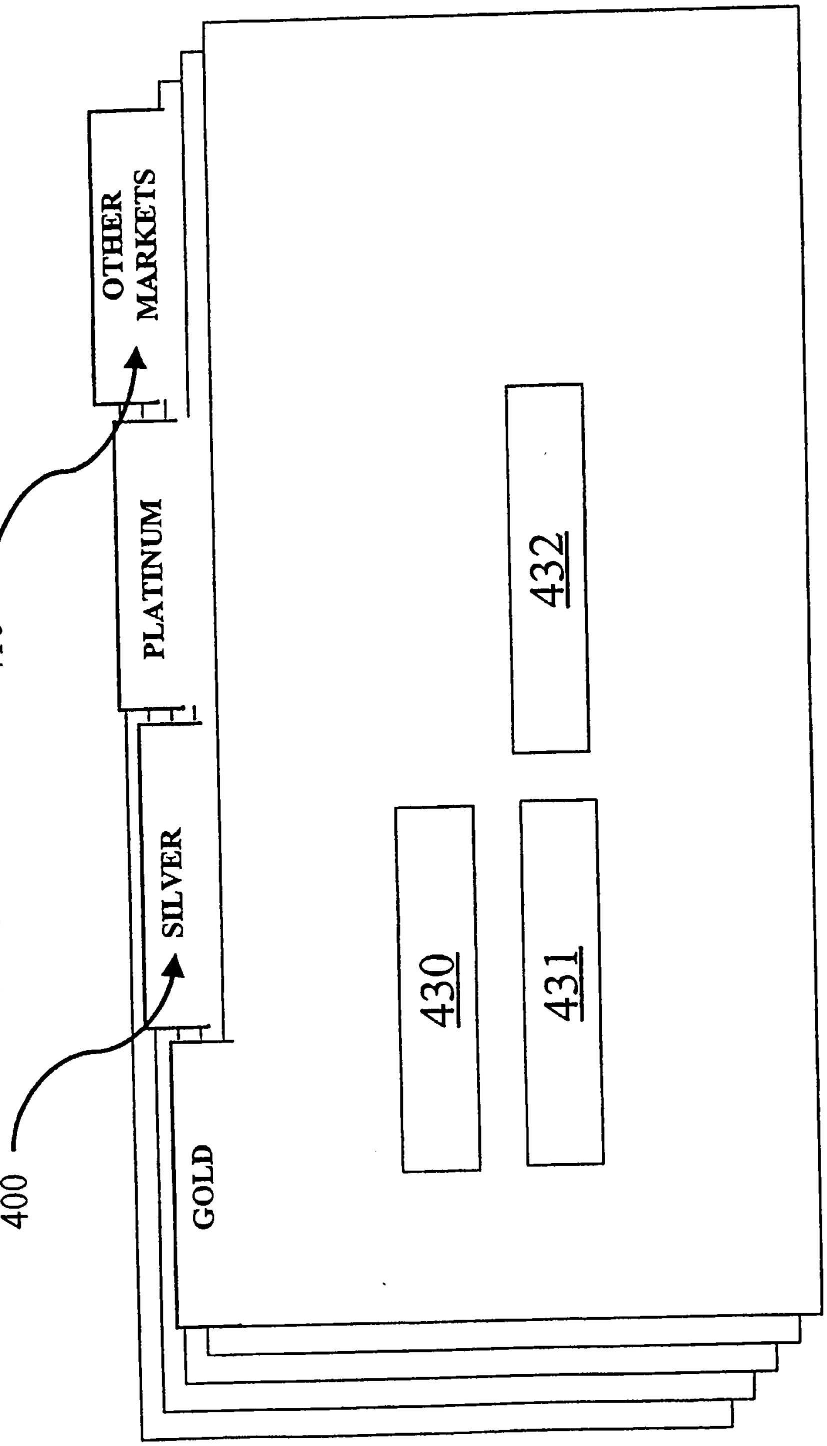


FIG. 9A



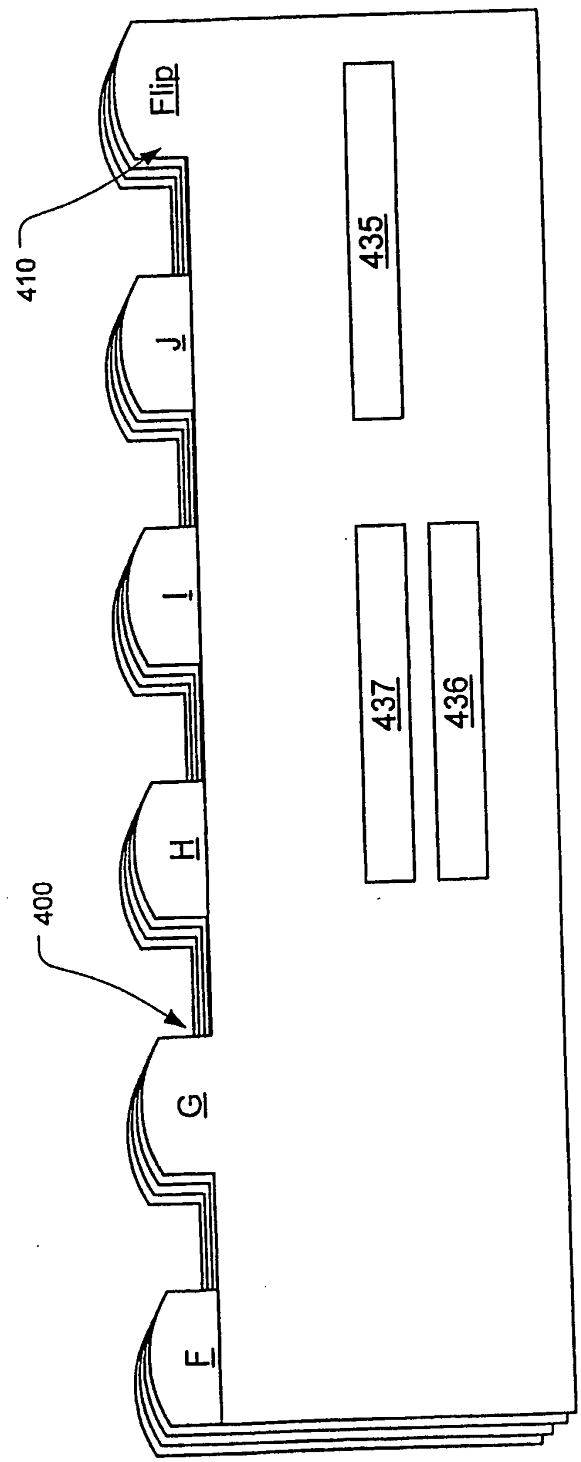
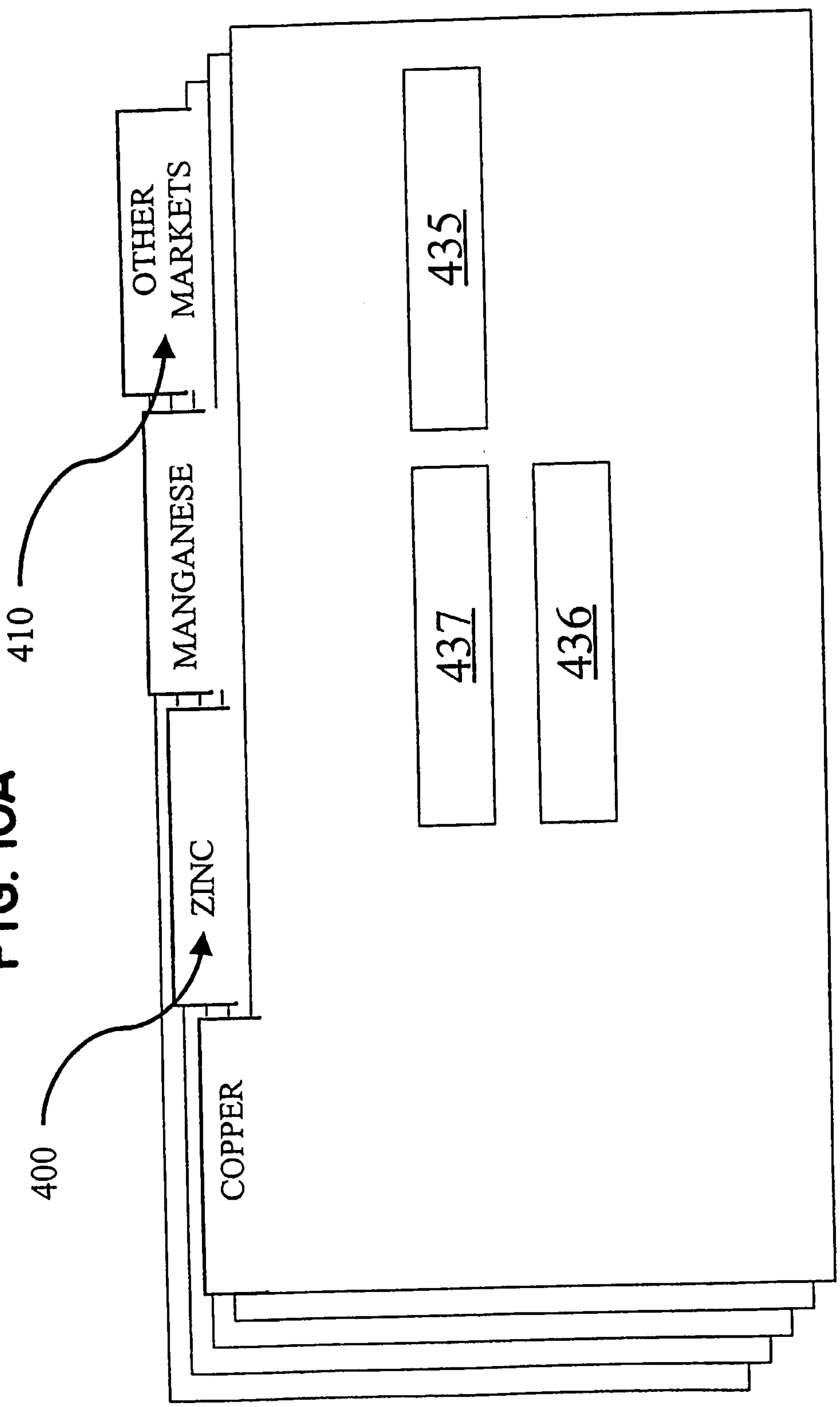


FIG. 10

SUBSTITUTE SHEET (RULE 26)

FIG. 10A



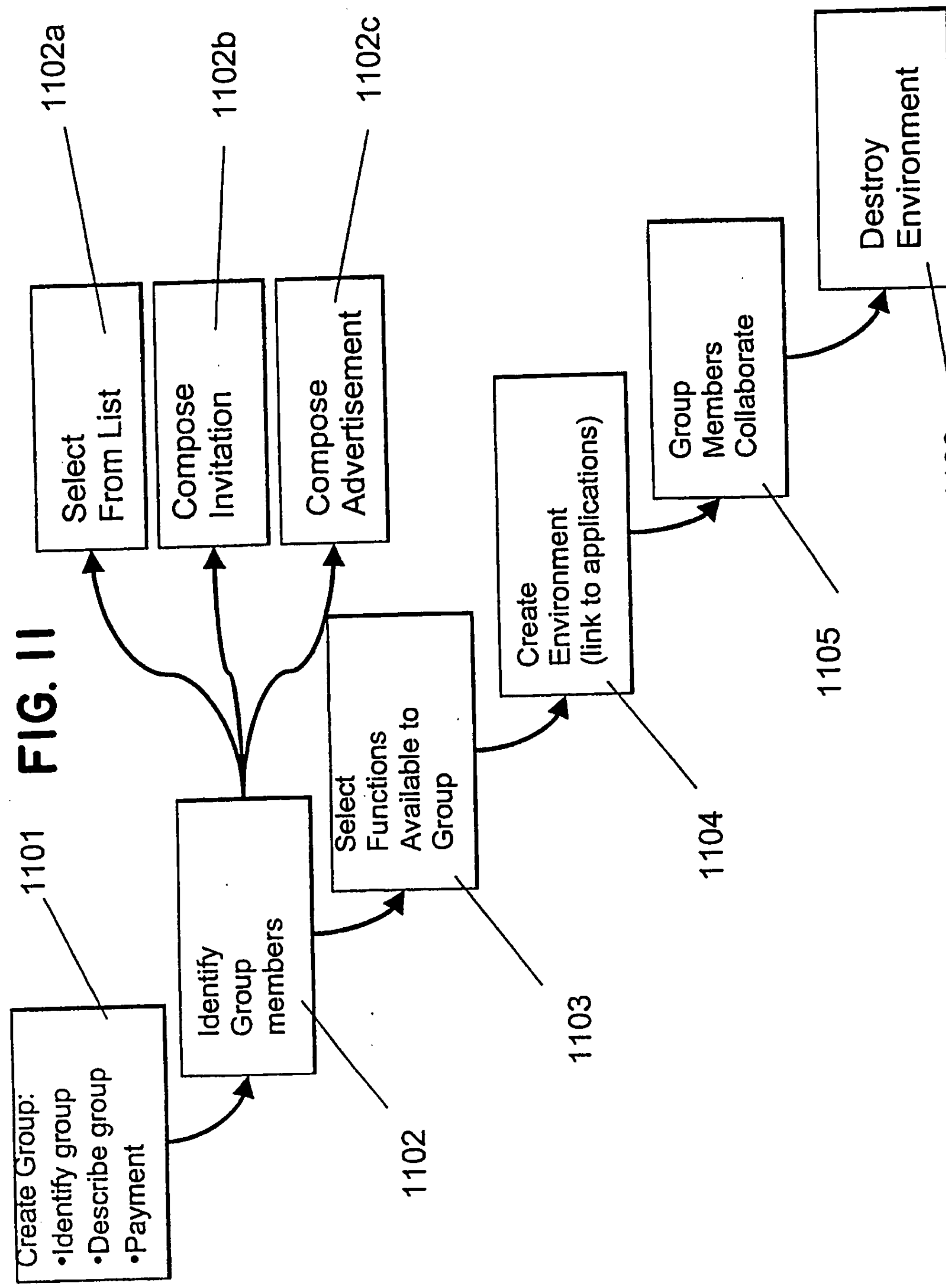

Create Group:
•Identify group
•Describe group
•Payment
1101
Identify Group members
1102
Select From List
1102a
Compose Invitation
1102b
Compose Advertisement
1102c
Select Functions Available to Group
1103
Create Environment (link to applications)
1104
Group Members Collaborate
1105
Destroy Environment
1106

FIG. 11

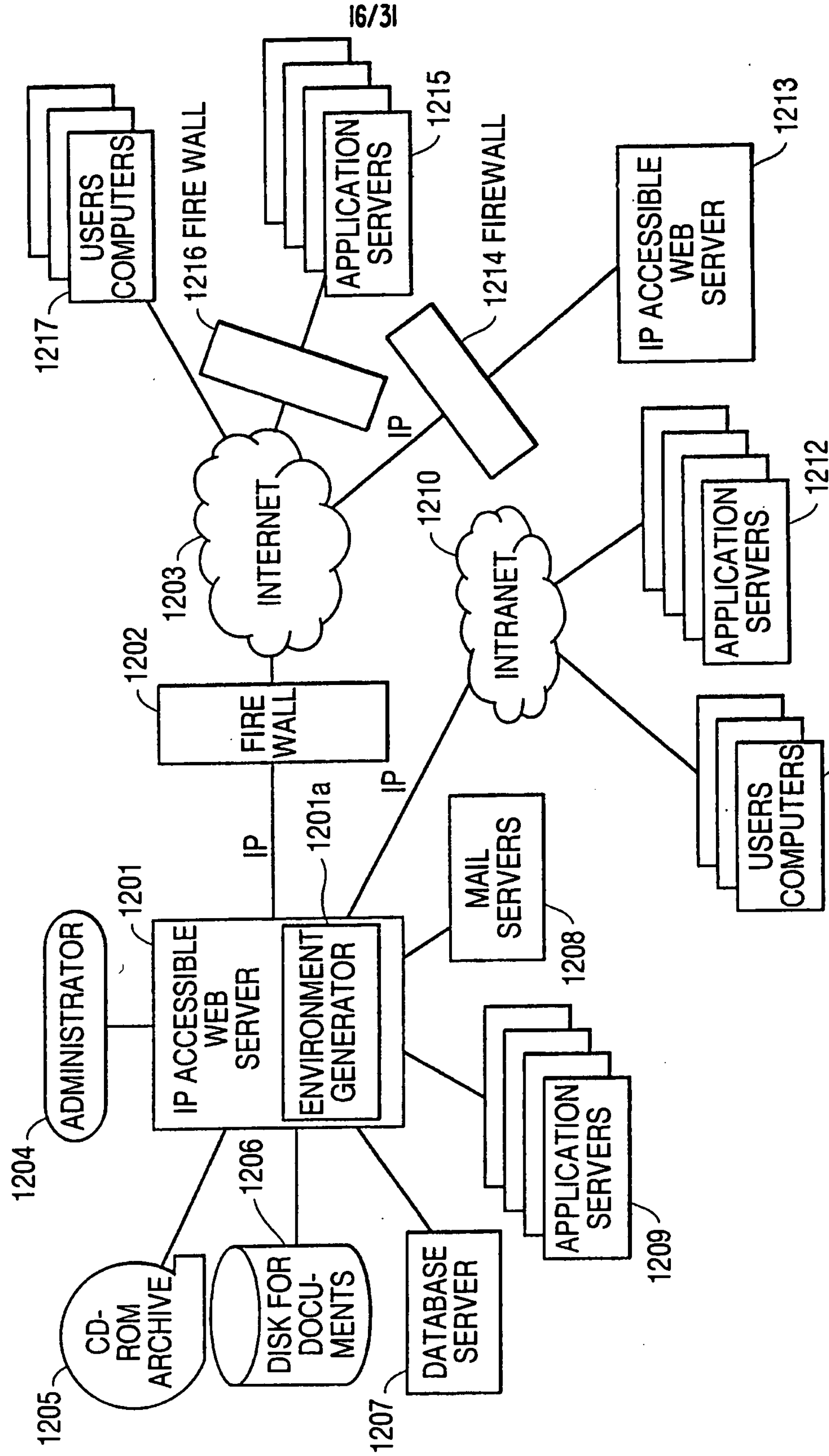

USERS COMPUTERS
1217
1216 FIRE WALL
APPLICATION SERVERS
1215
1214 FIREWALL
IP ACCESSIBLE WEB SERVER
1213
IP
INTERNET
1203
1210
INTRANET
APPLICATION SERVERS
1212
USERS COMPUTERS
1211
1202
FIRE WALL
IP
1201a
IP
MAIL SERVERS
1208
1201
ADMINISTRATOR
1204
IP ACCESSIBLE WEB SERVER
ENVIRONMENT GENERATOR
APPLICATION SERVERS
1209
1206
CD-ROM ARCHIVE
1205
DISK FOR DOCU-MENTS
DATABASE SERVER
1207

FIG. 12

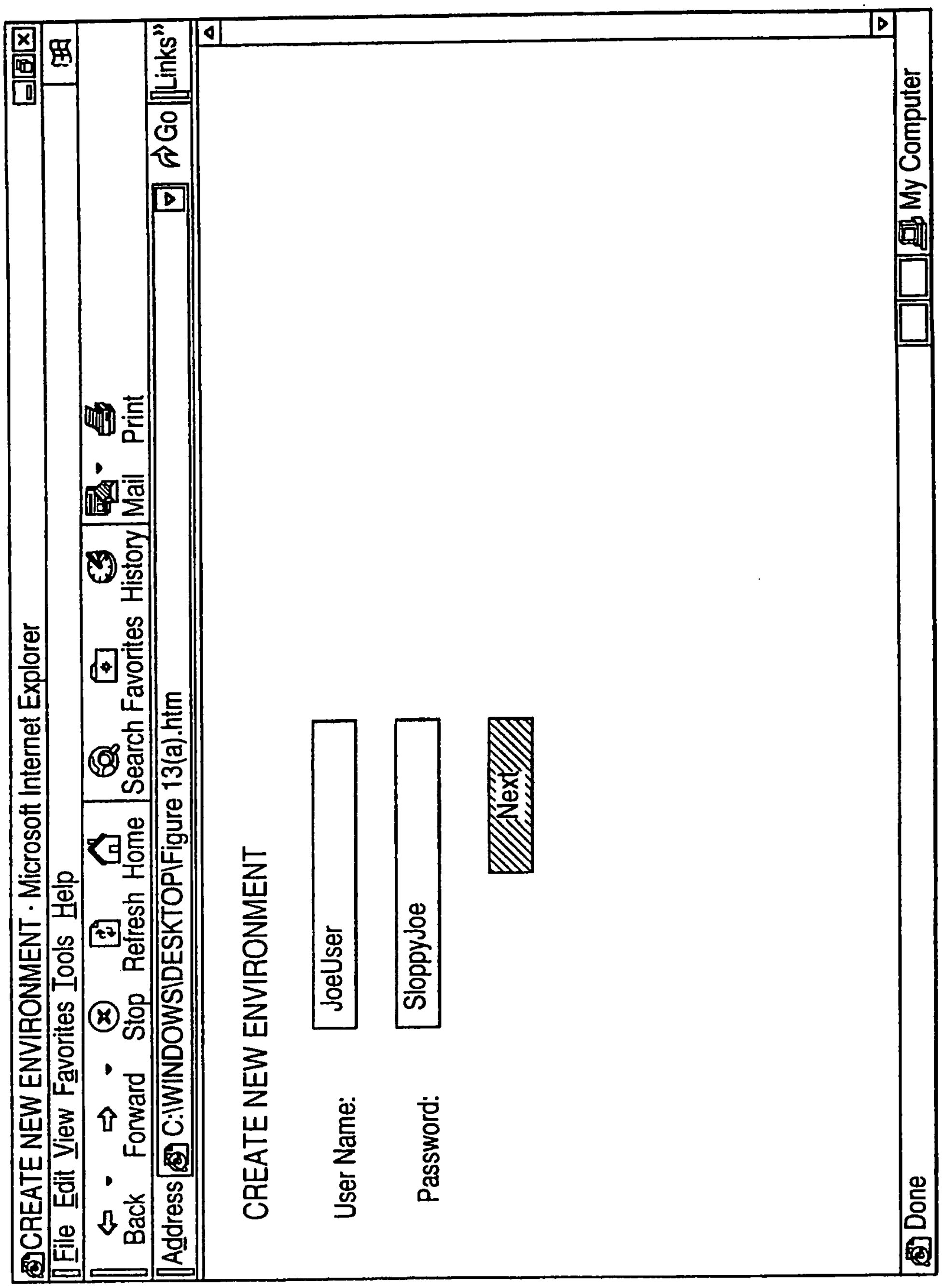

CREATE NEW ENVIRONMENT - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address C:\WINDOWS\DESKTOP\Figure 13(a).htm
Go Links"
CREATE NEW ENVIRONMENT
User Name:
JoeUser
Password:
SloppyJoe
Next
Done
My Computer

FIG. 13A

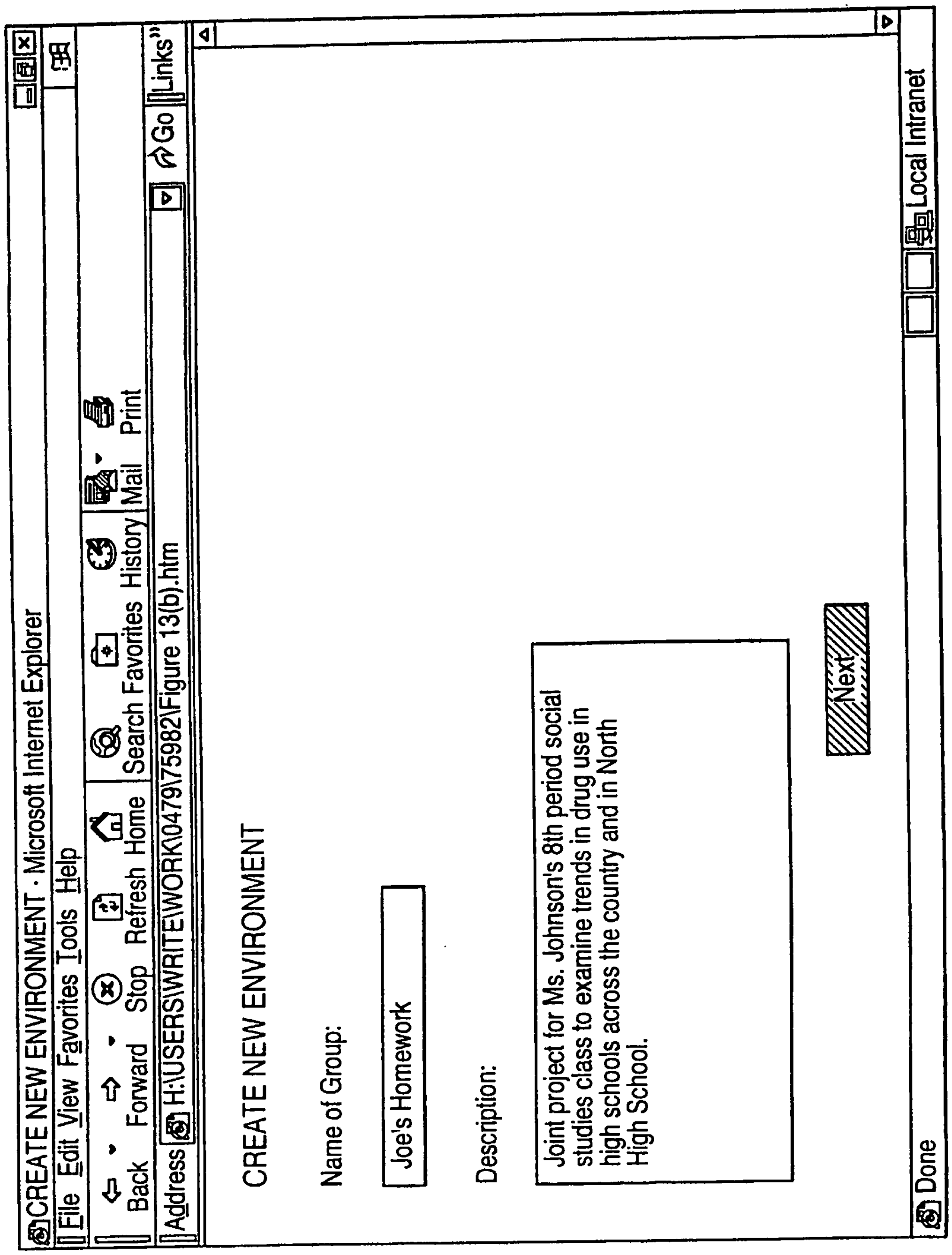

CREATE NEW ENVIRONMENT - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address H:\USERS\WRITEWORK\0479\75982\Figure 13(b).htm
Go
Links"
CREATE NEW ENVIRONMENT
Name of Group:
Joe's Homework
Description:
Joint project for Ms. Johnson's 8th period social studies class to examine trends in drug use in high schools across the country and in North High School.
Next
Done
Local Intranet

FIG. 13B

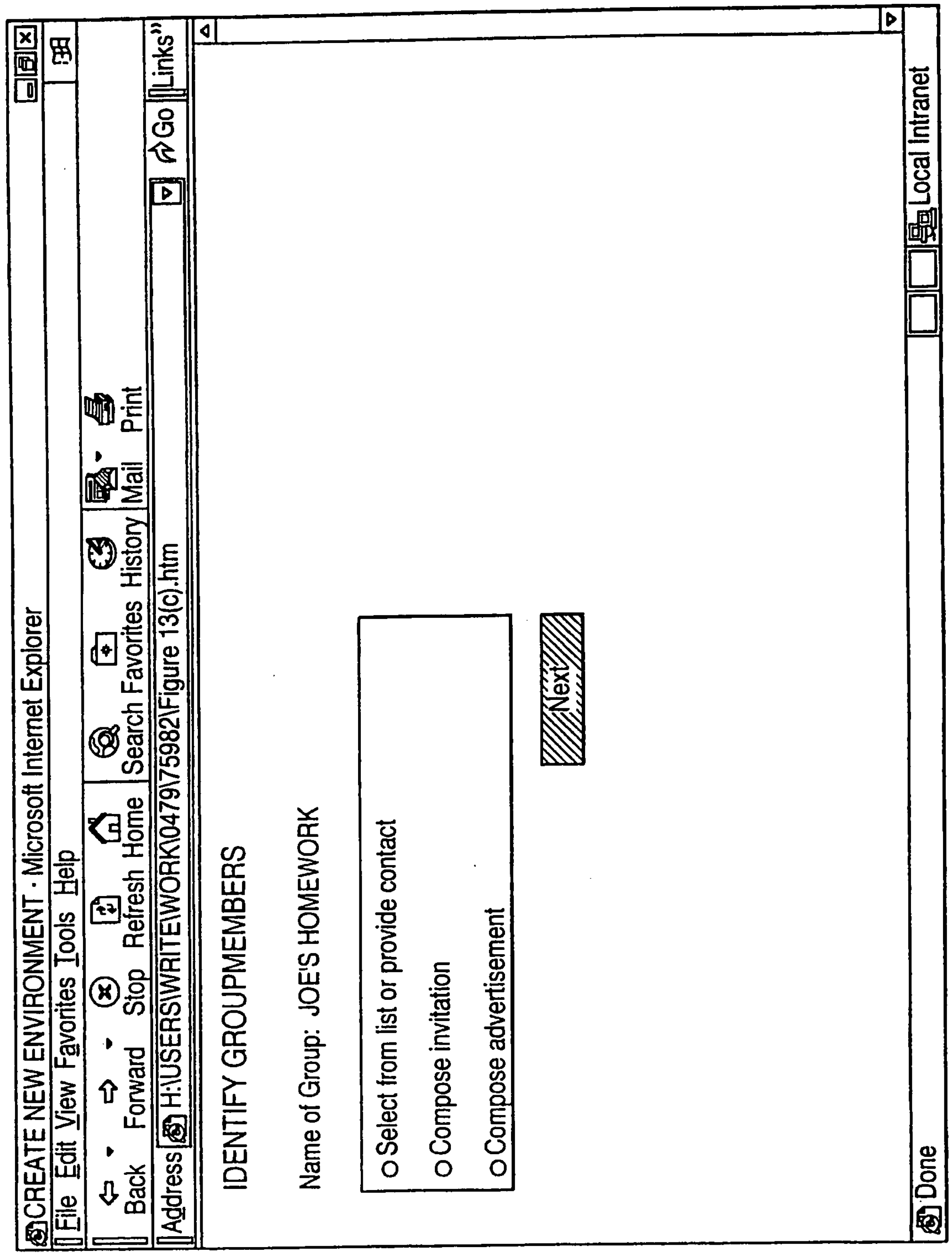

CREATE NEW ENVIRONMENT - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address H:\USERS\WRITEWORK\0479\75982\Figure 13(c).htm
Go
Links"
IDENTIFY GROUPMEMBERS
Name of Group: JOE'S HOMEWORK
Select from list or provide contact
Compose invitation
Compose advertisement
Next
Done
Local Intranet

FIG. 13C

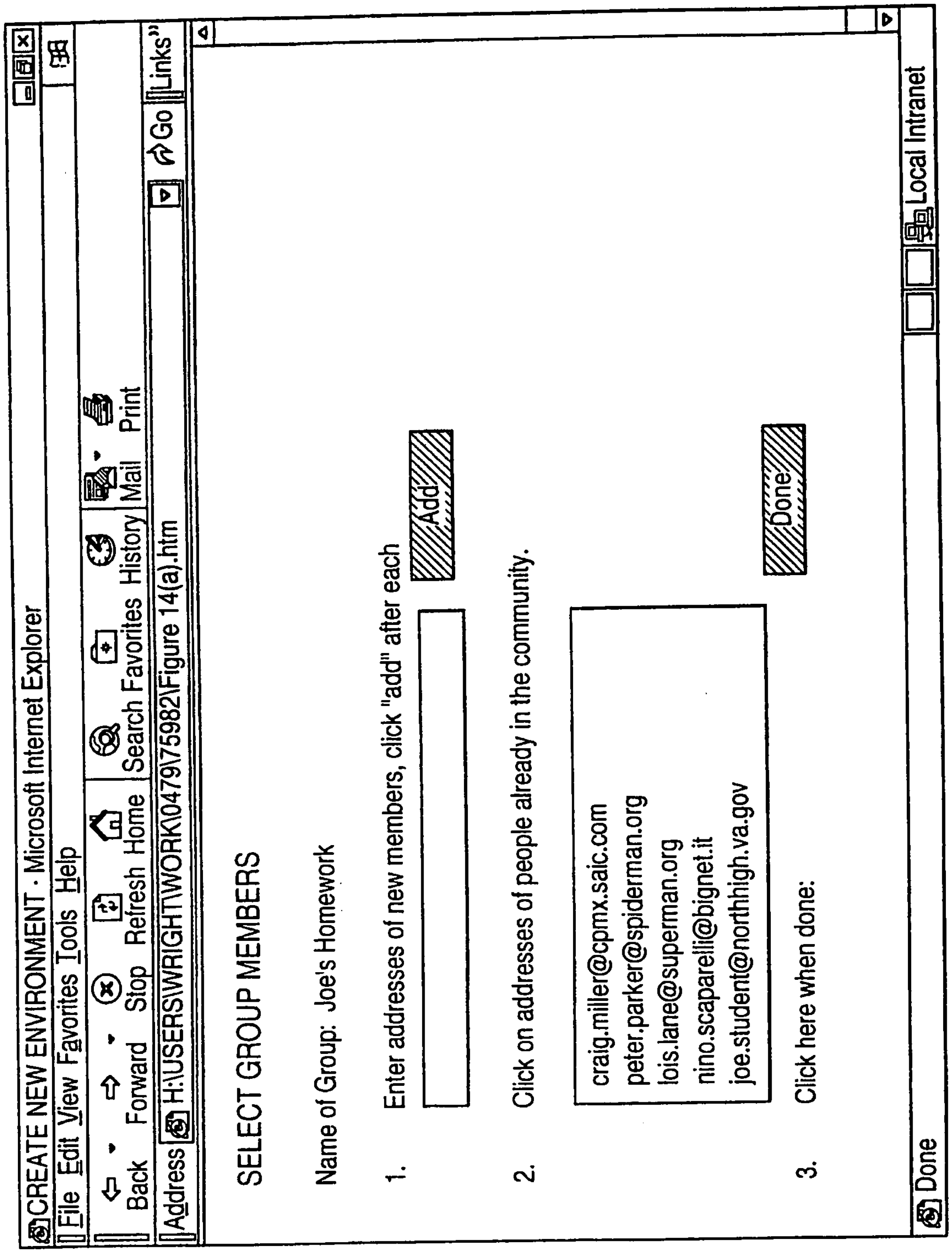
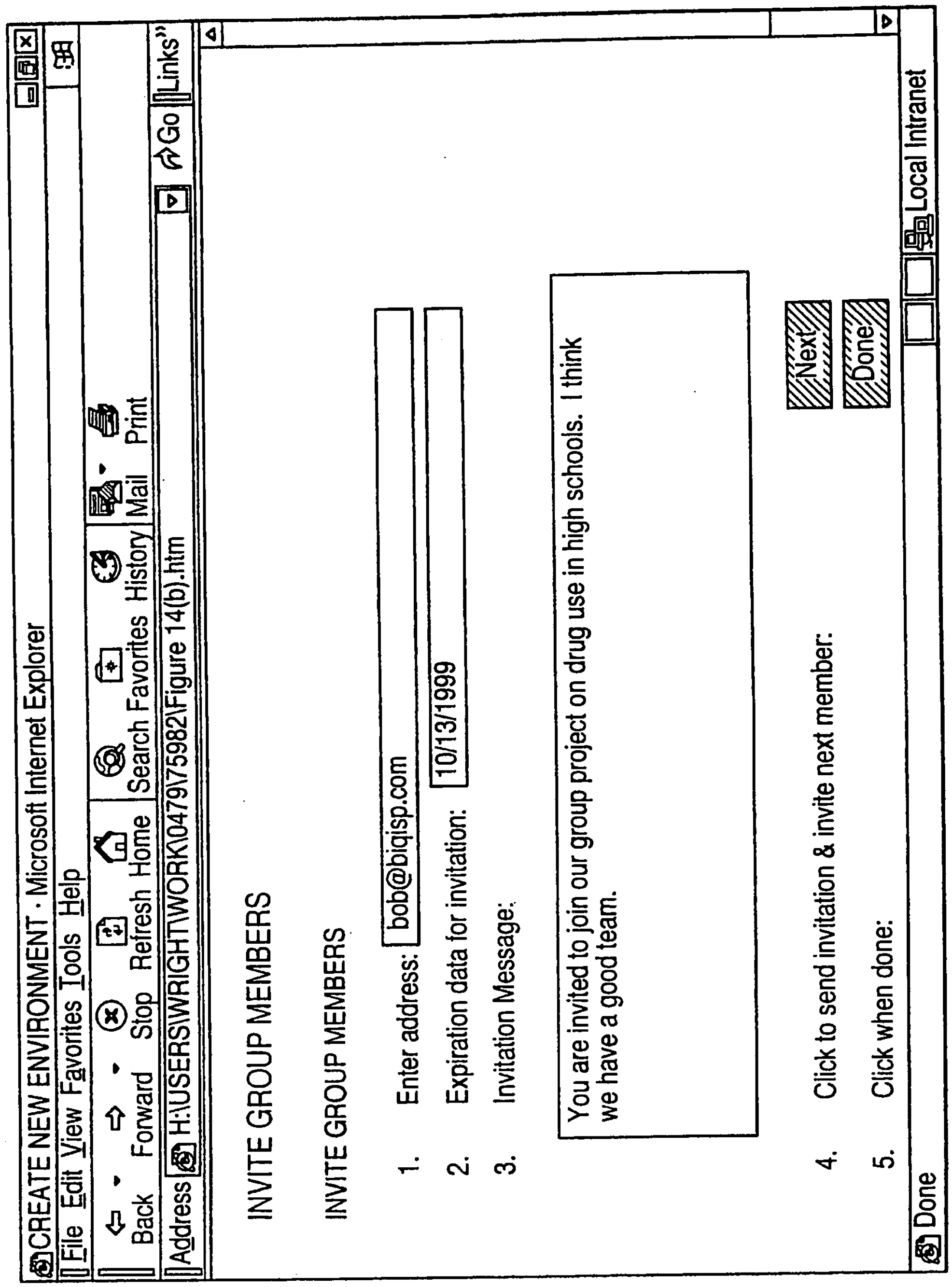

CREATE NEW ENVIRONMENT - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address H:\USERS\WRIGHT\WORK\04791\75982\Figure 14(a).htm
Go
Links"
SELECT GROUP MEMBERS
Name of Group: Joe's Homework
1. Enter addresses of new members, click "add" after each
Add
2. Click on addresses of people already in the community.
craig.miller@cpmx.saic.com
peter.parker@spiderman.org
lois.lane@superman.org
nino.scaparelli@bignet.it
joe.student@northhigh.va.gov
3. Click here when done:
Done
Done
Local Intranet

FIG. 14A

FIG. 14B


CREATE NEW ENVIRONMENT - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address H:\USERS\WRIGHT\WORK\04791\75982\Figure 14(b).htm
Go Links"
INVITE GROUP MEMBERS
INVITE GROUP MEMBERS
1. Enter address:
bob@bigisp.com
2. Expiration data for invitation:
10/13/1999
3. Invitation Message:
You are invited to join our group project on drug use in high schools. I think we have a good team.
4. Click to send invitation & invite next member:
Next
5. Click when done:
Done
Done
Local Intranet

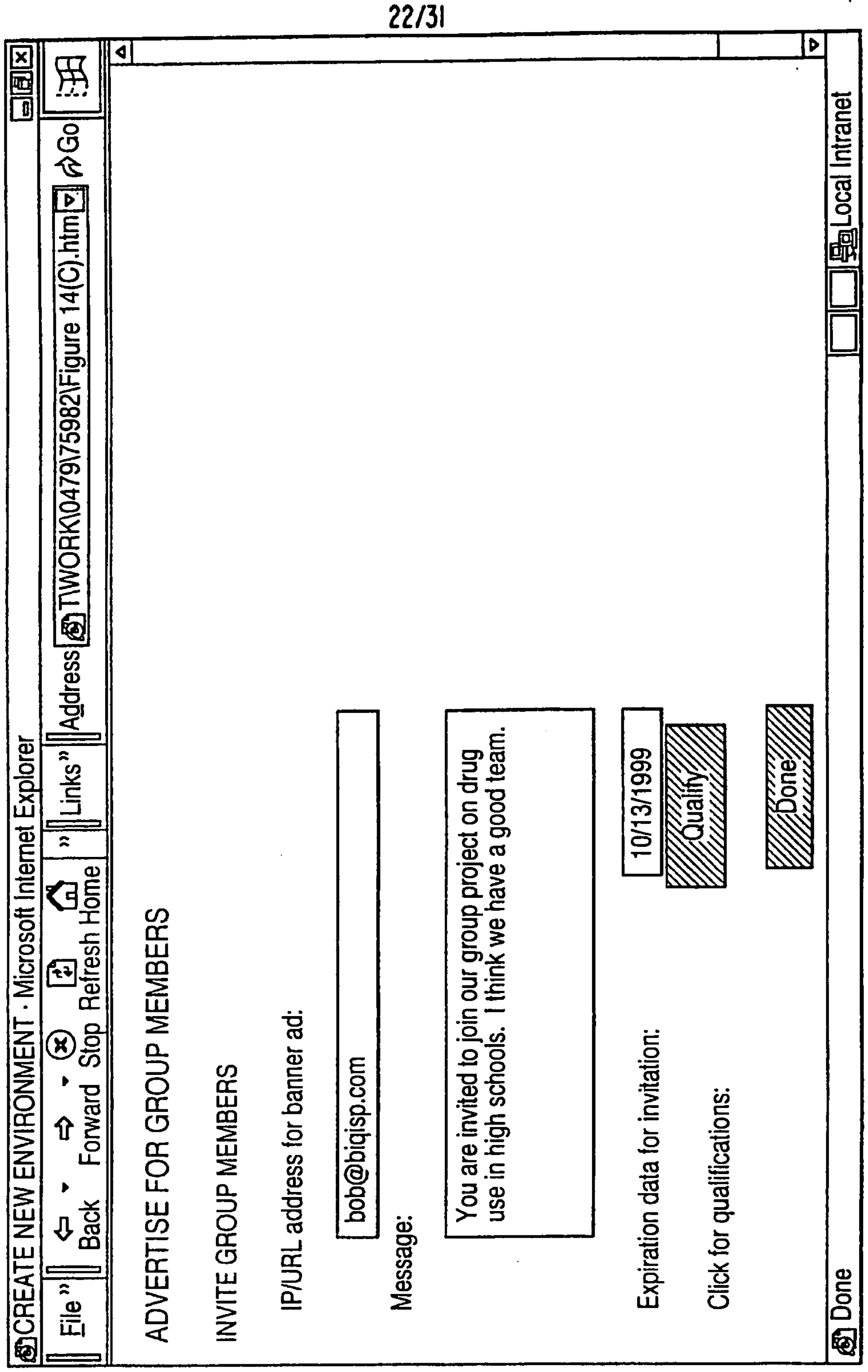

CREATE NEW ENVIRONMENT - Microsoft Internet Explorer
File
Back
Forward
Stop
Refresh
Home
Links
Address
T:\WORK\0479\75982\Figure 14(C).htm
Go
ADVERTISE FOR GROUP MEMBERS
INVITE GROUP MEMBERS
IP/URL address for banner ad:
bob@biqisp.com
Message:
You are invited to join our group project on drug use in high schools. I think we have a good team.
Expiration data for invitation:
10/13/1999
Click for qualifications:
Qualify
Done
Done
Local Intranet

FIG. 14C

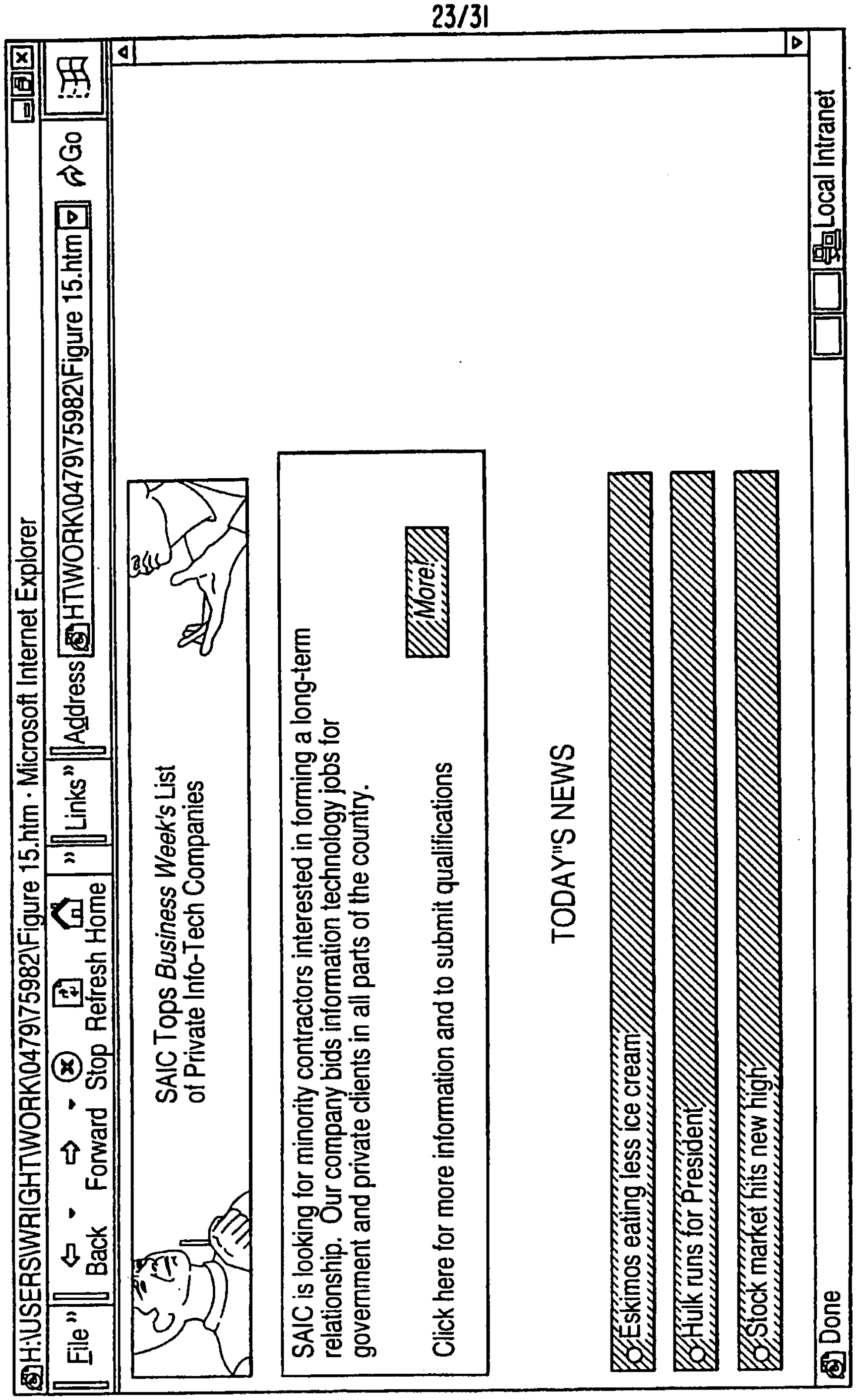

H:\USERS\WRIGHT\WORK\0479\75982\Figure 15.htm - Microsoft Internet Explorer
File
Back
Forward
Stop
Refresh
Home
Links
Address
H:\WORK\0479\75982\Figure 15.htm
Go
SAIC Tops Business Week's List
of Private Info-Tech Companies
SAIC is looking for minority contractors interested in forming a long-term relationship. Our company bids information technology jobs for government and private clients in all parts of the country.
Click here for more information and to submit qualifications
More!
TODAY"S NEWS
Eskimos eating less ice cream
Hulk runs for President
Stock market hits new high
Done
Local Intranet

FIG. 15

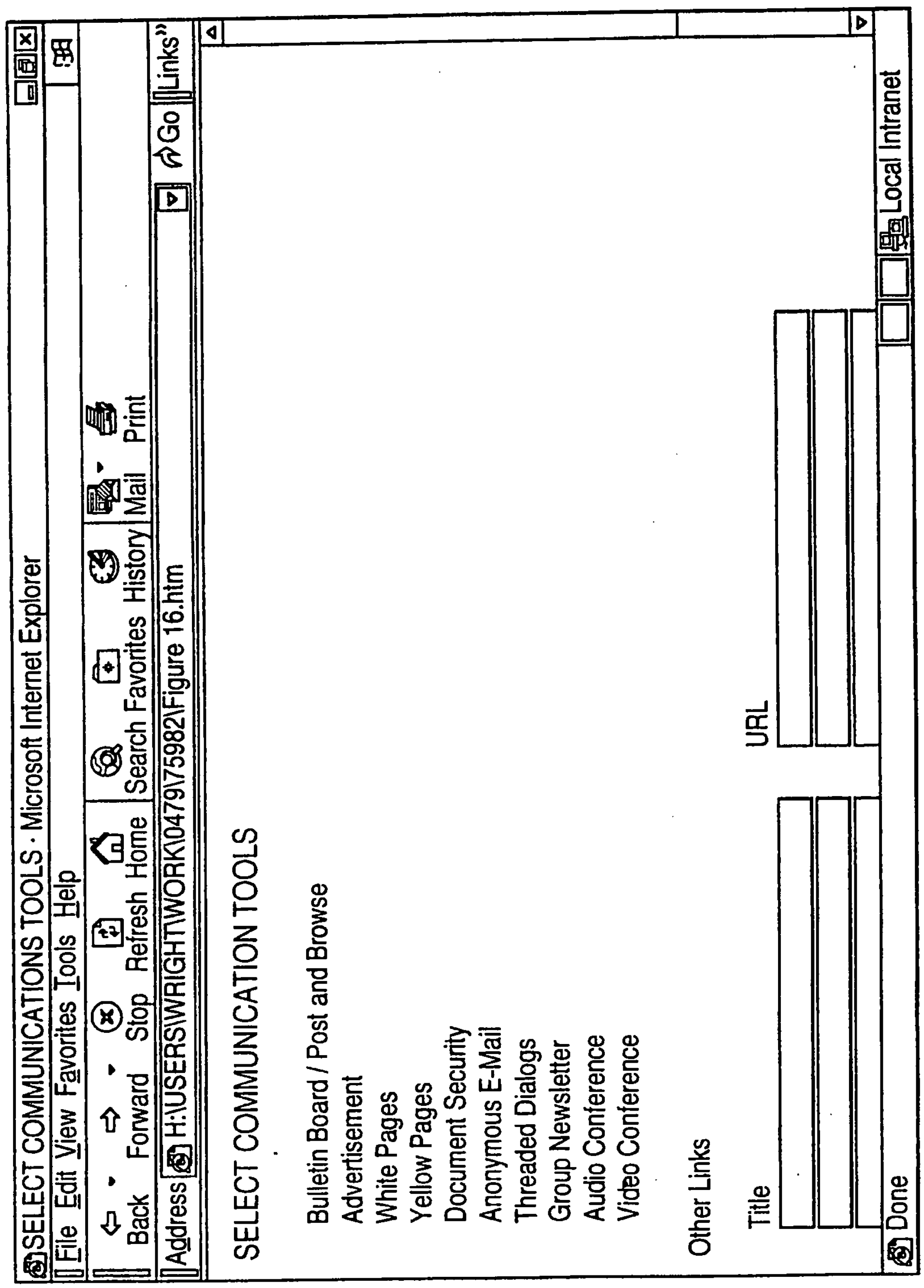

SELECT COMMUNICATIONS TOOLS - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address H:\USERS\WRIGHT\WORK\04791\75982\Figure 16.htm
Go
Links"
SELECT COMMUNICATION TOOLS
Bulletin Board / Post and Browse
Advertisement
White Pages
Yellow Pages
Document Security
Anonymous E-Mail
Threaded Dialogs
Group Newsletter
Audio Conference
Video Conference
Other Links
Title
URL
Done
Local Intranet

FIG. 16

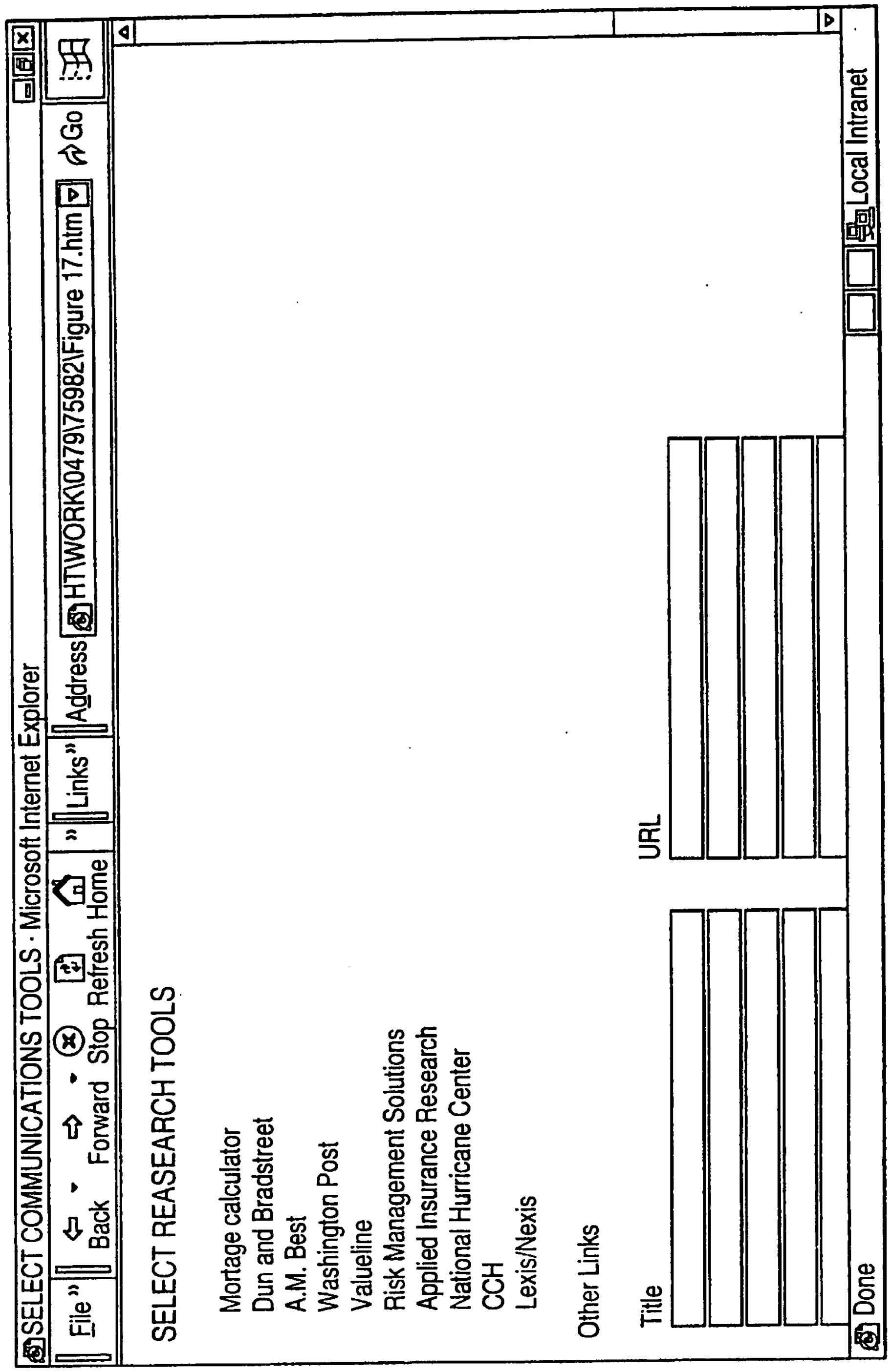

SELECT COMMUNICATIONS TOOLS - Microsoft Internet Explorer
File
Back
Forward
Stop
Refresh
Home
Links
Address
HT\WORK\0479\75982\Figure 17.htm
Go
SELECT REASEARCH TOOLS
Mortage calculator
Dun and Bradstreet
A.M. Best
Washington Post
Valueline
Risk Management Solutions
Applied Insurance Research
National Hurricane Center
CCH
Lexis/Nexis
Other Links
Title
URL
Done
Local Intranet

FIG. 17

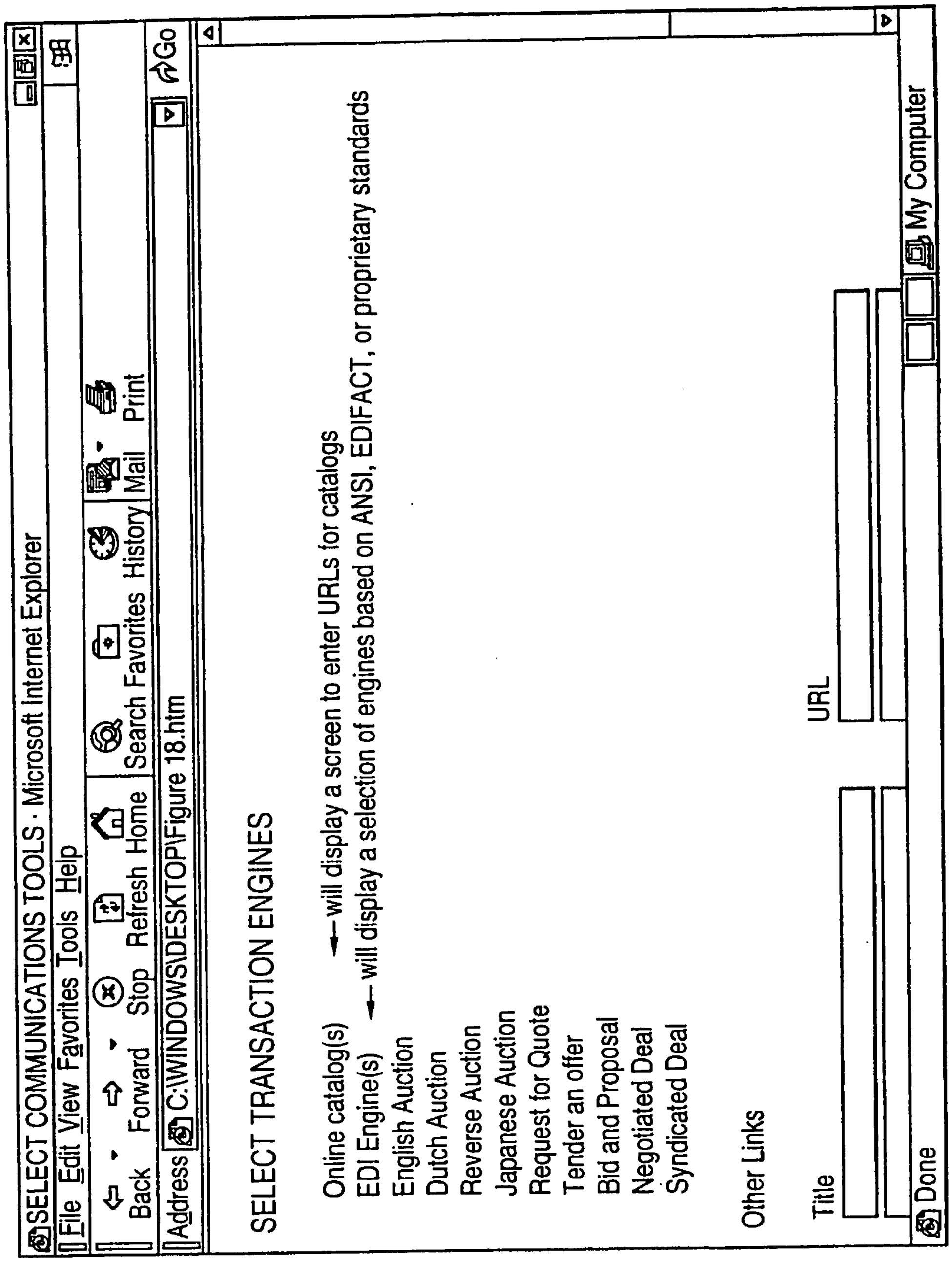

SELECT COMMUNICATIONS TOOLS - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print
Address C:\WINDOWS\DESKTOP\Figure 18.htm
Go
SELECT TRANSACTION ENGINES
Online catalog(s) ← will display a screen to enter URLs for catalogs
EDI Engine(s) ← will display a selection of engines based on ANSI, EDIFACT, or proprietary standards
English Auction
Dutch Auction
Reverse Auction
Japanese Auction
Request for Quote
Tender an offer
Bid and Proposal
Negotiated Deal
Syndicated Deal
Other Links
Title
URL
Done
My Computer

FIG. 18

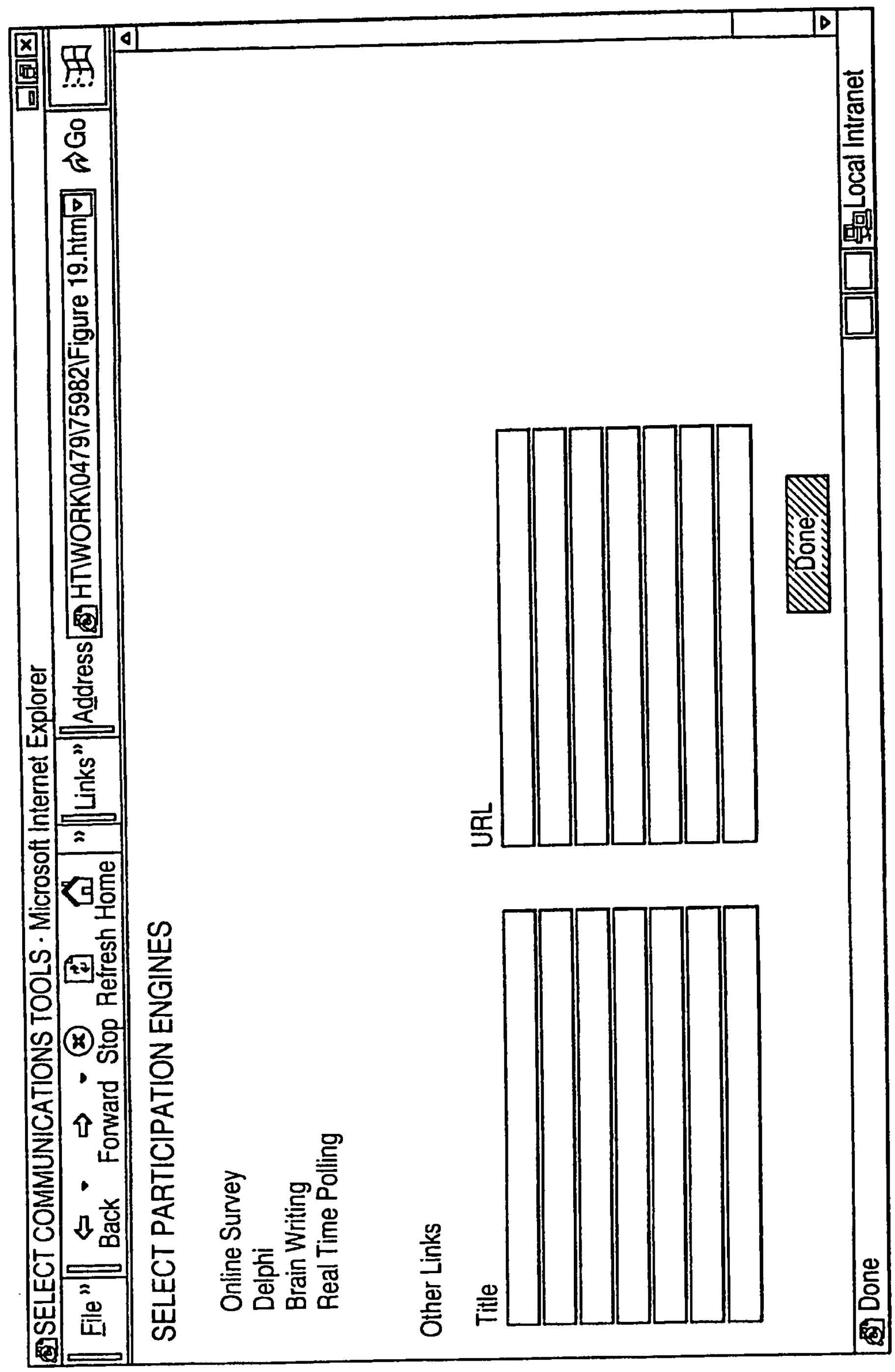
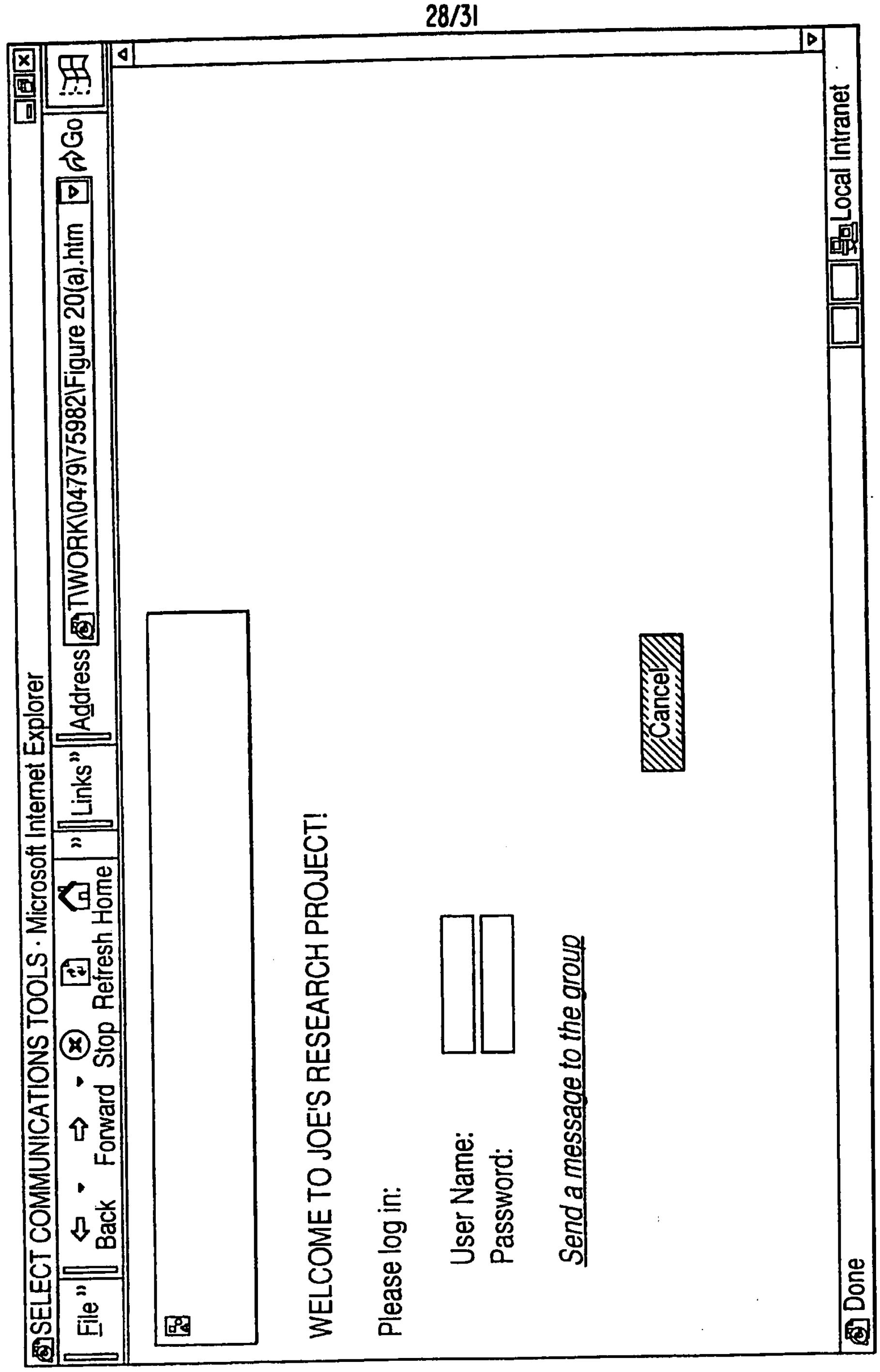


FIG. 19

SUBSTITUTE SHEET (RULE 26)



SELECT COMMUNICATIONS TOOLS - Microsoft Internet Explorer
File
Back
Forward
Stop
Refresh
Home
Links
Address
T:\WORK\0479\75982\Figure 20(a).htm
Go
WELCOME TO JOE'S RESEARCH PROJECT!
Please log in:
User Name:
Password:
Send a message to the group
Cancel
Done
Local Intranet

FIG. 20A

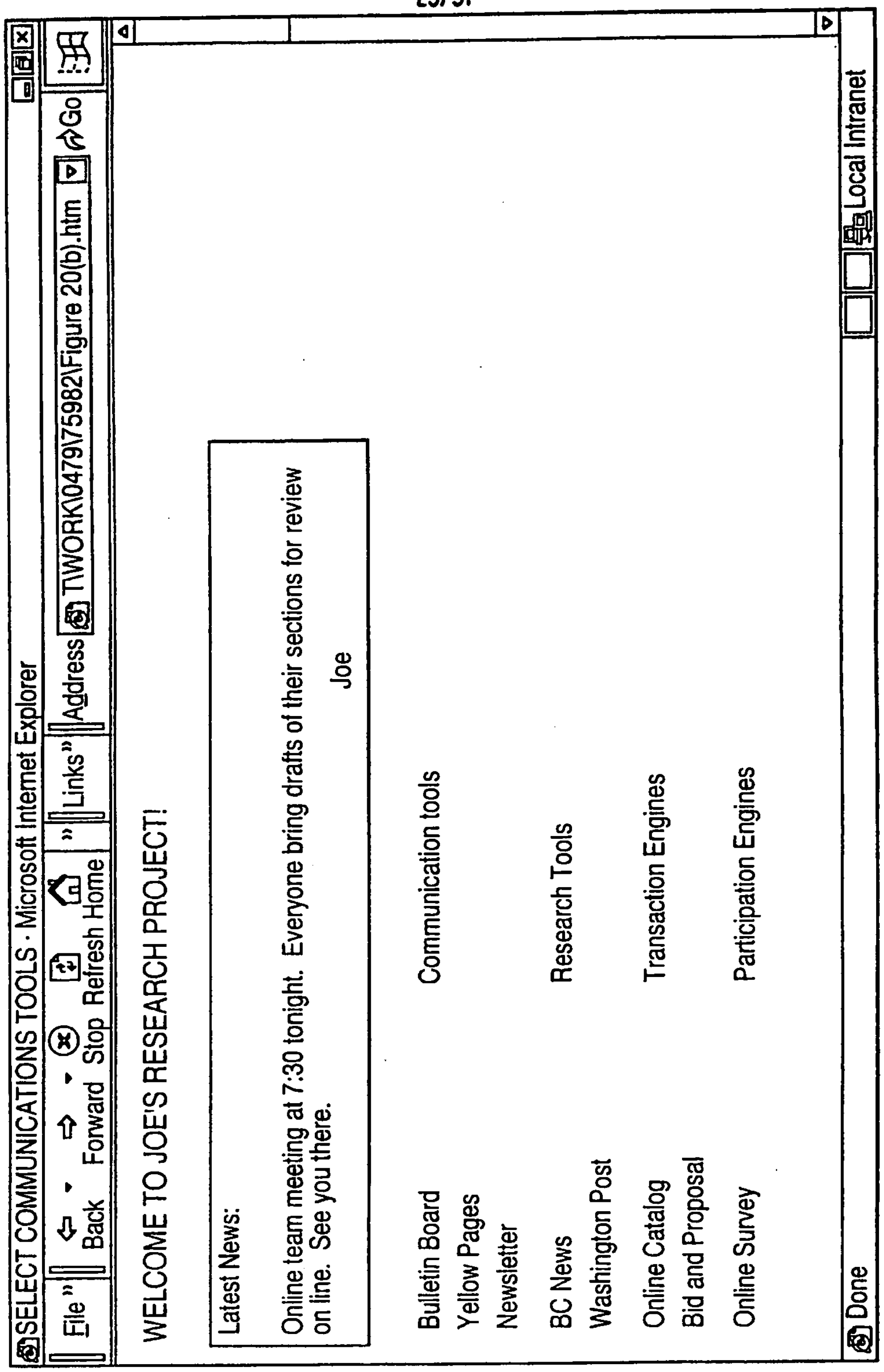

SELECT COMMUNICATIONS TOOLS - Microsoft Internet Explorer
File
Back
Forward
Stop
Refresh
Home
Links
Address
TWORK\0479\75982\Figure 20(b).htm
Go
WELCOME TO JOE'S RESEARCH PROJECT!
Latest News:
Online team meeting at 7:30 tonight. Everyone bring drafts of their sections for review on line. See you there.
Joe
Bulletin Board
Yellow Pages
Newsletter
Communication tools
BC News
Washington Post
Research Tools
Online Catalog
Bid and Proposal
Transaction Engines
Online Survey
Participation Engines
Done
Local intranet

FIG. 20B

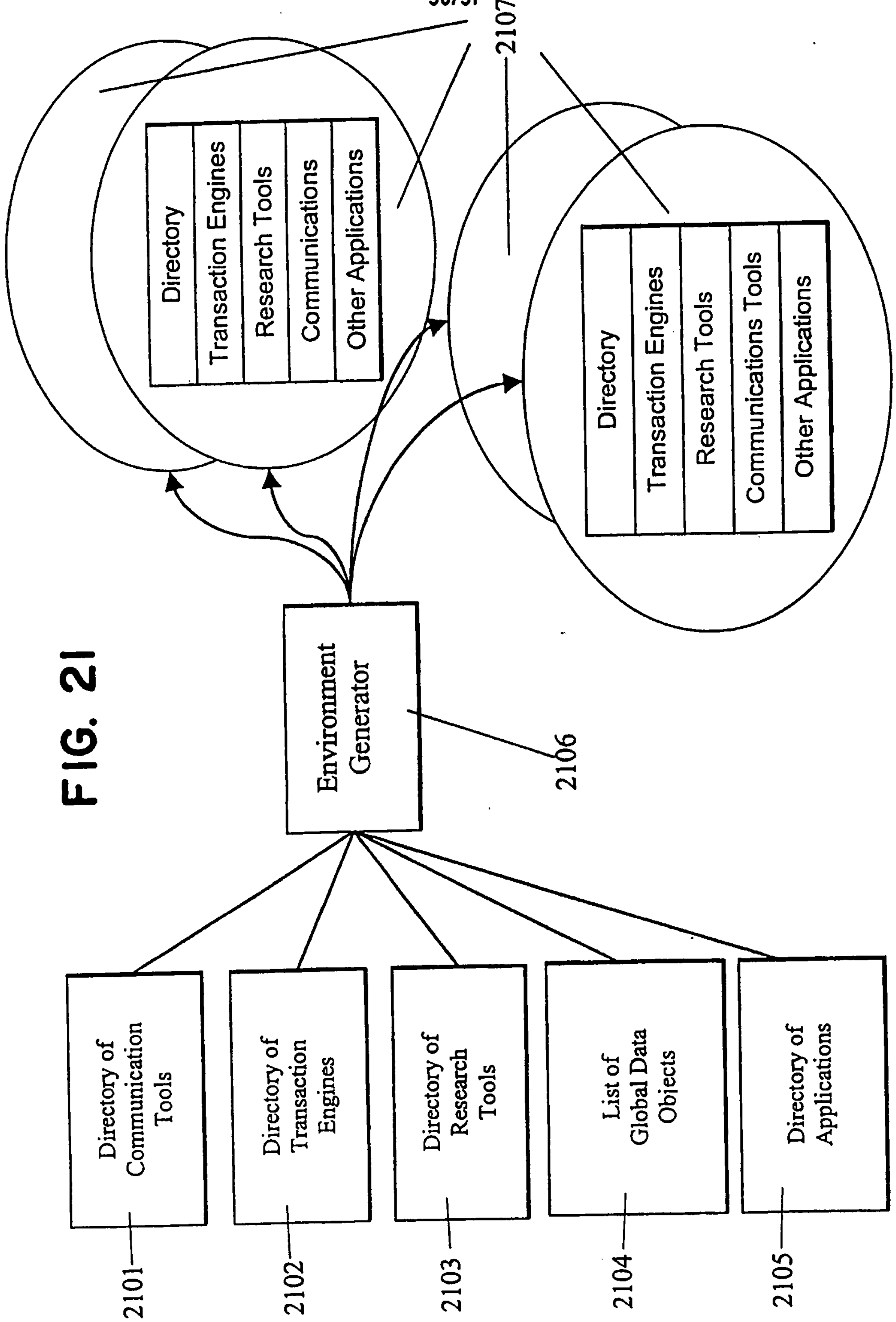
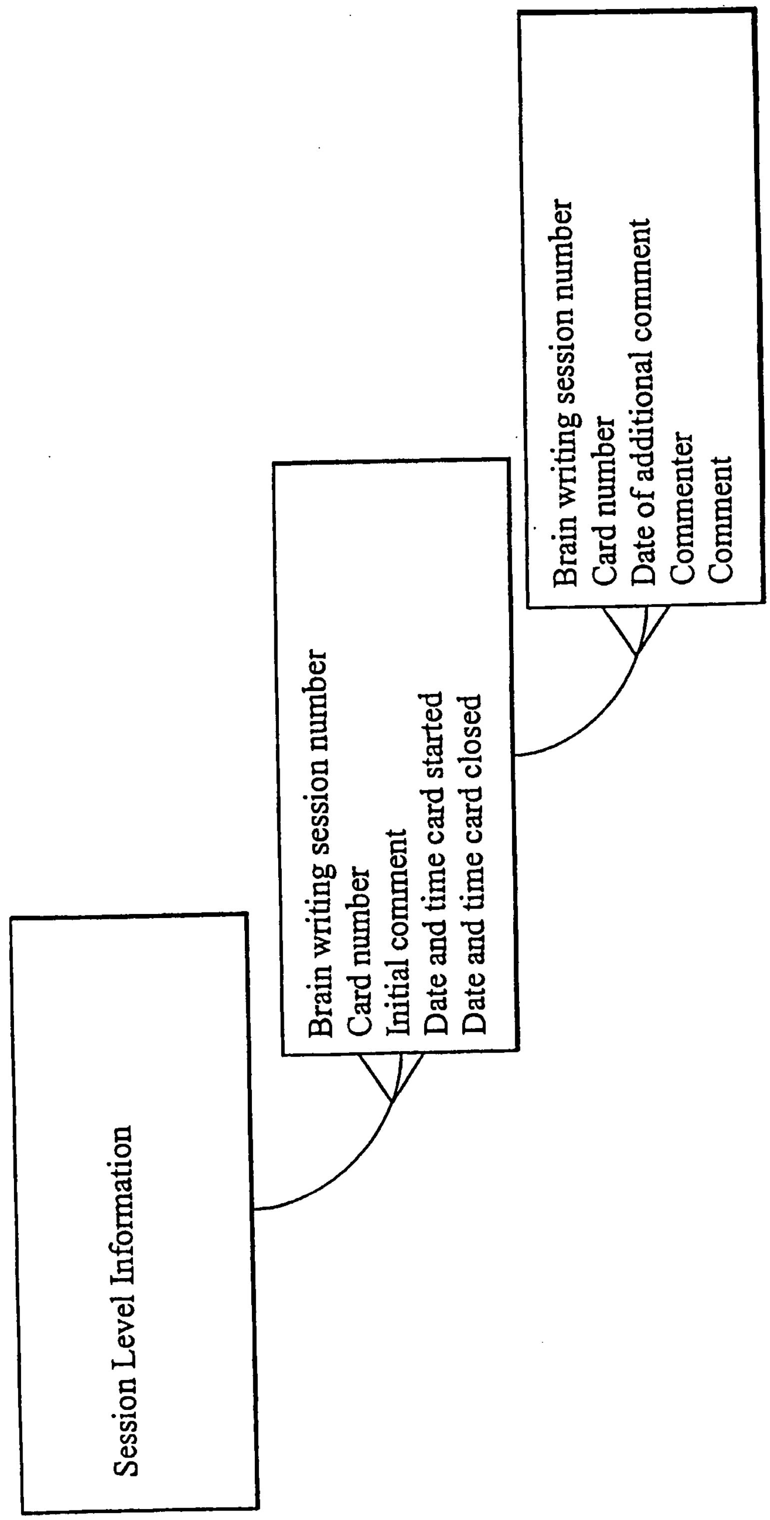

2107
Directory
Transaction Engines
Research Tools
Communications
Other Applications
Directory
Transaction Engines
Research Tools
Communications Tools
Other Applications
Environment Generator
2106
Directory of Communication Tools
Directory of Transaction Engines
Directory of Research Tools
List of Global Data Objects
Directory of Applications
2101
2102
2103
2104
2105

FIG. 21

FIG. 22


Session Level Information
Brain writing session number
Card number
Initial comment
Date and time card started
Date and time card closed
Brain writing session number
Card number
Date of additional comment
Commenter
Comment

B5

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 7 : G06F 11/00 | A1

(11) International Publication Number: WO 00/70458

(43) International Publication Date: 23 November 2000 (23.11.00)

(21) International Application Number: PCT/US00/08219

(22) International Filing Date: 15 May 2000 (15.05.00)

(30) Priority Data:
60/134,547 17 May 1999 (17.05.99) US

(71) Applicant: COMSEC CORPORATION [US/US]; 10217 Cedar Pond Drive, Vienna, VA 22182 (US).

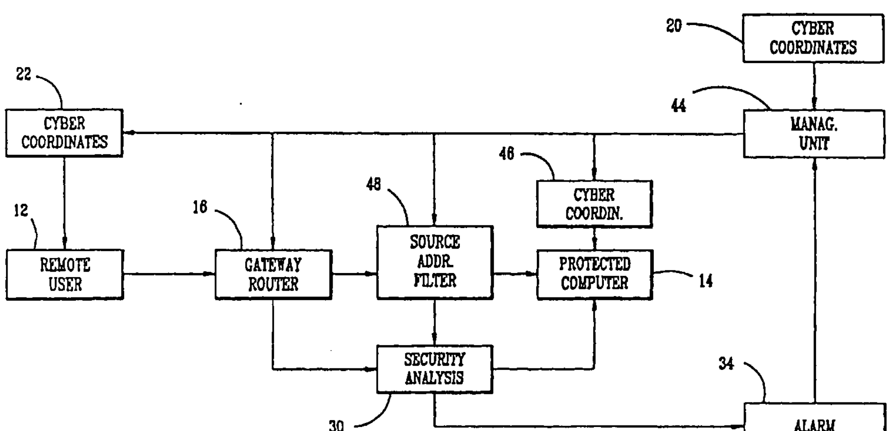
(72) Inventor: SHEYMOV, Victor, I.; 10217 Cedar Pond Drive, Vienna, VA 22182 (US).

(74) Agent: SIXBEY, Daniel, W.; Nixon Peabody LLP, Suite 800, 8180 Greensboro Drive, McLean, VA 22102 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: METHOD OF COMMUNICATIONS AND COMMUNICATION NETWORK INTRUSION PROTECTION METHODS AND INTRUSION ATTEMPT DETECTION SYSTEM



(57) Abstract

The intrusion protection method and system for a communication network provides address agility wherein the cyber coordinates of a target host (14) are changed both on a determined time schedule and when an intrusion attempt is detected. The system includes a managment unit (18) which generates a random sequence of cyber coordinates and maintains a series of tables containing the current and next set of cyber coordinates. These cyber coordinates are distributed to authorized users (12) under an encryption process to prevent unauthorized access.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav	TM	Turkmenistan
BF	Burkina Faso	GR	Greece		Republic of Macedonia	TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's	NZ	New Zealand		
CM	Cameroon		Republic of Korea	PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

METHOD OF COMMUNICATIONS AND COMMUNICATION NETWORK INTRUSION PROTECTION METHODS AND INTRUSION ATTEMPT DETECTION SYSTEM

This application is a continuation-in-part application of U.S. Serial No. 60/134,547 filed May 17, 1999.

Background Art

Historically, every technology begins its evolution focusing mainly on performance parameters, and only at a certain developmental stage does it address the security aspects of its applications. Computer and communications networks follow this pattern in a classic way. For instance, first priorities in development of the Internet were reliability, survivability, optimization of the use of communications channels, and maximization of their speed and capacity. With a notable exception of some government systems, communications security was not an early high priority, if at all. Indeed, with a relatively low number of users at initial stages of Internet development, as well as with their exclusive nature, problems of potential cyber attacks would have been almost unnatural to address, considering the magnitude of other technical and organizational problems to overcome at that time. Furthermore, one of the ideas of the Internet was "democratization" of communications channels and of access to information, which is almost contradictory to the concept of security. Now we are faced with a situation, which requires adequate levels of security in communications while preserving already achieved "democratization" of communications channels and access to information.

All the initial objectives of the original developers of the Internet were achieved with results spectacular enough to almost certainly surpass their expectations. One of the most remarkable results of the Internet development to date is the mentioned "democratization". However in its unguarded way "democratization" apparently is either premature to a certain percentage of the Internet users, or contrary to human nature, or both. The fact remains that this very percentage of users presents a serious threat to the integrity of national critical infrastructure, to privacy of information, and to further advance of commerce by utilization of the Internet capabilities. At this stage it seems crucial to address security issues but, as usual, it is desirable to be done within already existing structures and technological conventions.

Existing communications protocols, while streamlining communications, still lack underlying entropy sufficient for security purposes. One way to increase entropy, of course, is encryption as illustrated by U.S. Patent No. 5,742,666 to Finley. Here each node in the Internet encrypts the destination address with a code which only the next node can unscramble.

Encryption alone has not proven to be a viable security solution for many communications applications. Even within its core purpose, encryption still retains certain security problems, including distribution and safeguarding of the keys. Besides, encryption represents a "ballast", substantially reducing information processing speed and transfer time. These factors discourage its use in many borderline cases.

Another way is the use of the passwords. This method has been sufficient against humans, but it is clearly not working against computers. Any security success of the password-based security is temporary at best. Rapid advances in computing power make even the most sophisticated password arrangement a short-term solution.

Recent studies clearly indicate that the firewall technology, as illustrated by U.S. Patent No. 5,898,830 to Wesinger et al., also does not provide a sufficient long-term solution to the security problem. While useful to some extent, it cannot alone withstand the modern levels of intrusion cyber attacks.

On the top of everything else, none of the existing security methods, including encryption, provides protection against denial of service attacks. Protection against denial of service attacks has become a critical aspect of communication system security. All existing log-on security systems, including those using encryption, are practically defenseless against such attacks. Given a malicious intent of a potential attacker, it is reasonable to assume that, even having failed with an intrusion attempt, the attacker is still capable of doing harm by disabling the system with a denial of service attack. Since existing systems by definition have to deal with every log-on attempt, legitimate or not, it is certain that these systems cannot defend themselves against a denial of service attack.

The deficiencies of existing security methods for protecting communications systems leads to the conclusion that a new generation of cyber protection technology is needed to achieve acceptable levels of security in network communications.

Summary of the Invention

It is a primary object of the present invention to provide a novel and improved method of communications, and a novel and improved communication network intrusion protection method and systems and novel and improved intrusion attempt detection method and systems, adapted for use with a wide variety of communication networks including Internet based computers, corporate and organizational computer networks (LANs), e-commerce systems, wireless computer communications networks, telephone dial-up systems, wireless dial-up systems, wireless telephone and computer communications systems, cellular and satellite telephone systems, mobile telephone and mobile communications systems, cable based systems and computer databases, as well as protection of network nodes such as routers, switches, gateways, bridges, and frame relays.

Another object of the present invention is to provide a novel and improved communication network intrusion protection method and system which provides address agility combined with a limited allowable number of log-on attempts.

Yet another object of the present invention is to provide a novel and improved intrusion protection method for a wide variety of communication and other devices which may be accessed by a number, address code, and/or access code. This number, address code, and/or access code is periodically changed and the new number, address code, or access code is provided only to authorized users. The new number, address code, or access code may be provided to a computer or a device for the authorized user and not be accessible to others. This identifier causes the user's computer to transmit the otherwise unknown and inaccessible number, address code, and/or access code.

A still further object of the present invention is to provide a novel and improved communication network intrusion protection method and system wherein a plurality of different cyber coordinates must be correctly provided before access is granted to a protected communications unit or a particular piece of information. If all or some cyber coordinates are not correctly provided, access is denied, an alarm situation is instigated and the affected cyber coordinates may be instantly changed.

For the purposes of this invention cyber coordinates are defined as a set of statements determining location of an object (such as a computer) or a piece of information (such as a computer file) in cyber space. Cyber coordinates include but are not limited to private or public protocol network addresses such as an IP address in the Internet, a computer port number or designator, a computer or database directory, a file name or designator, a telephone number , an access number and/or code, etc.

These and other objects of the present invention are achieved by providing a communication network intrusion protection method and system where a potential intruder must first guess where a target computer such as a host workstation is in cyber space and to predict where the target computer such as a workstation will next be located in cyber space. This is achieved by changing a cyber coordinate (the address) or a plurality of cyber coordinates for the computers such as workstations on a determined or random time schedule and making an unscheduled cyber coordinates change when the system detects an intrusion attempt. A limited number of log-on attempts may be permitted before an intrusion attempt is confirmed and the cyber coordinates are changed. A management unit is provided for generating a random sequence of cyber coordinates and which maintains a series of tables containing current and the next set of addresses. These addresses are distributed to authorized parties, usually with use of an encryption process.

The present invention further provides for a piece of information, a computer or a database intrusion protection method and system where a potential intruder must first guess where a target piece of information such as a computer file or a directory is in cyber space and to predict where the target piece of information will be next in cyber space. This is achieved by changing a cyber coordinate or a plurality of cyber coordinates for the piece of information on a determined or random time schedule and making an unscheduled cyber coordinates change when the system detects an intrusion attempt. A limited number of log-on attempts may be permitted before an intrusion attempt is confirmed and the coordinates changed. A management unit is provided for generating a random sequence of cyber coordinates and which maintains a series of tables containing current and the next set of cyber coordinates. These coordinates are distributed to authorized parties, usually by means of an encryption process.

The intrusion attempt detection methods and systems are provided to the protected devices and pieces of information as described above by means of categorizing a log-on attempt when all or some of the correct cyber coordinates are not present as an intrusion attempt and by instigating an alarm situation.

Brief Description of the Drawings

Figure 1 is a block diagram of the communication network protection system of the present invention;

Figure 2 is a flow diagram showing the operation of the system of Figure 1;

Figure 3 is a block diagram of a second embodiment of the communication network protection system of the present invention;

Figure 4 is a flow diagram showing the operation of the system of Figure 3;

Figure 5 is a block diagram of a third embodiment of the communication network protection system of the present invention;

Figure 6 is a flow diagram showing the operation of the system of Figure 5; and

Figure 7 is a block diagram of a fourth embodiment of the communication network protection system of the present invention.

Description of the Preferred Embodiments

Existing communications systems use fixed coordinates in cyber space for the communications source and communications receiver. Commonly accepted terminology for the Internet refers to these cyber coordinates as source and destination IP addresses. For purposes of an unauthorized intrusion into these communication systems, the situation of a cyber attack might be described in military terms as shooting at a stationary target positioned at known coordinates in cyber space. Obviously, a moving target is more secure than the stationary one, and a moving target with coordinates unknown to the intruder is more secure yet. The method of the present invention takes advantage of the cyber space environment and the fact that the correlation between the physical coordinates of computers or other communication devices and their cyber coordinates is insignificant.

While it is difficult to change the physical coordinates of computers or other communications devices, their cyber coordinates (cyber addresses) can be changed much easier, and in accordance with the present invention, may be variable and changing over time. In addition to varying the cyber coordinates over time, the cyber coordinates can immediately be changed when an attempted intrusion is sensed. Furthermore, making the current cyber coordinates available to only authorized parties makes a computer or other communications device a moving target with cyber coordinates unknown to potential attackers. In effect, this method creates a device which perpetually moves in cyber space.

Considering first the method of the present invention as applied to computers and computer networks, the computer's current cyber address may serve also as its initial log-on password with a difference that this initial log-on password is variable. A user, however, has to deal only with a computer's permanent identifier, which is, effectively its assigned "name" within a corresponding network. Any permanent identifier system can be used, and an alphabetic "name" system seems to be reasonably user-friendly. One of such arrangements would call for using a computer's alphabetic Domain Name System, as a cyber address permanent identifier, while subjecting its numeric, or any other cyber address to a periodic change with regular or irregular intervals. This separation will make the security system transparent to the user, who will have to deal only with the alphabetic addresses. In effect, the user's computer would contain an "address book" where the alphabetic addresses are permanent, and the corresponding variable addresses are more complex and periodically updated by a network's management. While a user is working with other members of the network on the name or the alphabetic address basis, the computer conducts communications based on the corresponding variable numeric or other addresses assigned for that particular time.

A variable address system can relatively easily be made to contain virtually any level of entropy, and certainly enough entropy to defy most sophisticated attacks. Obviously, the level of protection is directly related to the level of entropy contained in the variable address system and to the frequency of the cyber address change.

This scenario places a potential attacker in a very difficult situation when he has to find the target before launching an attack. If a restriction on a number of allowable log-on tries is implemented, it becomes more difficult for an attacker to find the target than to actually attack it. This task of locating the target can be made difficult if a network's cyber address system contains sufficient entropy. This difficulty is greatly increased if the security system also limits the number of allowable log-on tries, significantly raising the entropy density.

For the purpose of this invention, entropy density is defined as entropy per one attempt to guess a value of a random variable.

Figure 1 illustrates a simple computer intrusion protection system 10 which operates in accordance with the method of the present invention. Here, a remote user's computer 12 is connected to a protected computer 14 by a gateway router or bridge 16. A management system 18 periodically changes the address for the computer 14 by providing a new address from a cyber address book 20 which stores a plurality of cyber addresses. Each new cyber address is provided by the management system 18 to the router 16 and to a user computer address book 22. The address book 22 contains both the alphabetic destination address for the computer 14 which is available to the user and the variable numeric cyber address which is not available to the user. When the user wants to transmit a packet of information with the alphabetic address for the computer 14, this alphabetic address is automatically substituted for the current numerical cyber address and used in the packet.

With the reference to Figures 1 and 2, when a packet is received by the gateway router or bridge 16 as indicated at 24, the cyber address is checked by the gateway router or bridge at 26, and if the destination address is correct, the packet is passed at 28 to the computer 14. If the destination address is not correct, the packet is directed to a security analysis section 30 which, at 32 determines if the packet is retransmitted with a correct address within a limited number of log-in attempts. If this occurs, the security analysis section transmits the packet to the computer 14 at 28. However, if no correct address is received within the allowed limited number of log-in attempts, the packet is not transmitted to the computer 14 and the security analysis section activates an alarm section 34 at 36 which in turn causes the management section to immediately operate at 38 to change the cyber address.

Sophisticated cyber attacks often include intrusion through computer ports other than the port intended for a client log-on. If a system principally described in connection with Figures 1 and 2 is implemented, the port vulnerability still represents an opening for an attack from within the network, that is if an attacker has even a low-level authorized access to a particular computer and thus knows its current variable address.

Computer ports can be protected in a way similar to protection of the computer itself. In this case port assignment for the computer becomes variable and is changed periodically in a manner similar to that described in connection with Figures 1 and 2. Then, a current assignment of a particular port is communicated only to appropriate parties and is not known to others. At the same time, similarly to methods described, a computer user would deal with permanent port assignments, which would serve as the ports' permanent "names".

This arrangement in itself may not be sufficient, however, to reliably protect against a port attack using substantial computing power because of a possible insufficient entropy density. Such a protection can be achieved by implementing an internal computer "port router" which would serve essentially the same role for port identifiers as the common gateway router or bridge 16 serves for computer destination addresses.

With reference to Figures 3 and 4 wherein like reference numerals are used for components and operations which are the same as those previously described in connection with Figures 1 and 2, a port router 40 is provided prior to the protected computer 14, and this port router is provided with a port number or designator by the management unit 18. This port number or designator is also provided to the user address book 22 and will be changed when the cyber address is changed, or separately. Thus, with reference to Figure 4, once the cyber address has been cleared at 26, the port number or designator is examined at 42. If the port number is also correct, the data packet will be passed to the computer 14 at 28. If the port number is initially incorrect, the packet is directed to the security analysis section 30 which at 32 determines if the packet is retransmitted with the correct port number within the limited number of log-in attempts.

The port protection feature can be used independently of other features of the system. It can effectively protect nodes of the infrastructure such as routers, gateways, bridges, and frame relays from unauthorized access. This can protect systems from an attacker staging a cyber attack from such nodes.

The method and system of the present invention may be adapted to provide security for both Internet based computer networks and private computer networks such as LANs.

Internet structure allows the creation of an Internet based Private Cyber Network (PCN) among a number of Internet-connected computers. The main concern for using the Internet for this purpose as an alternative to the actual private networks with dedicated communication channels is security of Internet-based networks.

The present invention facilitates establishment of adequate and controllable level of security for the PCNs. Furthermore, this new technology provides means for flexible structure of a PCN, allowing easy and practically instant changes in its membership. Furthermore, it allows preservation of adequate security in an environment where a computer could be a member of multiple PCNs with different security requirements. Utilizing the described concept, a protected computer becomes a "moving target" for the potential intruders where its cyber coordinates are periodically changed and the new coordinates are communicated on a "need to know" basis only to the other members of the PCN authorized to access this computer along with appropriate routers and gateways. This change of cyber coordinates can be performed either by previous arrangement or by communicating future addresses to the authorized members prior to the change. Feasible frequency of such a change can range from a low extreme of a stationary system changing cyber coordinates only upon detection of a cyber attack to an extremely high frequency such as with every packet. The future coordinates can be transmitted either encrypted or unencrypted. Furthermore, each change of position of each PCN member can be made random in terms of both its current cyber coordinates and the time of the coordinates change. These parameters of a protected PCN member's cyber moves are known only to the PCN management, other PCN members with authorization to communicate with this particular member, and appropriate gateways and routers. PCN management would implement and coordinate periodic cyber coordinates changes for all members of the PCN. While the PCN management is the logical party to make all the notification of the cyber coordinates changes, in certain instances it could be advantageous to shift a part of this task to a PCN member computer itself. With certain limitations, the routers and gateways with the "need to know" the current address of the protected computer are located in cyber space in the general vicinity of the protected computer. In such instances the protected computer could be in a better position to make the mentioned notifications of nearby routers and gateways.

The address changes could be done simultaneously for all the members of the PCN, or separately, particularly if security requirements for the members substantially differ. The latter method is advantageous, for instance, if some of the computers within the PCN are much more likely than others to be targeted by potential intruders. A retail banking PCN could be an example of such an arrangement where the bank's computer is much more likely to be attacked than a customer's computer. It should be noted that, while in certain cases some members of the PCN may not require any protection at all, it still is prudent to provide it as long as the computer belongs to a protected PCN. The correct "signature" of the current "return address" would serve as additional authenticity verification. In the above example of the retail banking, while many customers' computers may not require any protection, assigning variable addresses to them would serve as an additional assurance to the bank that every log-on is authorized. In fact, this system automatically provides two-tier security. In order to reach a protected computer, the client computer has to know the server computer current cyber address in the first place. Then, even if a potential intruder against odds "hits" the correct current address the information packet is screened for the correct "signature" or return address. If that signature does not belong to the list of the PCN's current addresses, the packet is rejected. In high security instances this should trigger an unscheduled address change of the protected computer.

With the reference to Figures 5 and 6 which illustrate this two-tier security system, a network management unit 44 provides different unique cyber coordinates to the address books for each computer in the system (two computers 12 and 14 with address books 22 and 46 respectively being shown). Now when the computer 12 sends a data packet to the computer 14, the gateway router or bridge 16, first checks for the correct current destination address for the computer 14 at 26 in the manner previously described. If the destination address is correct, a source address sensor 48 checks at 50 to determine if the correct source address (i.e. return address) for the computer 12 is also present. If both correct addresses are present, the data packet is passed to the computer 14 at 28, but if the correct source address is not present, the data packet is passed to the security analysis section 30 where at 32 where it is determined if a correct source address is received within the acceptable number of log-on tries. If the correct return address is not received, an alarm situation is activated at 36 and the network management system operates at 38 to change the cyber address of the computer 14

In addition to the penetration (hacking) detection and protection, the system above provides real-time detection of a cyber attack and protection against "flooding" denial of service attacks. A gateway router or bridge 16 filters all the incorrectly addressed packets thus protecting against "flooding". Further yet, since the "address book" of the protected network contains only trusted destinations, this system also protects against instructive viruses or worms if such are present or introduced into the network. For the purpose of this invention, an instructive virus or worm is defined as a foreign unit of software introduced into a computer system so it sends certain computer data to otherwise unauthorized parties outside of the system.

Elements of the system described above are: a gateway router or bridge 16, a computer protection unit, and a management unit. A gateway router or bridge represents an element of collective defense for the network, while the source address filter and the "port router" and filter represent a unit of individual defense for a member computer. This individual defense unit (server unit) can be implemented either as a standalone computer, as a card in the protected computer, as software in the protected computer, or imbedded into the protected computer operating system. For further improvement of the overall security, port assignments can be generated autonomously from the management unit thus creating a "two keys" system in a cryptographic sense. This would allow for security to still be in place even if a security breach happened at the security management level.

The method and system of the present invention minimize human involvement in the system. The system can be configured in such a way that computer users deal only with simple identifiers or names permanently assigned to every computer in the network. All the real (current) cyber coordinates can be stored separately and be inaccessible to the user, and could be available to the appropriate computers only. This approach both enhances security and makes this security system transparent to the user. The user deals only with the simple alphabetic side of the "address book", and is not bothered with the inner workings of the security system. A telephone equivalent of this configuration is an electronic white pages residing in a computerized telephone set, which is automatically updated by the telephone company. The user just has to find a name, and push the "connect" button while the telephone set does the rest of the task.

A numeric cyber address system, based on the Internet host number could be relatively easily utilized for the discussed security purposes, however a limitation exists for this address system in its current form represented by the IPv.4 protocol. This limitation is posed by the fact that the address is represented by a 32-bit number. 32-bit format does not contain sufficient entropy in the address system to enable establishment of adequate security. This is a particularly serious limitation in regard to securing an entire network. The availability of the network numbers are limited to the extent that not only entropy, but a simple permanently assigned number is becoming more and more difficult to obtain with the rapid expansion of the Internet.

If this address system is to be used for the security purposes, than the format of the host number should be adequately expanded to create sufficient size of the address numbers field in the system. If this is done, than the corresponding address in the Domain Name System (DNS) could be conveniently used as permanent identifier for a particular computer and the Internet host number would be variable, creating a moving regime of a protected computer. Currently being implemented IPv.6 (IPNG) protocol solves this problem by providing sufficient entropy.

Another way to achieve the same goal is to use the DNS address as a variable for security purposes. This way, the traditional Internet DNS address system would not be affected and no change in format is required. The relevant part of the protected computer's DNS address would become a variable, utilizing more characters than the alphabet, with a very large number of variations, also creating sufficient level of entropy.

Yet another way to implement the same method is to utilize the geographic zone-based system. While its utilization is somewhat similar to the DNS system, it offers some practical advantages for security use. Naturally, when a computer is protected by a security system, it is still essential to preserve the communication redundancy of the Internet communications. However, the redundancy may suffer if only a limited number of the routers and gateways are informed of the protected computer current cyber address. This effect could be particularly important with the members of a particular protected network vastly remote in geographic terms. The necessary notification of a large number of the routers and gateways can also become problematic, not only technically, but also because it can decrease the level of security. In this sense a geographic zone-based system offers advantages since the variable part of the computer's cyber address could be made to involve only certain geographic locale while initial routing of the information packet could be done by the traditional method. After the packet has been moved to the general vicinity of the addressee computer, it would get into the area of the "informed" routers and gateways. This scheme would simplify the notification process of the routers as well as improve security by limiting the number of the "need to know" parties. It is important to recognize that, after the "general" part of the cyber address caused the information packet to arrive in a cyber vicinity of the addressee, virtually any, even private, address system can be used for the rest of he the delivery. This would further increase the level of underlying entropy in the system.

While certain specific address systems have been discussed, it is an important quality of the present invention that it can be implemented with virtually any address system.

Corporate and organizational computer networks such as LANs or, at least those in closed configurations, do not possess as much vulnerability to cyber attacks as Internet-based networks. However, even in these cases, their remote access security is a subject of concern. This is especially visible when a private network (PN) contains information of different levels of confidentiality with access restricted to appropriate parties. In other words, along with other generally accessible organizational information, an organizational PN can contain information restricted to certain limited groups. Enforcement of these restrictions requires a remote access security system. Usually these security systems employ a password-based scheme of one type or another and, perhaps, a firewall. However, reliance on passwords may not be entirely justified since the passwords can be lost or stolen, giving a malicious insider with a low access level a reasonable chance of access to information intended only for higher levels of access. Furthermore, in some cases use of cracking techniques from such a position is not entirely out of the question. Such an occurrence can relatively easily defeat both the password and the firewall. This would prevent a LAN from a cyber attack launched from within the network.

The present invention provides adequate security to such PCNs without reliance on the passwords and to limit access to only appropriate computers. Then, the task of overall information access security practically would be narrowed down to control of physical access to a particular computer, usually a less complicated feat.

Similarly to the systems described for Internet-based networks, a "closed" LAN as well as an Internet-based LAN can be protected by implementation of periodic changes of the members' network addresses and communicating those changes to the appropriate parties. This way, the lowest access level computers would have the lowest rate of address change. The rate of the address change would increase with the level of access. This system would ensure that all the PCN computers with legitimate access to a particular computer within the PCN would be informed of its location. Furthermore, it will ensure that the current location of a computer with restricted information would be unknown to the parties without the legitimate access clearance. For instance, a superior's computer would be able to access his subordinate's computer but not vice versa.

Also similarly to the systems described for the PCNs, a PCN computer would contain an "address book" where the user can see and use only the permanent side of it with identifiers of all computers accessible to him while the actual communication functions are performed by the computer using the variable side of the "address book" periodically updated by the PN management. To further enhance security, in addition to the computer address system management, the PCN Administrator can implement an automatic security monitoring system where all wrongly addressed log-on attempts would be registered and analyzed for security purposes.

Thus the method and system of the present invention would allow reliable protection against unauthorized remote access to information from within a PN while providing a great deal of flexibility, where the granted access can be revised easily and quickly.

A greatly enhanced intrusion protection system and method can be achieved by combining the operating systems of Figures 1-6. Now an arriving data packet would first be screened by a gateway router or a similar device for a correct destination address. If the destination address is correct, the packet is passed for further processing. If the destination address is incorrect, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

The packet with correct destination address is then screened for a correct source address. If the source address is correct, the packet is passed to the receiver computer. If the source address is incorrect, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

Then, the packet with a correct destination address and a correct source address is screened for a correct allowed port coordinate such as port number. If the port coordinate is correct, the packet is passed for further processing. If the port coordinate is incorrect, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

Finally, the packet with a correct destination and source addresses and a correct port designator is screened for data integrity by application of authentication check such as a checksum. If the authentication check is passed, the packet is passed to the addressee computer. If the authentication check is failed, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

The security managing unit analyses all the alarms and makes decisions on necessary unscheduled changes of addresses for appropriate network servers. Also, it can notify law enforcement and pass appropriate data on to it.

Figure 7 illustrates an enhanced computer intrusion protection system indicated generally at 52 for one or more network computers 54. A gateway router or a bridge 58 includes a destination address filter 60 which receives data packets which pass in through a load distribution switch 62. A non-interrogatable network address book 64 stores current network server addresses for the destination address filter 60, and the destination address filter checks each data packet to determine if a legitimate destination address is present.

Packets with legitimate destination addresses are forwarded to a source address filter 66, while packets with illegitimate destination addresses are sent to a security analysis section 68 in a management unit 70.

When a preset traffic load level is reached indicating that an attempt at flooding is being made, the destination address filter causes the load distribution switch 62 to distribute traffic to one or more parallel gateway routers or bridges which collectively forward legitimate traffic and dump the flooding traffic. An alternative arrangement would call for the load distribution function to be done irrespective of the load, utilizing all the parallel

The packet with correct destination address is then screened for a correct source address. If the source address is correct, the packet is passed to the receiver computer. If the source address is incorrect, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

Then, the packet with a correct destination address and a correct source address is screened for a correct allowed port coordinate such as port number. If the port coordinate is correct, the packet is passed for further processing. If the port coordinate is incorrect, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

Finally, the packet with a correct destination and source addresses and a correct port designator is screened for data integrity by application of authentication check such as a checksum. If the authentication check is passed, the packet is passed to the addressee computer. If the authentication check is failed, the alarm is triggered and the packet is passed to the network security managing unit for security analysis.

The security managing unit analyses all the alarms and makes decisions on necessary unscheduled changes of addresses for appropriate network servers. Also, it can notify law enforcement and pass appropriate data on to it.

Figure 7 illustrates an enhanced computer intrusion protection system indicated generally at 52 for one or more network computers 54. A gateway router or a bridge 58 includes a destination address filter 60 which receives data packets which pass in through a load distribution switch 62. A non-interrogatable network address book 64 stores current network server addresses for the destination address filter 60, and the destination address filter checks each data packet to determine if a legitimate destination address is present.

Packets with legitimate destination addresses are forwarded to a source address filter 66, while packets with illegitimate destination addresses are sent to a security analysis section 68 in a management unit 70.

When a preset traffic load level is reached indicating that an attempt at flooding is being made, the destination address filter causes the load distribution switch 62 to distribute traffic to one or more parallel gateway routers or bridges which collectively forward legitimate traffic and dump the flooding traffic. An alternative arrangement would call for the load distribution function to be done irrespective of the load, utilizing all the parallel gateways all the time. A source address table 74 stores accessible server's designators and corresponding current addresses for all system servers which may legitimately have access to the computer or computers 54. These addresses are accessed by the source address filter which determines whether or not an incoming data packet with the proper destination address originates from a source with a legitimate source address entered in the source address table 74. If the source address is determined to be legitimate, the data packet is passed to a port address filter 76. Data packets with an illegitimate source address are directed to the security analysis section 68. Alternatively, source address screening can be done at the gateway router or bridge 58 first prior to port filter 76.

A port protection table 78 includes the current port assignments for the computer or computers 54, and these port assignments are accessed by the port designator filter 76 which then determines if an incoming data packet contains legitimate port designation. If it does, it is passed to an actual address translator 80 which forwards the data packet to the specific computer or computers 54 which are to receive the packet. If an illegitimate port address is found by the port address filter 76, the data packet is transmitted to the security analysis section 68.

The management unit 70 is under the control of a security administrator 82. A network membership master file 84 stores a master list of legitimate server's designators along with respective authorized access lists and corresponding current cyber coordinates. The security administrator can update the master list by adding or removing authorized access for every protected computer. An access authorization unit 86 distributes the upgraded relevant portions of the master lists to the address books of the respective authorized servers.

A random character generator 88 generates random characters for use in forming current port designators, and provides these characters to a port designator forming block 90. This port designator forming block forms the next set of network current port designators in conjunction with the master list and these are incorporated for transmission by a port table block 92. Alternatively, port designators can be formed in the computer unit instead of the management unit.

Similarly, a random character generator 94 generates random characters for use in forming current server addresses, and provides these characters to a server address forming block 96. This server address forming block forms the next set of current network server addresses, and an address table 98 assigns addresses to servers designated on the master list.

A coordinator/dispatcher block 100 coordinates scheduled move of network servers to their next current addresses, provides the next set of network addresses for appropriate servers and routers and coordinates unscheduled changes of addresses on command from the security analysis unit 68. The coordinator/dispatcher block 100 may be connected to an encode/decode block 102 which decodes received address book upgrades from input 104 and encodes new port and server destination addresses to be sent to authorized servers in the system over output 106. Where encoding of new cyber coordinates is used, each authorized computer in the network will have a similar encoding/decoding unit.

The security analysis unit 68 analyses received illegitimate data packets and detects attack attempts. If needed, the security analysis unit orders the coordinator/dispatcher block 100 to provide an unscheduled address change and diverts the attack data packets to an investigation unit 108. This investigation unit simulates the target server keeping a dialog alive with the attacker to permit security personnel to engage and follow the progress of the attacker while tracing the origin of the attack.

Providing security against intrusion for e-commerce systems presents a unique problem, for an important peculiarity of an e-commerce system is that its address must be publicly known. This aspect represents a contradiction to the requirement of the address being known to authorized parties only. However, the only information intended for the general public usually relates to a company catalog and similar material. The rest of the information on a merchant's network is usually considered private and thus should be protected. Using this distinction, a merchant's e-commerce site should be split into two parts: public and private. The public part is set up on a public "catalog" server with a fixed IP address and should contain only information intended for the general public. The rest of the corporate information should be placed in a separate network and protected as described in relation to Figures 1-7.

When a customer has completed shopping and made purchasing decisions concerning the terms and price of the sale, pertinent for the transaction, information is placed in a separate register. This register is periodically swept by a server handling financial transactions ("financial" server), which belongs to the protected corporate network. In fact, the "catalog" server does not know the current address of the financial transactions server. Thus, even if an intruder penetrates the "catalog" server, the damage is limited to the contents of the catalog and the intruder cannot get an entry to the protected corporate network.

The financial server, having received pending transaction data, contacts the customer, offering a short-term temporary access for finalizing the transaction. In other words, the customer is allowed access just long enough to communicate pertinent financial data such as a credit card number and to receive a transaction confirmation at which point the session is terminated, the customer is diverted back to the catalog server and the financial server is moved to a new cyber address thus making obtained knowledge of its location during the transaction obsolete.

Dial-up communications systems, in respect to their infrastructure channels susceptibility to transmission intercept by unrelated parties, can be separated into two broad categories: easily interceptable, such as cellular and satellite telephone systems and relatively protected such as conventional land-line based telephone systems. Relatively protected systems such as conventional land-line based telephone systems can be protected in the following way. Phone numbers, assigned by a telephone company to a dial-up telephone-based private network serve as the members' computer addresses. As described previously, such a private network can be protected from unauthorized remote access by implementing periodic changes in the addresses, i.e. telephone numbers assigned to the members for transmission by the network along with other designators such as access codes and communicating the changed numbers to the appropriate parties.

For the conventional land-line dial-up telephone systems, while the "last mile" connection remains constant, the assigned telephone number is periodically changed, making the corresponding computer a moving target for a potential attacker. In this case the telephone company serves as the security system manager. It assigns the current variable telephone numbers to the members of a protected, private network, performs notification of all the appropriate parties, and changes the members' current numbers to a new set at an appropriate time. The telephone company switches naturally serve in the role of routers, and thus they can be programmed to perform surveillance of the system, to detect potential intrusion attacks and to issue appropriate alarms.

Periodically changing the current assigned numbers creates system entropy for a potential intruder, making unauthorized access difficult. Obviously, the implementation of this security system is dependent on availability of sufficient vacant numbers at a particular facility of the telephone company. Furthermore, for a variety of practical reasons it is advisable to keep a just vacated number unassigned for a certain period of time. All this may require additional number capacity at the telephone company facility in order to enable it to provide remote access security to a larger number of personal networks while preserving a comfortable level of system entropy.

If the mentioned additional capacity is not available, or a still higher level of entropy is desired, it could be artificially increased by adding an access code to the assigned number. This would amount to adding virtual capacity to the system, and would make a combination of the phone number and access code an equivalent of a computer's telephone address. In effect, this would make a dialed number larger than the conventional format. This method makes a virtual number capacity practically unlimited and, since the process is handled by computers without human involvement, it should not put any additional burden on a user. With or without a virtual number capacity, utilization of this method allows the intrusion attempts to be easily identified by their wrong number and/or code. At the same time, implementation of this system might require some changes in dialing protocols as well as additional capabilities of the telephone switching equipment.

Entropy density can be increased by limiting the number of allowable connection attempts. Similarly to the method described previously, telephone company switching equipment can be made to perform a role of an outside security barrier for the private network. In this case wrongly addressed connection attempts should be analyzed in order to detect possible "sweeping". If such an attempt is detected, tracing the origin of the attempt and notifying the appropriate phone company should not present a problem even with the existing technology.

The simplest form of private network protection under the proposed method and system is when at a predetermined time all the members of a particular network are switched to the new "telephone book" of the network. However, in some cases required level of security for some members of the same private network could substantially differ, or they may face different levels of security risk. In such cases frequency of the phone number change could be set individually with appropriate notification of the other members of the network. This differentiation enables the telephone company to offer differentiated levels of security protection to its customers even within the same private network.

A telephone company can also offer its customers protected voice private networks which would provide a higher level of privacy protection than the presently used "unlisted numbers." In this configuration the customers' telephone sets are equipped with a computerized dialing device with remotely upgradeable memory which would allow each member of a protected voice network to contain the network "telephone book" and that book is periodically updated by the telephone company.

The telephone company would periodically change thc assigned telephone numbers of a protected network to a new set of current numbers. These new numbers would be communicated to the members of a protected voice network through updating their computerized dialing devices.

As a derivative of the described system, an updateable electronic telephone directory system can be also implemented. In this case a customer's phone set would include a computerized dialing device with electronic memory containing a conventional telephone directory and a personal directory as well. This telephone directory can be periodically updated on-line by the telephone company.

Easily interceptable systems such as cellular and satellite telephone systems, in addition to the protection described above, can be protected from "cloning" when their signals can be intercepted and the "identity" of the phone can be cloned for gaining unauthorized access and use of the system by unauthorized parties.

Mobile telephone and mobile communications systems are protected in a manner similar to networks or land based telephone systems. In this instance, the novel and improved method of changing cyber coordinates is designed to reliably protect mobile phone systems from unauthorized use commonly known as cloning as well as to make intercept of wireless communications more difficult than it is at present. With this system the static wireless phone number or other similar identifier is not used for identification and authorization. Instead, a set of private identifiers is generated known only to the phone company and base stations controlling mobile phone calls and used to continually update the mobile phone and base station directories with current valid identifiers. This approach provides vastly superior protection over current methods requiring that each call be intercepted in order to track and keep current with changing identifiers. Immediate detection of unauthorized attempts to use a cloned phone is realized and law enforcement may be notified in near real time for appropriate action.

Other electronic devices using wireless communications can be protected by the methods and systems described above.

Finally, computers often contain databases with a variety of information. That information in a database often has wide-ranging levels of sensitivity or commercial value. This creates a situation when large computers serve multiple users with vastly different levels of access. Furthermore, even within the same level of access, security considerations require compartmentalization of information when each user has to have access to only a small portion of the database.

The existing systems try to solve this situation by utilizing passwords and internal firewalls. As it was mentioned earlier, password-based systems and firewalls are not sufficient against computerized attacks. In practical terms it means that a legitimate user with a low level of access, utilizing hacking techniques from his station, potentially can break into even the most restricted areas of the database.

This problem can be solved by using the method of the present invention. A piece of information such as a file or a directory in a computer exists in cyber space. Accordingly, it has its cyber address, usually expressed as a directory and/or a file name which defines its position in a particular computer file system. This, in effect, represents the cyber coordinates of that piece of information within a computer.

As described earlier, information security can be provided if a system manager periodically changes the directories and/or file names in the system, i.e. the cyber addresses of the information, and notifies only appropriate parties of the current file names. This method would ensure that each user computer knows locations of only files to which it has legitimate access. Furthermore, a user would not even know of existence of the files to which he has no access.

To further strengthen the system and make it user-friendly, the user would have a personal directory similar to an address book, where only permanent directory and/or file names are accessible to him, while the variable side of the "address book" would be accessible only to the system manager and upgraded periodically. In this arrangement variable directory and/or file names can contain any required level of entropy, further increasing resistance to attacks from within the system. Additionally, an internal "router" or "filter" can also perform information security monitoring functions, detect intrusion attempts and issue appropriate alarms in real time.

Obviously, in order to ensure information security in such arrangement any computer-wide search by keywords or subject should be disabled and substituted with a search within specific clients' "address books".

The systems and methods described above allow for creation of a feasible infrastructure protection system such as a national or international infrastructure protection system. When detected at specific points cyber attacks are referred to such a system for further analysis and a possible action by law enforcement authorities.

I claim:

1. A method for protecting a communications device which is connected to a communications system against an unauthorized intrusion which includes:

providing the communications device with at least one identifier,

providing the at least one identifier for use in accessing the communications device to entities authorized to access said communications device,

sensing the presence or absence of said identifier before granting access to said communications device,

providing access to said communications device when the use of said at least one correct identifier is sensed

denying access to said communications device and providing said communications device with at least one new identifier when the absence of the correct at least one identifier is sensed during an attempt to access said communications device, and providing said at least one new identifier to entities authorized to access said communications device.

2. The method of claim 1 which includes periodically changing the at least one identifier and providing the changed at least one identifier to the entities authorized to access said communications device.

3. The method of claim 1 which includes providing said communications device with a plurality of separate identifiers,

sensing the presence or absence of all of said plurality of identifiers before granting access to said communications device,

providing access to said communications device when the use of all of said identifiers is sensed, and

denying access to said communications device and providing said communications device with a new plurality of identifiers to replace the previous plurality of identifiers when the absence of any one of the correct identifiers is sensed.

4. The method of claim 3 which includes periodically changing said plurality of

separate identifiers and providing the changed identifiers to the entities authorized to access said communications device.

5. The method of claim 1 which includes permitting a predetermined number of attempts to access said communications device with a correct at least one identifier after the absence of the correct at least one identifier is sensed before providing said communications device with at least one new identifier,

and providing access to said communications device if the correct at least one identifier is sensed during the predetermined number of attempts to access.

6. The method of claim 2 wherein said communications system is a telephone system and said communications device is a telephone.

7. The method of claim 1 wherein said communications system is a computer network with said entities authorized to access said communications device being authorized computers having access to said computer network, said communications device including at least one host computer having access to said computer network.

8. The method of claim 7 which includes periodically changing the at least one identifier for the host computer and providing the changed at least one identifier to the authorized computers.

9. The method of claim 7 which includes providing the authorized computers with an unchangeable, accessible address for the host computer which is used by the authorized computer to activate and transmit the at least one identifier for the host computer when the authorized computer initiates access to the host computer.

10. The method of claim 8 which includes providing each authorized computer with an authorized computer identifier,

providing the host computer with a destination identifier,

causing each authorized computer to access said host computer with at least a host computer destination identifier and the authorized computer identifier,

sensing the presence or absence of both said host computer destination identifier and an authorized computer identifier before granting access to said host computer,

providing access to said host computer when the use of both a correct host computer destination identifier and an authorized computer identifier is sensed, and

denying access to said host computer and providing said host computer with a new host computer destination identifier when the absence of either a correct host computer destination identifier or a correct authorized computer identifier is sensed.

11. The method of claim 10 which includes permitting a predetermined number of attempts to access said host computer with both a correct host computer destination identifier and an authorized computer identifier after the absence of a correct host computer destination identifier or an authorized computer identifier is sensed before providing said host computer with a new host computer destination identifier, and

providing access to said host computer if correct host computer destination and authorized computer identifier are sensed during the predetermined number of attempts to access the host computer.

12. The method of claim 11 which includes storing said host computer destination identifier as an inaccessible identifier in said authorized computers, and providing said authorized computers with an unchangeable, accessible host computer address, which will activate and transmit the host computer destination identifier when an authorized computer initiates access to the host computer.

13. The method of claim 8 which includes providing said host computer with a host computer destination identifier and a host computer port identifier,

causing each authorized computer to access said host computer with at least the host computer destination identifier and the host computer port identifier,

sensing the presence or absence of both said host computer destination identifier and said host computer port identifier before granting access to said host computer,

providing access to said host computer when the use of both a correct host computer destination identifier and a correct host computer port identifier are sensed, and

denying access to said host computer and providing said host computer with a new destination identifier and port identifier when the absence of either or both of a correct host computer destination or port identifier is sensed.

14. The method of claim 13 which includes permitting a predetermined number of attempts to access said host computer with both a correct host computer destination and port identifier when either or both an incorrect host computer destination or port identifier is sensed before providing said host computer with a new destination and port identifier, and

providing access to said host computer if both correct host computer destination and port identifiers are sensed during the predetermined number of attempts to access said host computer.

15. The method of claim 14 which includes storing said host computer destination and port identifiers as inaccessible identifiers in said authorized computers and providing said authorized computers with an unchangeable, accessible host computer address which will activate and transmit the host computer destination and port identifiers when an authorized computer initiates access to said host computer.

16. An intrusion protection method for protecting a host computer connected to a computer communications system which includes one or more authorized computers having access to said computer communications system which are authorized to access said host computer which includes:

providing each authorized computer with an authorized computer identifying address,

providing said host computer with a host computer destination identifier and a host computer port identifier,

providing said host computer destination identifier and said host computer port identifier to said authorized computers,

causing each authorized computer to access said host computer with the host computer destination and port identifiers and said authorized computer identifying address,

sensing the presence or absence of said host computer destination and port identifiers and said authorized computer identifying address before granting access to said host computer,

providing access to said host computer when the use of correct computer destination and port identifiers and a correct authorized computer identifying address is sensed, and

denying immediate access to said host computer when the absence of any one or more of the correct host computer destination and port identifiers or the authorized computer identifying address is sensed.

17. The method of claim 16 which includes periodically changing the host computer destination and port identifiers and providing these changes to the authorized computers.

18. The method of claim 17 which includes storing said host computer destination and port identifiers as inaccessible identifiers in said authorized computer and providing said authorized computers with an unchangeable, accessible host computer address which will activate and transmit the host computer destination and port identifiers when an authorized computer initiates access to said host computer.

19. The method of claim 16 which includes changing the host computer destination and port identifiers when access is denied to said host computer after at least one access attempt has been made and providing these changed identifiers to the authorized computers.

20. The method of claim 16 which includes permitting a predetermined number of attempts to access said host computer with correct host computer destination and port identifiers and a correct authorized computer identifying address after the absence of at least a correct one of said identifiers and authorized computer identifying address is sensed by the host computer and

providing access to said host computer if correct host computer destination and port identifiers and a correct authorized computer identifying address are sensed during the predetermined number of attempts to access said host computer.

21. The method of claim 19 which includes storing said host computer destination and port identifiers as inaccessible identifiers in said authorized computer and providing said authorized computers with an unchangeable, accessible host computer address which will activate and cause transmission of the host computer destination and port identifiers when an authorized computer initiates access to said host computer.

22. The method of claim 20 which includes changing the host computer destination and port identifiers when access is denied to said host computer after at least one access attempt has been made and providing these changed identifiers to the authorized computers.

23. The method of claim 22 which includes storing said host computer destination and port identifiers as inaccessible identifiers in said authorized computer and providing said authorized computers with an unchangeable, accessible host computer address which will activate and cause transmission of the host computer destination and port identifiers when an authorized computer initiates access to said host computer.

24. A method of communication with a remote entity over a communication system which includes

providing the remote entity with at least one remote entity cyber coordinate identifier,

providing the remote entity cyber coordinate identifier to one or more base entities authorized to communicate with said remote entity,

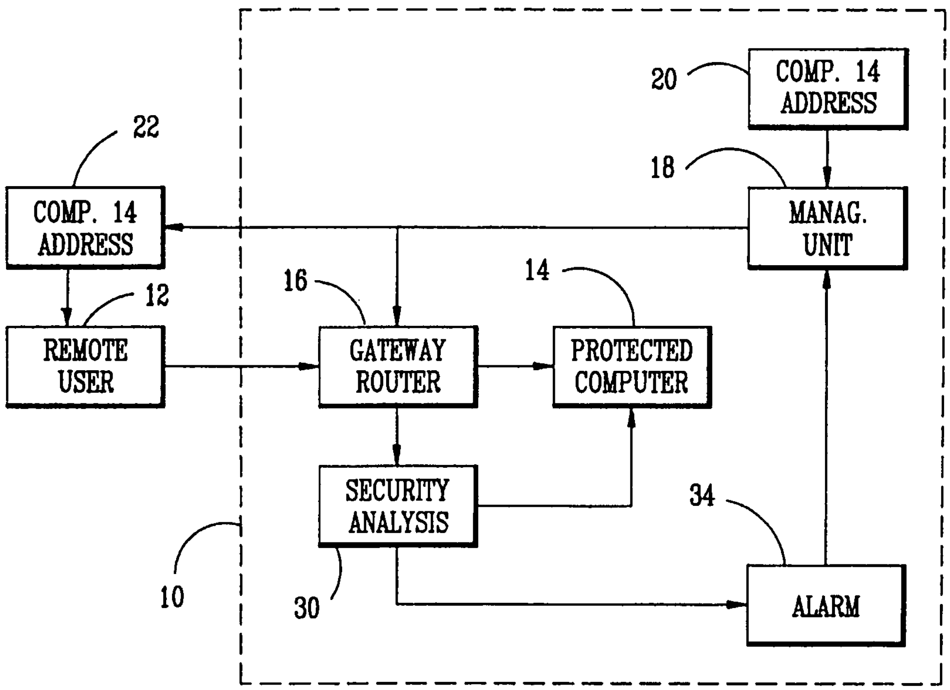
periodically changing the remote entity cyber coordinate identifier to a new remote entity cyber coordinate identifier and

providing the new remote entity cyber coordinate identifier to said one or more base entities.

25. The method of claim 24 which includes changing the remote entity cyber coordinate identifier to a new cyber coordinate identifier in response to an attempt to communicate with said remote entity with an incorrect remote entity cyber coordinate identifier and

providing the new remote entity cyber coordinate identifier to said one or more base entities.

FIG. 1



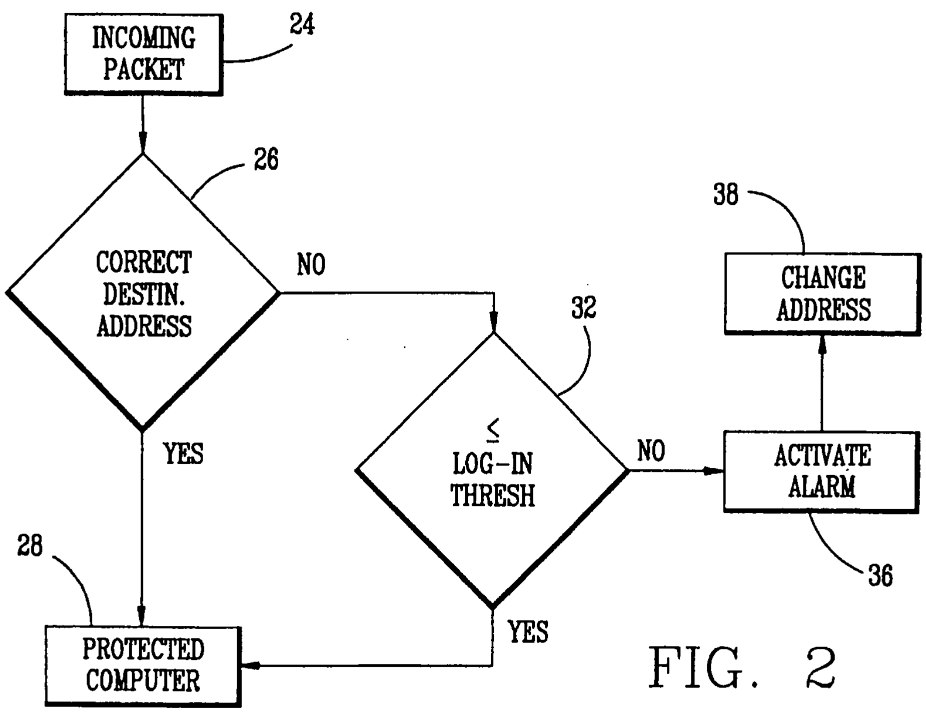


FIG. 2

SUBSTITUTE SHEET (RULE 26)

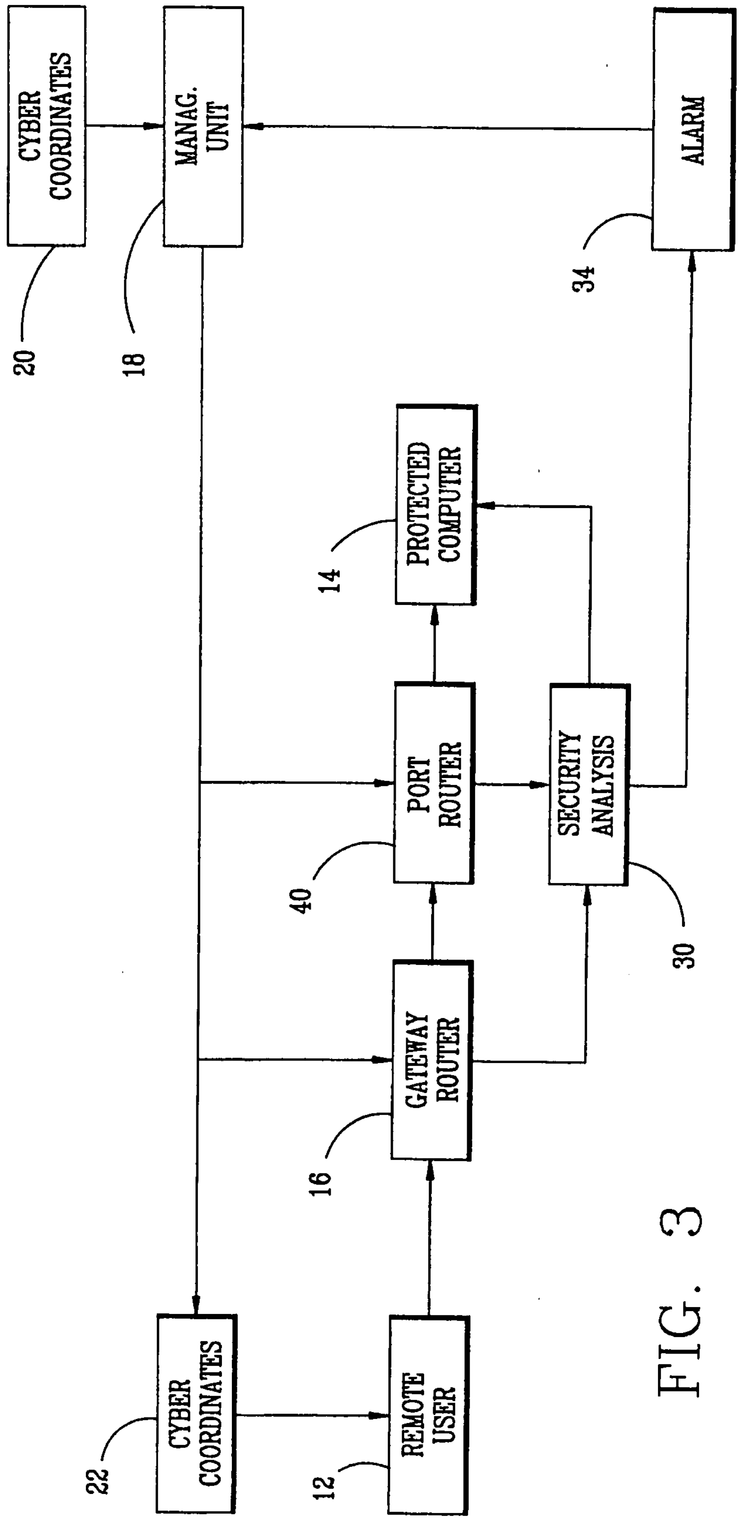

CYBER COORDINATES
MANAG. UNIT
ALARM
PROTECTED COMPUTER
PORT ROUTER
SECURITY ANALYSIS
GATEWAY ROUTER
CYBER COORDINATES
REMOTE USER
20
18
34
14
40
30
16
22
12

FIG. 3

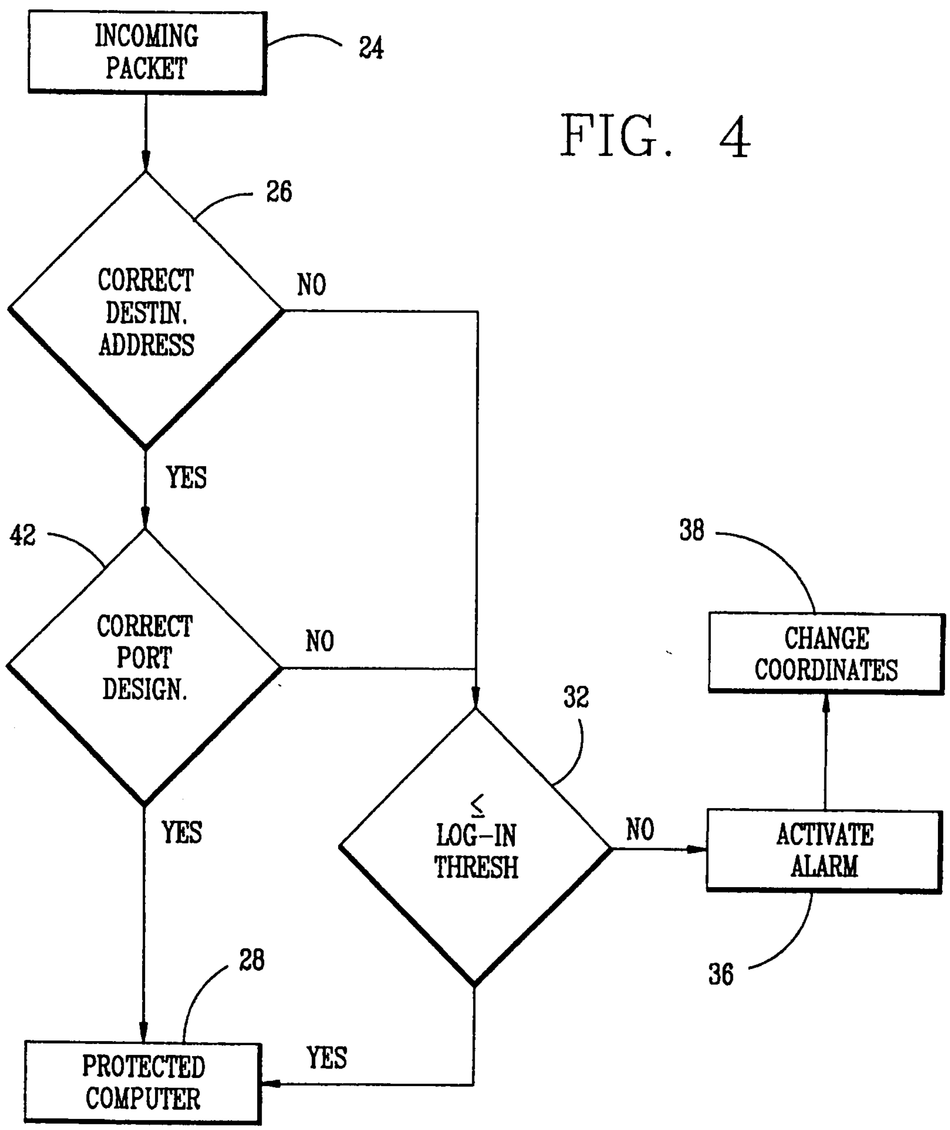

INCOMING PACKET
24
26
CORRECT DESTIN. ADDRESS
NO
YES
42
CORRECT PORT DESIGN.
NO
38
CHANGE COORDINATES
32
YES
≤ LOG-IN THRESH
NO
ACTIVATE ALARM
28
36
PROTECTED COMPUTER
YES

FIG. 4

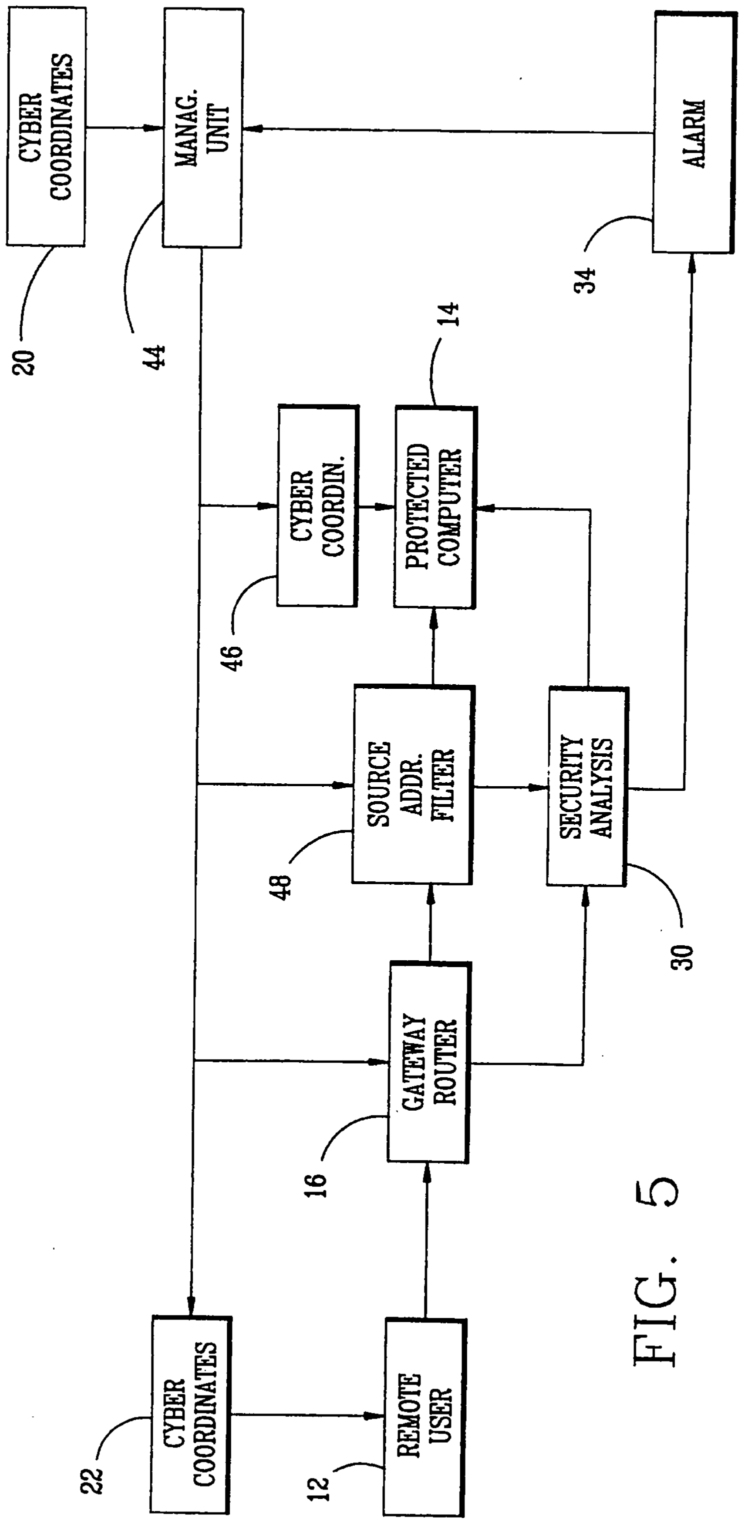


FIG. 5

SUBSTITUTE SHEET (RULE 26)

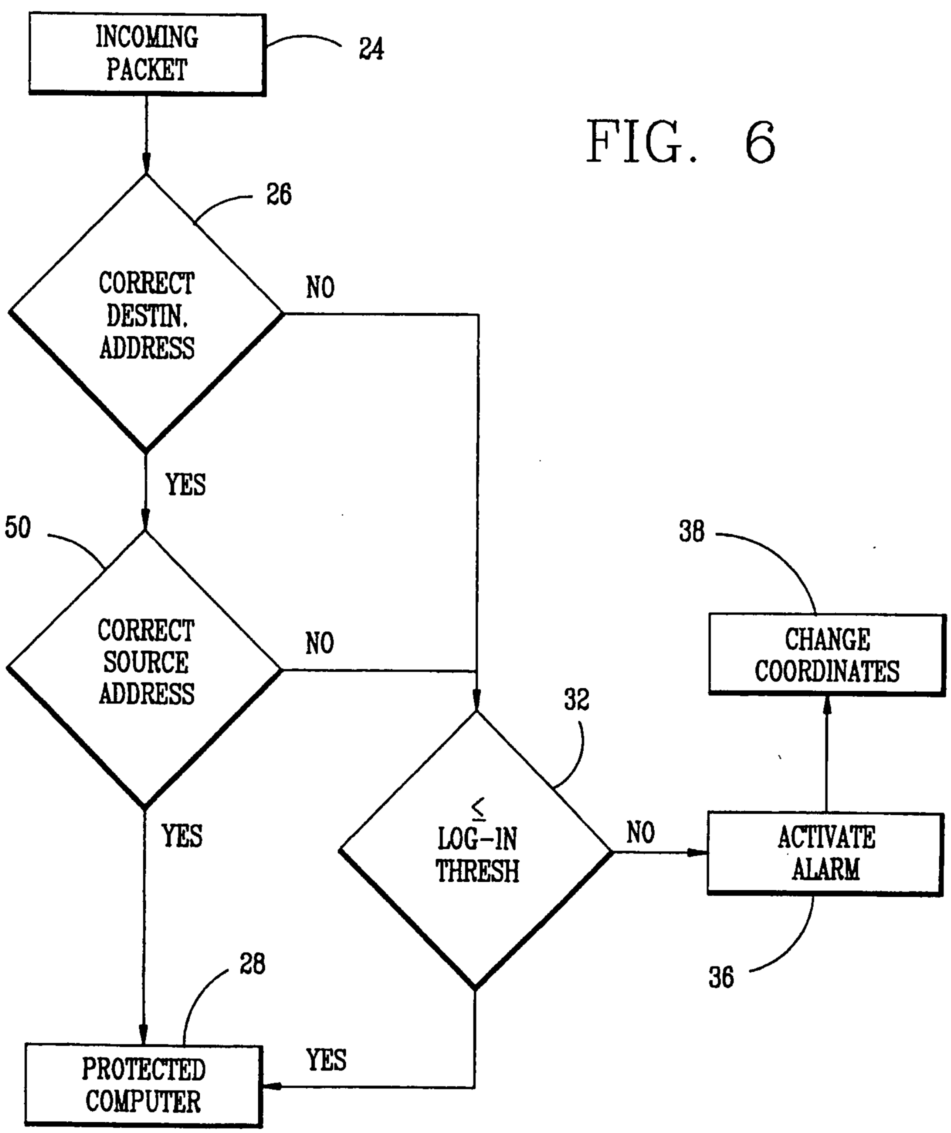
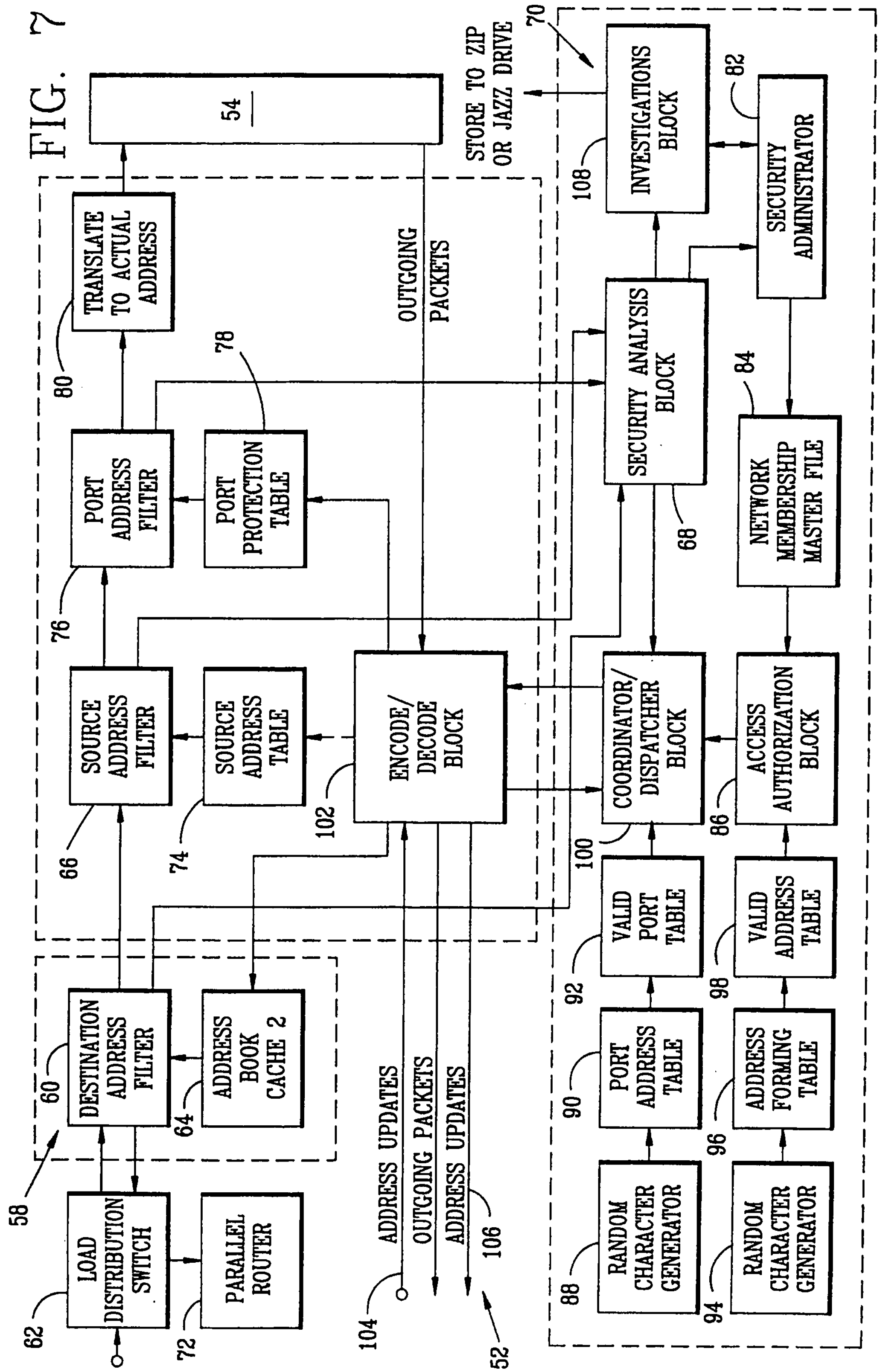


FIG. 6

SUBSTITUTE SHEET (RULE 26)



SUBSTITUTE SHEET (RULE 26)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/08219

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 11/00
US CL : 713/201
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 713/201,200,202; 340/825.31,825.34; 380/255;

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS US PATENT FILE; WEST; JPAB; EPAB; DWPI; TDBD;

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5,805,801 A (HOLLOWAY ET AL) 08 SEPTEMBER 1998, Entire document.	1-25
Y	US 5,796,942 A (ESBENSEN) 18 AUGUST 1998, Entire document.	1-25
Y,P	US 5,905,859 A (HOLLOWAY ET AL) 18 MAY 1999, Entire document.	1-25
Y	US 5,892,903 A (KLAUS) 06 APRIL 1999, Entire document.	1-25
A	US 5,537,099 A (LIANG) 16 JULY 1996, Entire document.	1-25
A	US 5,278,901 A (SHIEH ET AL) 11 JANUARY 1994, Entire document.	1-25

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 JULY 2000	22 AUG 2000
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer NADEEM IQBAL
Facsimile No. (703) 305-3230	Telephone No. (703) 308-5228

Form PCT/ISA/210 (second sheet) (July 1998)★

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US00/08219

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 5,991,881 A (CONKLIN ET AL) 23 NOVEMBER 1999, Entire document.	1-25

Form PCT/ISA/210 (continuation of second sheet) (July 1998)*

B6

PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 12/14, 1/00	A1	(11) International Publication Number: WO 98/55930 (43) International Publication Date: 10 December 1998 (10.12.98)

(21) International Application Number: PCT/IB98/00578

(22) International Filing Date: 20 April 1998 (20.04.98)

(30) Priority Data:
97201667.9 3 June 1997 (03.06.97) EP
(34) Countries for which the regional or international application was filed: NL et al.

(71) Applicant: KONINKLIJKE PHILIPS ELECTRONICS N.V. [NL/NL]; Groenewoudseweg 1, NL-5621 BA Eindhoven (NL).

(71) Applicant *(for SE only)*: PHILIPS AB [SE/SE]; Kottbygatan 7, Kista, S-164 85 Stockholm (SE).

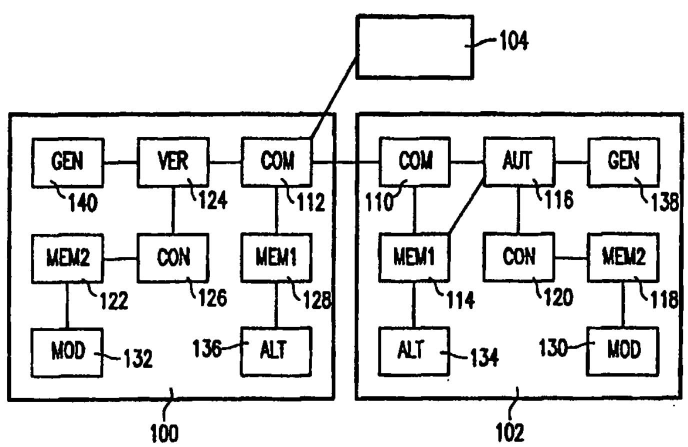
(72) Inventor: TANG, Wei-Jin; Prof. Holstlaan 6, NL-5656 AA Eindhoven (NL).

(74) Agent: GROENENDAAL, Antonius, W., M.; Internationaal Octrooibureau B.V., P.O. Box 220, NL-5600 AE Eindhoven (NL).

(81) Designated States: JP, European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: AUTHENTICATION SYSTEM



(57) Abstract

The authentication system comprises at least one station (102) and a host (104). The station (102) comprises a memory (118) for electronically storing a plurality of authentication items. Constructing means (120) are used to construct an authentication control element, such as a key, from a part of the authentication items which is selected for each message. Authentication means (116) authenticate the message substantially uniquely under control of the authentication control element constructed for the message. The authenticated message is sent to the host. The host comprises a memory (122) for electronically storing the authentication items of the station. The host comprises constructing means (126) for constructing for each received authenticated message an authentication control element in the same way as the station. Verification means (124) are used for, under control of the authentication control element, verifying the authenticity of the received message.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

Authentication system.

The invention relates to an authentication system comprising at least one station and a host; the station comprising: authentication means for, based upon an authentication algorithm, authenticating a message; and communication means for sending the authenticated message to the host; the host comprising: communication means for receiving an authenticated message; and verification means for verifying the authenticity of the received message by checking the received message with an authentication algorithm corresponding to a station which sent the message.

With the increase of electronic communication and electronic financial transactions, identification and authentication has become an essential aspect of many systems. Normally in an authenticated transaction three parties are involved: a host, a station and a user of the station. The host may, for example, be a central computer at a bank, at a retailer, or at a company providing services via Internet, or be a file server. The station may be a personal computer (PC), a Personal Digital Assistant (PDA) or a hand-held PC (HPC), usually connected or connectable via telecommunications to the host computer. The message may be a digital representation of a user generated message, including an instruction to a bank, but may also be computer data or computer code, such as a Java applet. In many applications, the station is split into two parts: a user station and an access station.

An identification, such as a communication address, which uniquely identifies the station is stored in the memory of the station. A message generated in the station, usually at the request of the user, is authenticated using an authentication algorithm. Typically, the message is authenticated by generating an additional digital signature. The authenticated message is sent to the host together with the identification of the station. The host uses the same or a complementary authentication algorithm to verify the authenticity of the message.

For certain applications, like a user instructing a bank to transfer money from a bank account, it may be required that the station performs some form of access control ensuring that only an authorised user can issue the instruction. The access control may, for instance, be based on a PIN-code or password. Also more advanced methods, for instance based on biometrical information, may be used. The access information may be passed on to the host as part of the message. For other applications, like a transfer of a small amount of electronic money, it may not be required or, in view of privacy or safety, even be undesired that additional access control is performed or that the access information is transferred to the host. The access control is not part of the invention.

Most authentication algorithms are based on encryption algorithms, such as the symmetrical DES algorithm or the asymmetrical public-key RSA algorithm. Typically, the same algorithm is used for each station and a dedicated key is used to make the algorithm act in a manner specific for the station. The security provided by such algorithms is mainly based in the algorithmic strength of the involved algorithms, which are, as a consequence, complicated and costly to implement, which is a particular drawback for simple consumer electronic products.

It is an object of the invention to provide an authentication system of the kind set forth, which is simple to develop. It is a further object to provide such a system which can be cost-effectively implemented in consumer electronic products. It is a further object to provide such an authentication system which offers a high level of security.

To achieve this object, the authentication system according to the invention is characterised in that the station comprises a memory for electronically storing a plurality of authentication items; the host comprises a memory for electronically storing the authentication items of the station in association with an identification of the station;

the station comprises constructing means for constructing for each message a corresponding authentication control element; the constructing means being operable to select for the message a part of the plurality of authentication items and to construct the authentication control element from the selected part, where the authentication control element in practical circumstances causes the authentication algorithm to substantially authenticate the corresponding message uniquely; and

the host comprises constructing means for constructing for each received authenticated message an authentication control element from the authentication items associated with a station which sent the message; the construction being the same as performed by the associated station.

The system according to the invention is based on the insight that the simple manner in which parents and children identify each other when they are not in direct contact, such as in the case of a kidnapping, can form the basis of an automatic authentication system. If for instance a child is kidnapped, the parents want to be sure that the kidnappers indeed hold the child and that, for instance, a ransom demand genuinely relates to their child. At the moment when identification of, for instance, the child is required, the child informs the kidnappers of a few events from a large set of events known to the child and parents and unknown to others (or at least to the kidnappers). For each communication with the kidnappers, the parents may request that the child recalls other events. This ensures that the kidnappers have to keep the child alive. It also ensures that no fraudulent kidnappers, who in one way or another intercepted a set of identifying events, can re-use this set for authenticating a fraudulent demand.

Based on this insight, the host (parent) and the station (child) share a large set of authentication items. For each message which needs to be authenticated, a small sub-set from the authentication items is selected and used to form an authentication control element which controls an authentication algorithm. In practical circumstances the authentication algorithm authenticates with a high likelihood the corresponding message uniquely under control of the authentication control element. A main strength of the system according to the invention lies in unpredictably authenticating messages by selecting a subset of authentication items from a relatively large set, where for each next message other items may be selected. This allows the use of a simple authentication algorithm, where the emphasis is not on the algorithmic strength of the algorithm, such as the difficulty of predicting for a message the corresponding authenticated message, but on using the algorithm in an unpredictable manner. A correlation which might occur in the authentications generated for successive messages can be broken by using an authentication control element, which is not related to the authentication algorithm. The authentication items, which determine the authentication control element can be generated in advance using sophisticated means, such as real random sequence generators, if desired. For a fraudulent party to be able to break the system, the fraudulent party needs not only to intercept sufficient messages to be able to break the authentication algorithm but also to determine the entire set of authentication items. The size of the set of authentication items and the size of the subset used to generate an authentication control element can be chosen to optimally suit the application in which the system is used. As an example, for a not very demanding application, a set of authentication items formed by a couple of hundred random bytes may be used, where the authentication algorithm may be based on a substitution, using a substitution table. Some or all elements of the substitution table which have been used during the substitution are replaced by new elements derived from the random bytes. These new elements form the authentication control element. The new elements may, for instance, be selected using a (pseudo-)random number generator. For more demanding applications, more authentication items may be used. If desired, also the complexity of the involved algorithm may be increased, for instance by basing the authentication algorithm on algorithmically strong encryption algorithms, such as DES, where the authentication control element forms a key for DES. For applications which require a high level of security, the authentication items and the algorithms are preferably stored in, respectively, executed in a secure module, such as a tamper-proof IC.

It should be noted that the Dutch Giro (Postbank) uses the TAN (Transaction Number) system for electronic payments by customers using a PC and a modem. The customers of the Postbank receive via regular mail several transaction numbers printed on a piece of paper. For each transaction the client has to enter a next transaction number until all numbers have been used, at which moment the client receives a new set of numbers. A fraudulent party has, in general, easy access to the transaction numbers at the customers premises. Furthermore, the distribution of the transaction numbers from the host to the customer makes the system vulnerable for fraudulent parties intercepting the list.

For simple systems, for instance used to check the authenticity of an entry ticket to a sporting event or concert, it may be sufficient to differentiate between authentic and non-authentic stations. The station, such as an electronic ticket, may be re-used for authenticating a series of events by using an event-specific message. For a more demanding system, such as involving financial transactions, a message is advantageously authenticated in a manner unique for the station.

The measure as defined in the dependent claim 2 has the advantage that the uniqueness of the station identification is used for authenticating a message in a manner unique for the station. The station identification, which is used to distinguish the station amongst the other stations of the system with respect to the host, may, for instance, be a communication address or an account number.

The measure as defined in the dependent claim 3 has the advantage that a fraudulent party needs to intercept messages for each station in order to determine the authentication items specific for the station, making the task of the fraudulent party more complicated.

The measure as defined in the dependent claim 4 has the advantage that the set of shared information (the authentication item) is updated as the station and the host experience more shared events, like authenticating a message. In this way a fraudulent party not only has to determine the set of authentication items but also how the items are modified over time.

In a further embodiment of the system according to the invention, the system is characterised in that the modification means is operative to modify an authentication item at least partly based on an event independent of the authentication items. In this way it becomes even more important for a fraudulent party to intercept and record all messages in order to be able to determine the authentication items. In a simple form the modification may be (partly) based on the previously authenticated message(s). This has the advantage that the message is already transferred to the host allowing the host to perform the same modification. In some systems the content of the message may be too predictable to significantly increase the task of a fraudulent party in breaking the authentication. In such systems more unpredictable events, such as the time at which the last message was authenticated, may be used. To allow the host to perform the same modification, the event has to be informed to the host, for instance, in the form of a time-stamp. Preferably, the modification is at least partly based a random or pseudo-random event. Advantageously, the least-significant bits of a clock are used, giving for most systems a sufficiently random event, particularly if messages are authenticated individually and not processed in a sequential batch.

The measure as defined in the dependent claim 6 has the advantage that it becomes more difficult for a fraudulent party to collect messages relating to the same station. The identification may, for instance, be a conventional communication identification, such as a communication address or a telephone number. Instead of in addition to such a communication identification, the station identification may also be based on an identification, such as an indication of an account number, which is chosen independent of the communication identification. For such a combined identification only part of the identification, e.g. only the account number, may be modified, whereas the other part remains fixed. Preferably, the host locates information, such as the authentication items, used for verifying the authentication at least partially based on the variable part of the identification. As an example, the host may locate the relevant information for verifying a message in dependence on a bank account number. Instead of using the real bank account number as the identification (and exchanging the real bank account number), a virtual bank account number is used. The station and the host are initially loaded with the same virtual account number. The host also knows how to associate (map) the virtual number with the real bank account number. Usually, the station also knows the real bank account number for local operations, such as display to the user, and preferably hides the virtual number from the user. The virtual number is exchanged and not the real number. Only in exceptional cases, e.g. when the synchronisation in updating the virtual number is lost between the host and the station, it may be required to re-synchronise to a new virtual number using the real number for once as an identification. Both the station and the host can alter the virtual number in the same way, keeping the real underlying bank account number fixed (i.e. only the mapping between a variable virtual number and a fixed real number is changed). In such a scenario, the virtual number acts as the identification according to the invention. Particularly, for mobile stations, such as a PDA or a smart-card, with no fixed communication link to the host, it becomes practically impossible for a fraudulent party to collect messages related to a specific station or a specific application within the station, such as an application for financial transfers/information retrieval, downloading of software or playing of a network game, where each application uses an application-specific authentication algorithm or set of authentication items. This allows the use of less authentication items or a simpler authentication algorithm. It further allows to detect fraudulent messages in an early stage. As an example, in a system where no more than 65,000 stations need to distinguished (implying that in principle a two-byte identification would be sufficient) a larger identification of, for instance, four of six bytes may be used, where the identification is chosen dynamically. If a four-byte identification is used, the host can identify almost all received fraudulent messages as being fraudulent simply by checking the identification. Only for in average 1 out of 65,000 fraudulent messages the authentication of the message (which typically involves more processing) needs to be checked. This makes the system suitable for use in environments, such as Internet, where brute-force attacks by generating many different fraudulent messages may occur. Preferably, the response time of the host is similar regardless of the station identification being valid or not, ensuring that fraudulent parties can not distinguish between valid and invalid station identifications. Advantageously, the alteration means alters the station identification at least partly based on a message and/or a time-stamp. In this way it becomes even more important for a fraudulent party to intercept and record all messages in order to be able to determine the current station identification.

The measure as defined in the dependent claim 7 has the advantage that in a simple way it can be ensured that messages, even the same messages, are with a high likelihood authenticated differently. Furthermore, it limits the possibilities of a fraudulent party, including the legitimate owner or designer of the station, to generate known messages and from the corresponding authenticated message derive the authentication items of the station.

The measure as defined in the dependent claim 8 has the advantage that the host independently generates the same additional data, providing a further check of the authenticity of the message.

The measure as defined in the dependent claim 9 has the advantage that by incorporating the additional data into the message, for instance by mixing the additional data with the message, and authenticating the resulting message, the host only needs to verify the resulting message as before and can then discard the additional data, without being able to generate the additional data. Advantageously, each station generates the additional data in a manner unique for the station.

The measure as defined in the dependent claim 10 has the advantage that by using a state variable, such as a feedback state for a random number generator, the construction means can autonomously select different parts of the authentication items for a large sequence of messages. Preferably any periodicity in the selection is sufficiently large in view of the application. By ensuring that the selection also depends on the authentication items (for instance on a subset of the authentication items), which have been generated independent of the construction means, a correlation which might occur in successive selections can be reduced. As an example, the control vector could be one authentication item which is XOR-ed over the basic output, such as a random number, of the construction means. The control vector itself may be each time randomly selected from the set of authentication items.

These and other aspects of the invention will be apparent from and elucidated with reference to the embodiments shown in the drawings.

Fig. 1 shows a block diagram of a system according to the invention,
Fig 2 shows a flow-chart of a possible operation in the station 102, and
Fig. 3 shows a flow diagram of a basic operation which can be used in steps of Fig. 2.

Fig. 1 shows a block diagram of a system according to the invention. The authentication system comprises a host 100 and at least one station. As an example two stations 102 and 104 are shown. Further details of the stations will be given with reference to station 102 only. Typically, the host 100 is implemented on a computer suitable for acting as a server. The station may be implemented on, for instance, a Personal Computer (PC), a Personal Digital Assistant (PDA) or a hand-held PC (HPC). In an exemplary application, a user instructs a bank to perform a financial transaction. The user enters the instruction in his station. The station generates a corresponding message, authenticates the message and transfers the message to the host computer at the bank. The host checks the authentication and, if valid, proceeds with processing the instruction. The authentication serves various well-known purposes, such as reducing the chance of a third party, pretending to be another party, transmitting a message on behalf of the other party (the third party may have generated the fraudulent message or may be a re-transmitting an intercepted message which has been validly transmitted by the original party) and reducing the chance of the original party repudiating the message. The system may also be used for various other forms of electronic communication, such as for authenticating electronic mail, the exchange of electronic documents (such as an HTML document) or program modules (such as Java applets), or the communication between software objects located in different computers. If the receiving party can trust the sending party, the authentication ensures that the receiving party can safely use the received digital data, without having any risk of, for instance, having received virus-infected data or data which may adversely effect the local station (e.g. by discarding locally stored data). Particularly in situations where stations do not share prior knowledge and wish to safely communicate, the communication preferably takes place via a trusted party. With respect to the sending station the trusted party acts like a host according to the invention and the sending station acts like a station according to the invention. With respect to the receiving station the trusted party acts like a station according to the invention and the receiving station acts like a host according to the invention. The trusted party relays a message received from a sending station to a receiving station if the trusted party has successfully verified the authenticity of the received message, using a verification procedure matching the authentication procedure of the sending station. The trusted party authenticates the received message, using a procedure agreed with the receiving station, before transmitting the message to the receiving station.

It will be understood that, particularly for mobile applications, the station 102 may be split into, for instance, a user station and an access station. The access station establishes the communication with the host 100 and may, for instance, be fixedly located in a shop, a petrol station or integrated with an automatic teller machine of a bank. The access station may also be located at the premises of the user and, for instance, be integrated with a personal computer or audio/video set-top box. The user station ensures a station-specific authentication. The user station may, for instance, be formed by a PDA communicating via IrDA to another PDA or personal computer acting as an access station. As another example, the user station may be formed by a smart-card, where the access station includes a smart-card reader.

The station 102 and the host computer 100 can communicate digitally. To this end, the station comprises communication means 110 for digitally communicating with the host 100. Similarly, the host 100 comprises communication means 112 for digital communication with the station. Usually, the communication will take place using telecommunication, either wired or wireless. The communication means 110 and 112 may be formed by a conventional modem, operated under control of the processor of the station or, respectively, the host. The communication may also be based on local communication, such as a Local Area Network (LAN), infra-red communication or local RF communication, such as for instance used in walkie-talkies. Conventional hardware/software, such as a LAN interface and driver software, may be used for implementing locally operating communication means 110 and 112.

Using the communication facilities, digital data can be exchanged between the station and the host. Typically, the exchange is bi-directional. In certain simple systems, it may be sufficient if communication is only possible from the station to the host. The station 102 comprises a memory 114 for electronically storing a station identification, which uniquely identifies the station to the host. The identification may take several forms. For instance the identification may be an identification at communication level, such as a communication address or a telephone number. The identification may also take the form of an account number, which may also be used in combination with a communication identification. For each communication session, the identification of the station 102 is transferred to the host 100, allowing the host 100 to correlate data exchanged during the session to the identified station 102. It will be appreciated that in certain circumstances, such as where the station is fixedly connected to the host, the identification may be implicit (e.g. which port the station is connected to).

The station 102 comprises authentication means 116 for authenticating a message. As described above, the message may, for instance, be a digital representation of a user generated message, including an instruction to a bank, but may also be computer data or computer code, such as a Java applet, or messages generated by computers (e.g. for playing a multi-user game on several computers). The authentication is based upon an authentication algorithm. In principle the authentication algorithm may be chosen to suit the security requirements of the application. Many techniques for authenticating a message are known. One way is to use a symmetrical encryption algorithm like DES, where the station and the host share the same private key. The station encrypts the message using the key and transmits the encrypted message to the host, along with the station identification. The host uses the station identification to retrieve the key corresponding to the station and uses this key to decrypt the message. Various schemes, such as encryption feedback, message counters or time-stamps, may be used to ensure that the same message is authenticated differently, eliminating the possibility of intercepting and re-transmitting the same message, which then would be accepted again as a valid message by the host. In situations where it is preferred that the entire message (or part of it) is readable, the message may be transmitted in addition to the encrypted message. In such a case, the host can also verify the authenticity of the received data by encrypting the received plain message and comparing it to the received encrypted message. To reduce the length of the data to be transmitted, the authentication part may also be smaller. It is well-known that this can be achieved by using a one-way hash function and encrypting the resulting hash value. The algorithm itself may simple and based on basic techniques, like confusion and diffusion. Using a confusion technique, such as substitution, the relationship between the plain text and the cipher text is obscured. For high security applications, it may be preferred to use a substitution scheme which operates on blocks of more than one letter. Using a diffusion technique, such as a transposition (also called permutation), the redundancy of the plain text is spread out over the cipher text. It is preferred that linear operations are used in combination with at least one non-linear operation. Whatever authentication algorithm is used, for the system according to the invention it is assumed that the algorithm is used under control of a so-called authentication control element. For an authentication algorithm using DES, this could be the private key. For an authentication algorithm based on substitution this may be (part of) a substitution table. For an authentication algorithm based on a permutation this may be (part of) a permutation matrix. In general, using a different authentication control element will with a high likelihood cause the authentication algorithm to authenticate a same message differently. For most algorithms it will hold that if the same authentication control element is used, the same messages will be authenticated in the same way. However, some authentication algorithms may have measures, such as an internal feedback, ensuring that this is not the case. For such algorithms, the authentication control element can, for instance, play the role of an initial seed, where the algorithm is (at least partly) reset each time a new authentication control element is provided, or the authentication control element may act as a supplementary control vector, which is, for instance, combined with the internal state variable or to the output of the algorithm. The combination may, for instance, take the form of an XOR operation or an operation in GF(2^8) for byte-oriented algorithms. Depending on the algorithm, the authentication control element may be regarded as data or more as an operation.

According to the invention, the station 102 electronically stores a plurality of authentication items in a memory 118. It will be appreciated that the memories 114 and 118 may be combined. The station further comprises constructing means 120 for constructing the authentication control element. The construction means 120 derives the authentication control element from a small part of the entire set of authentication items. This may be done in various ways, like randomly selecting some items or some bits of some items and using the selected parts directly or after a mixing operation as the authentication control element. For each message a corresponding authentication control element is constructed. The authentication items are independent of the authentication algorithm, and as a consequence also the authentication control element is independent of the authentication algorithm. In this way any correlation which might occur when the authentication algorithm were to be used for authenticating a sequence of messages under control of the same authentication control element is broken by the unrelated authentication control element. It will be understood that the size of a small part with respect to the entire set of authentication items has to be determined in view of the requirements of the application in which the system is used and in view of further improvements as described below for further embodiments. In systems where the set of authentication items is highly static, a small part may correspond to a few percent or less of the entire set. In a system where the set is highly dynamic (i.e. regularly updated), a small part may be over 50% of the current set of authentication items, where the selected part is small compared to the superset of authentication items formed by the current authentication items and future changed authentication items. Such a higher percentage can particularly be used if the influence of an update of authentication items is spread over substantially all authentication items of an involved set of authentication items. Preferably, the authentication items have been generated randomly or selected randomly from a very large set of suitable authentication items. For instance, for a system used for financial transactions the authentication items may be generated in a secure manner using a high quality (real-)random sequence generator located at secure premises of a bank. The authentication items are loaded into the memory 118 of the station 102. The host 100 electronically stores a copy of the authentication items of the station in a memory 122. It will be appreciated that, depending on the authentication algorithm, the authentication item may be a data element, such as a bit or a byte, or an operation, such as a byte-wise XOR or a $GF(2^8)$ multiplication.

The authentication control element in practical circumstances causes the authentication algorithm with a high likelihood to authenticate the corresponding message uniquely. For high demanding systems, preferably each authentication control element is derived from at least one authentication item which has not been used before. Such a new authentication item may be combined with (e.g. mixed in with) authentication items which have been used before. In less demanding systems, a same selection of authentication items may be used a number of times for constructing an authentication control element. The construction means 120 should be such that even then the authentication control elements are different.

The host 100 comprises verification means 124 for verifying the authenticity of the received message. The verification means 124 checks the received message with an authentication algorithm which corresponds to the algorithm used by the station which send the message. The algorithm may be the same for all stations. If more than one algorithm is used, the host can locate the algorithm based on the received station identification. To this end, the station identification may be stored in a memory 128 of the host. It will be appreciated that the host may perform the verification by using the same authentication algorithm as used by the station to generate an authentication from the message and checks whether this matches the received authentication. For certain algorithms, the host may need to use an inverse algorithm of the algorithm used by the station. The host 100 comprises constructing means 126 for constructing for each received authenticated message an authentication control element from the authentication items for the identified station in a same manner as the station identified for the message.

In a further embodiment, the authentication algorithm authenticates each message in a manner unique for the station. This may be achieved by making the message authentication dependent on the station identification, which is unique for the station. Such a dependency may be obtained by deriving a key of the authentication algorithm or the authentication control element (partly) from the station identification.

Preferably, the authentication is made unique for the station by using authentication items which are unique for the station. The host 100 associates the copy of the authentication items of the station with the station, for instance, by combining the memories 122 and 128 and storing the station identification together with the authentication items. The construction means 126 of the host uses the received station identification to locate a matching station identification in memory 128 and via the matching identification locate the authentication items in memory 122 corresponding to the station.

In a further embodiment, the station 102 comprises modification means 130 for modifying at least one of the authentication items after the authenticating means has authenticated a message. The host comprises modification means 132 for modifying at least one of the authentication items for the station in the same way as the modification means 130 of the station. Preferably, the station 102 effectuates the modification after the station has received a confirmation from the host 100 that the host has successfully received the message and verified the authentication of the message. It is preferred that any confirmation message is also authenticated in a manner similar to a message transferred from the station to the host. The modification means 132 performs the modification if the verification means 124 has successfully verified an authenticated message received from the station. Also, additional transaction and roll-back mechanisms as used for distributed databases may be used to ensure that the station 102 and the host 100 remain synchronised. The modification may take place in any suitable form. One way would be to combine a selection of other authentication items to one new authentication item and to replace an existing authentication item with the new item. Preferably, the modification means 130, 132 is operative to modify an authentication item at least partly based on an event independent of the authentication items. Advantageously, the modification is based on the content of one or more of the preceding messages. As an alternative or in combination, the modification may also be based on a time-stamp of one or more of the preceding messages. If a time-stamp is used, the time-stamp is also transferred to the host 100. The host 100 and the station 102 may also share an algorithm for generating or collecting the same random data elements, where information exchanged between the station 102 and the host 100 determines which of the random data elements is used for generating the new authentication item.

In a further embodiment, the station 102 comprises alteration means 134 for altering the station identification after the authenticating means 116 has authenticated a message. The host 100 comprises alteration means 136 for altering the station identification for the station in the same way as the station after the verification means 124 has successfully verified a received authenticated message. Preferably, as described for generating the authentication control element, the altering is performed under control of a set of authentication items, which are independent of the altering algorithm. For instance, a selection of the authentication items may be 'mixed-in' with the station identification to obtain a new station identification. Preferably, a separate set of authentication items is used for generating the station identifications. Similarly as described for the modification means 130, the alteration means 134, 136 may alter the station identification at least partly based on a message and/or a time-stamp. This may, for instance, be achieved by modifying the set of authentication items used for generating the station identifications. The identification associated with the station may be a communication identification, such as a communication address or a telephone number, which is also used for the communication hardware/software to transfer messages between the desired devices. Preferably, the identification is a higher level identification, which is independent of the communication identification. An example of such a higher level identification is a bank account number. Both types of identification may also be used in combination. For such a combination, the communication identification may be kept the same whereas the higher level identification may be altered. If an identification, such as a virtual bank account number representing a real bank account number, is changed the underlying item (e.g. the real bank account number) is preferably kept the same, implying that in the host only the mapping of the representation (virtual number) to the actual underlying item is changed. Particularly, if the station interfaces to the user using the real underlying item, also the mapping in the station is updated. In some systems it may not be required that the station is aware of the real underlying item. It will be appreciated that a station (and as a consequence also the host) may have several different identifications, e.g. several bank account numbers, associated with the station, where each identification corresponds to its own unique set of authentication items. In order to exchange messages with several hosts, preferably the station has several identifications (at least one for each host) with corresponding set of authentication items.

In a further embodiment, the station 102 comprises data generation means 138 for generating additional data. The authentication means 116 is operative to authenticate a message in dependence on the additional data. The generated additional data is such that in practical circumstances with a high likelihood the additional data is different for each message. The additional data may be used in various ways. One way is to use the additional data in a manner 'invisible' to the outside world, except to the host 100. This can, for instance, be achieved, by first concatenating the original message and the additional data. Next, the authentication of the message with the additional data is determined, followed by removing the additional data before transferring the authenticated message (i.e. the original message plus the authentication for both the original message and the additional data) to the host 100. In this scenario, the host 100 also comprises data generation means 140 for generating additional data for a received authenticated message in a same manner as the identified station. The verification means 124 is operative to verify the authenticity of the received authenticated message in dependence on the additional data. The verification may be done similar to the authentication by first adding the additional data before checking the authentication. If the use of additional data is optional, it is preferred that the station 102 informs the host 100 whether the option is used for a message or not. This can be achieved by using an additional field, of for instance only one bit, in the message.

As an alternative to using the 'invisible' additional data, the authentication means 116 may also incorporate the additional data into the message before authenticating the message. In this scenario the additional data is not removed from the message by the station 102. The additional data may be simply concatenated to or may be mixed in with the original message. The verification means 124 verifies the authentication of the entire message (original message plus the additional data). For the purpose of verification, the entire message can be regarded as the message. After the verification, the additional data is removed and the original message is passed on for further processing. The removal may be straightforward, particularly if the additional data is simply concatenated. For a more complex mix operation, the host 100 may need to perform a same mixing operation as the station in order to be able to determine at which positions in the message the data elements of the additional data are located or an inverse mixing operation to be able to remove the additional data from the message.

It will be appreciated that also a combination of using 'visible' and 'invisible' additional data can be advantageously used. In such a combination, for instance, the station 102 and the host 100 share some information A. The station 102 generates an additional part B and uses both parts A and B to generate additional data. The authentication is based on the entire additional data. The station 102 transfers in combination with the message the additional data as well as the additional part B to the host 100. The host 100 generates in the same way the additional data using the received part B and the part A, which was already stored in the host 100. The host 100 checks whether the generated additional data matches the received additional data. If so, the authenticity of the data is checked further. Particularly if the additional data and the parts A and B are relatively small compared to the message, this provides an effective filter for the host 100 for fraudulent messages without requiring a full verification of the entire message.

In a further embodiment, the construction means 120 and 126 comprise at least one state variable which influences the construction of the authentication control element. The construction means 120 and 126 update the state variable at least each time a message has been authenticated. The use of a state variable allows the construction means to autonomously select different parts of the authentication items for a large sequence of messages. The construction means may, for instance, be based on a (pseudo-)random sequence generator, where the state variable corresponds to a feedback state of the generator. Preferably, any periodicity in the sequence is sufficiently large in view of the application. For instance, the periodicity is larger than the expected number of messages authenticated by the station 102. The state variable may also be a pointer to an authentication item (in the set of authentication items), which has been last used for generating the authentication control element. If more than one authentication items is used for constructing the authentication control element, a separate state variable may be used for all of them. The construction means 120 and 126 construct the authentication control element under control of a control vector. The control vector may directly influence the operation of the construction means 120, 126, or, alternatively, may influence the output of the construction means 120,126 in a different manner, for instance by XOR-ing the control vector over the basic output (e.g. random number) of the construction means 120,126. The control vector is derived from a selection of the authentication items, for instance by 'randomly' selecting an authentication item from a given set of authentication items and using the selected item as the control vector. Preferably a separate set of authentication items are used for forming the control vector. Like described earlier, these authentication items may also be modified.

Fig. 2 shows a flow-chart of a possible operation in the station 102. In step 200, the station collects information regarding the identification of the user of the station, such as a user name and password, or a fingerprint. In step 202 the identification is checked. If not accepted, the previous steps are repeated one or more times, if required with a time delay and a limit on the number of retries. (Preferably, the station 102 reports a failed attempt when the legitimate owner successfully gains access). If accepted, in step 204 information is collected from the user based on which a message is compiled. Next, in step 206 it is checked whether additional data is required. If so, in step 208 the additional data is generated and added (for instance appended) in step 210 to the message. In step 212 it is checked whether the message needs to be scrambled. If so, the scrambling occurs in step 213. The scrambling may be restricted to the original message generated at step 204 or may cover the entire message created at step 210. In step 214 the authentication for the message is generated and added to the message (e.g. appended) in step 216. In step 218 it is checked whether the option of dynamically changing the station identification is used. If so, in step

220 a new station identification is created. In both cases, in step 222 the station identification is added (e.g. prefixed) to the message. In step 224, one or more of the authentication items are changed. Preferably, authentication items which have been involved in any of the preceding steps are modified. Finally, in step 226 the message is sent to the host 100. Steps may be added to ensure that the host 100 and the station 102 stay synchronised (i.e. that authentication items and shared state variables are updated synchronously). In the exceptional case that the synchronisation in updating the virtual number used as the station identification is lost between the host and the station, it may be required to re-synchronise to a new virtual number by once using the real number as an identification. It will be appreciated that a similar corresponding flow-chart can be used to describe the activities of the host 100.

Fig. 3 shows a flow diagram of a basic operation which can be used in various steps of Fig. 2. The core operation is performed in block 300, where a (pseudo-)random number is generated. In block 302 a seed for the generator is selected from a first set of authentication items. A correlation which might occur in the sequence of generated numbers is broken by using a feedback and combining in block 304 the feedback with at least one authentication item. The combination may simply be an XOR operation. The authentication item is selected in block 306 from a second set of authentication items. It will be appreciated that the combination may also be in the output path 308 of the generator 304 instead of in the feedback path. The sets of authentication items may, for instance, consist of 100 authentication items each. The actual number is preferably chosen to optimally suit the need of the application. The selection performed in blocks 302 and 306 may be straightforward, like each time selecting a next one of the authentication items. Using such a scheme, preferably the first authentication items have been changed, by the time all authentication items have been used. The basic operation of Fig. 3 may be used directly to generate the additional data of step 208 or the new station identification of step 220 in Fig. 2. For the scrambling of step 214, the random numbers can be used as entries in a substitution matrix. For instance, assuming that the data elements of a message are bytes, a substitution table may be used with 256 entries each with a byte value, where each byte value specifies a substitution value for a data element with a value matching the entry number in the table. Alternatively, the substitution byte may be selected based on the position of the byte in the original message, if desired, in combination with the value of the byte in the original message. As an example, a pointer which (logically) points to an element in the substitution matrix is loaded with an initial offset. This offset may be selected using the basic operation of Fig. 3. The value of the first byte of the message is combined with the pointer value (e.g. added to it). The value of the element in the substitution matrix to which the pointer points at that moment is chosen as the substitution value. For the next byte of the message, similarly the value of the next byte of the message is combined with the then valid pointer value, etc.. The pointer may be one-dimensional, where the substitution matrix is logically arranged as a sequence where each row follows the previous row to form a long row (alternatively the columns may be logically concatenated). Such arrangement usually matches the physical arrangement for storing a matrix in a memory. Using a suitable modulo operation the pointer can be kept within the desired range of, in the example, 256 matrix elements. It will be appreciated that instead of a one dimensional pointer also a separate row and column index may be used. Instead of using the pointer or index value directly for selecting the substitution element, the value may also be fed through a randomiser, such as a random sequence generator, whose output is used as a pointer into the substitution table. In these examples it is assumed that the output of the generation 300 is a byte value. If not, a conversion may be required. The random numbers may also be used to create a permutation matrix for permuting the positions of data elements in the message. The basic operation can also be used for changing an authentication item in step 224. Since the changing, preferably, also depends on an external event, additional information, such as a message, and/or a time-stamp and/or a message counter, is fed into the random number generator 300. The output of the generator 300 may directly replace a constituent element (e.g. a value) of an authentication item.

For generating the authentication in step 214 of Fig. 2, a similar routine as described for the substitution may be used. In such a routine, in one round one data element (one signature element) is selected from a matrix (or long row) with data elements. Preferably, the initial data elements of the matrix have been generated randomly, where the data elements are refreshed by using the output of basic operation of Fig. 3 as a new data element (preferably in combination with a historical influence, such as the content of a previous message or a time-stamp, as described before). Alternatively, the output of the basic operation may be used to randomly shuffle the data elements of the matrix. A pointer which (logically) points to an element in the matrix is loaded with an initial offset. This offset may be selected using the basic operation of Fig. 3, which is preferably used under control of different sets of authentication items as used for generating the elements of the matrix. The value of the first byte of the message is combined with the pointer value (e.g. added to it). Next, the value of the next byte of the message is combined with the then valid pointer value, etc.. When all bytes of the message have been processed, the value of the element in the matrix to which the pointer points at that moment is chosen as the signature value. The security can be improved by repeating the routine to generate further signature values. Preferably, for each successive round of generating a signature value a different initial offset value is chosen. Alternatively, a subsequent rounds continues using the last obtained pointer value of the previous as the starting value for the new round.

It will be appreciated that, although the description focuses on the communication from the station 102 to the host 100, the same authentication items can also be used for communication from the host 100 to the station 102.

CLAIMS

1. An authentication system comprising at least one station and a host;

the station comprising authentication means for, based upon an authentication algorithm, authenticating a message; and communication means for sending the authenticated message to the host;

the host comprising communication means for receiving an authenticated message; and verification means for verifying the authenticity of the received message by checking the received message with an authentication algorithm corresponding to a station which sent the message;

characterised in that:

the station comprises a memory for electronically storing a plurality of authentication items;

the host comprises a memory for electronically storing the authentication items of the station in association with an identification of the station;

the station comprises constructing means for constructing for each message a corresponding authentication control element; the constructing means being operable to select for the message a part of the plurality of authentication items and to construct the authentication control element from the selected part, where the authentication control element in practical circumstances causes the authentication algorithm to substantially authenticate the corresponding message uniquely; and

the host comprises constructing means for constructing for each received authenticated message an authentication control element from the authentication items associated with a station which sent the message; the construction being the same as performed by the associated station.

2. A system as claimed in claim 1, characterised in that the station comprises a further memory for electronically storing an identification uniquely identifying the station with respect to the host; the authentication means is operative to authenticate the message in dependence on the identification; and the verification means is operative to verify the authenticity of the received message in dependence on an identification of the station which sent the message.

3. A system as claimed in claim 1, characterised in that the authentication items are unique for the station; the station comprises a further memory for electronically storing an identification uniquely identifying the station with respect to the host; the communication means of the station being operative to send the identification to the host in association with an authenticated message; and the host comprises means for locating the authentication items of a station in dependence on an identification received in association with an authenticated message.

4. A system as claimed in claim 1, characterised in that the station comprises modification means for modifying at least one of the authentication items after the authenticating means has authenticated a message and in that the host comprises modification means for modifying at least one of the authentication items for the station in the same way as the station after the verification means has successfully verified an authenticated message received from the station.

5. A system as claimed in claim 4, characterised in that the modification means is operative to modify an authentication item at least partly based on an event independent of the authentication items.

6. A system as claimed in claim 2 or 3, characterised in that the station comprises alteration means for altering the identification associated with the station after the authenticating means has authenticated a message and in that the host comprises alteration means for altering the identification associated with the station in the same way as the station after the verification means has successfully verified a received authenticated message.

7. A system as claimed in claim 1, characterised in that the authentication means comprises data generation means for generating additional data and in that the authentication means is operative to authenticate a message in dependence on the additional data; the additional data in practical circumstances with a high likelihood being different for each message.

8. A system as claimed in claim 7, characterised in that the verification means comprises data generation means for generating additional data for a received authenticated message in a same manner as the identified station and in that the verification means is operative to verify the authenticity of the received authenticated message in dependence on the additional data.

9. A system as claimed in claim 7, characterised in that the authentication means is operative to incorporate the additional data into the message before authenticating the message; and in that the verification means is operative to remove the additional data

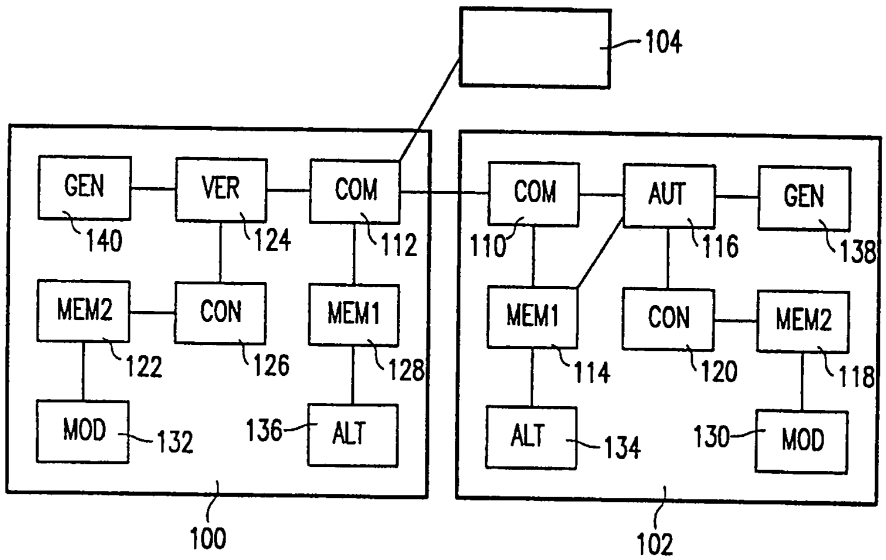
from a received authenticated message after having successfully verified the authentication of the message.

10. A system as claimed in claim 1, characterised in that:

the construction means comprises at least one state variable influencing the construction of the authentication control element;

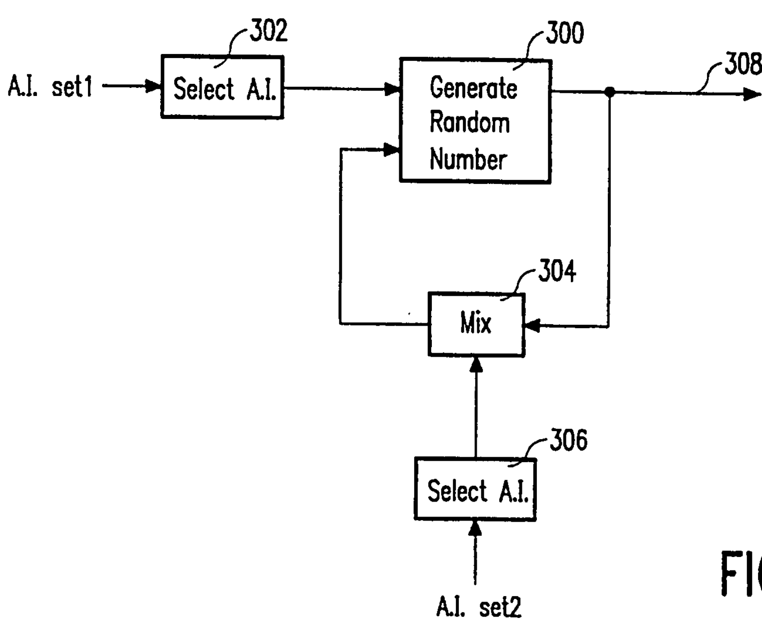
the construction means is operative to update the state variable at least each time a message has been authenticated; and to construct the authentication control element under control of a control vector derived from a selection of the authentication items.

1/2



104
GEN
VER
COM
COM
AUT
GEN
140
124
112
110
116
138
MEM2
CON
MEM1
MEM1
CON
MEM2
122
126
128
114
120
118
MOD
132
136
ALT
ALT
134
130
MOD
100
102

FIG. 1



302
300
308
A.I. set1
Select A.I.
Generate Random Number
304
Mix
306
Select A.I.
A.I. set2

FIG. 3

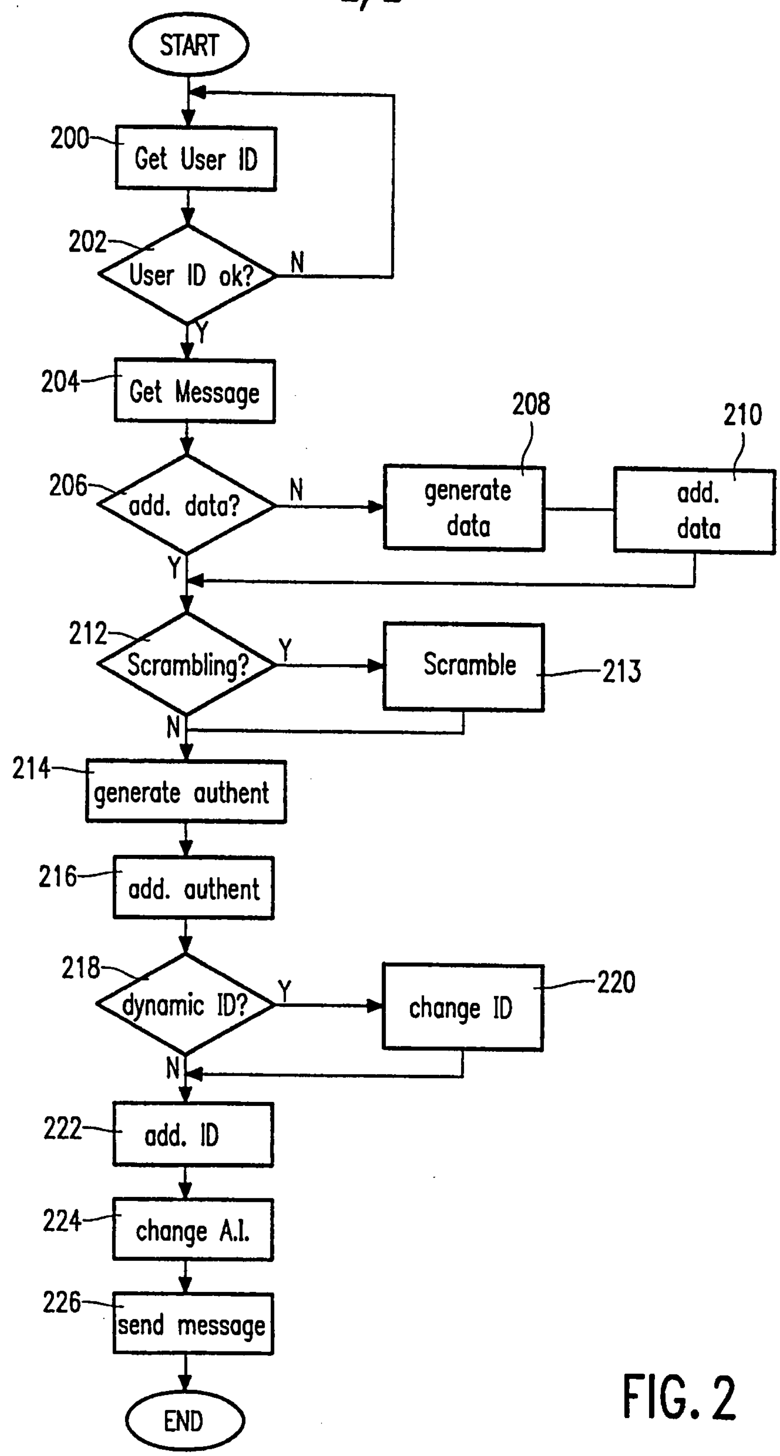

START
200
Get User ID
202
User ID ok?
N
Y
204
Get Message
208
210
206
add. data?
N
generate data
add. data
Y
212
Scrambling?
Y
Scramble
213
N
214
generate authent
216
add. authent
218
dynamic ID?
Y
change ID
220
N
222
add. ID
224
change A.I.
226
send message
END

FIG. 2

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IB 98/00578

A. CLASSIFICATION OF SUBJECT MATTER

IPC6: G06F 12/14, G06F 1/00
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC6: G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE,DK,FI,NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

WPI

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 0737907 A2 (SECURE COMPUTING CORPORATION), 16 October 1996 (16.10.96), column 6, line 44 - column 8, line 8 --	1
A	US 5615277 A (HOFFMAN), 25 March 1997 (25.03.97), column 6, line 52 - column 17, line 3 --	1-10
A	US 4677670 A (HENDERSON, JR.), 30 June 1987 (30.06.87), column 1, line 44 - column 12, line 15; column 3, line 25 - line 39 -- --------	1-10

☐ Further documents are listed in the continuation of Box C. ☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" erlier document but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance: the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance: the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
4 November 1998	06 -11- 1998
Name and mailing address of the ISA/ Swedish Patent Office Box 5055, S-102 42 STOCKHOLM Facsimile No. +46 8 666 02 86	Authorized officer Göran Magnusson Telephone No. +46 8 782 25 00

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IB 98/00578

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0737907 A2	16/10/96	AT 154150 T	15/06/97
		AU 667925 B	18/04/96
		AU 678937 B	12/06/97
		AU 4284793 A	18/11/93
		AU 5081196 A	18/07/96
		CA 2118246 A	28/10/93
		DE 69311331 D,T	30/10/97
		DK 636259 T	07/07/97
		EP 0636259 A,B	01/02/95
		SE 0636259 T3	
		JP 7505970 T	29/06/95
		US 5276735 A	04/01/94
		US 5499297 A	12/03/96
		US 5502766 A	26/03/96
		WO 9321581 A	28/10/93
US 5615277 A	25/03/97	US 5613012 A	18/03/97
		US 5764789 A	09/06/98
		US 5802199 A	01/09/98
		US 5805719 A	08/09/98
US 4677670 A	30/06/87	NONE	

Form PCT/ISA/210 (patent family annex) (July 1992)

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Edmund Colby Munger et al.
Serial Number:	10/259,494
Filing Date:	30 September 2002
Title:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST (As amended)
Examiner:	Krisna Lim.
Art Unit:	2153
Docket No.:	77580-012 (VRN- 1DV3)
Confirmation No.:	5257

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted (571-273-8300) to the US PTO, on the date indicated below.

Date: December 13 2007

Cynthia Joseph
Cynthia Joseph

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT "D"

This paper is in response to the non-final Office Action mailed 15 June 2007 for the above-identified application.

The Applicants appreciate the Examiner's thorough examination of the subject application, and request reconsideration and further examination in view of the following:

Amendments to the Claims, in the claim listing beginning on page 2 of this paper; and

Remarks, beginning on page 12 of this paper.

RECEIVED
CENTRAL FAX CENTER
DEC 13 2007

AMENDMENTS TO THE CLAIMS

Please cancel claims 1-10, 21-26, 33-38, and 40-68 without prejudice. . This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.-10. (Canceled)

11. (Original) A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:

(i) determining whether the intercepted DNS request corresponds to a secure server;

(ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer, and

(iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

12. (Original) The data processing device of claim 11, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

13. (Original) The data processing device of claim 12, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

14. (Original) The data processing device of claim 13, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

PAGE 6/12 * RCVD AT 12/13/2007 3:14:45 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-6/46 * DNIS:2738300 * CSID:1 617 535 3800 * DURATION (mm-ss):02-04

15. (Canceled)

16. (Original) A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) intercepting a DNS request sent by a client;

(ii) determining whether the intercepted DNS request corresponds to a secure server;

(iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

17. (Original) The computer readable medium of claim 16, wherein step (iii) comprises the steps of

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

18. (Original) The computer readable medium of claim 17, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

19. (Original) The computer readable medium of claim 18, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

20.-26. (Cancelled)

27. (Previously Presented) The data processing device of claim 11, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

28. (Previously Presented) The data processing device of claim 11, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

29.-30. (Canceled)

31. (Previously Presented) The computer readable medium of claim 16, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

32. (Previously Presented) The computer readable medium of claim 16, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

33.-38. (Canceled)

39. (Currently Amended) The computer readable medium of claim 20, wherein automatically initiating the encrypted channel between the client and the secure device comprises establishing an IP address hopping scheme between the client and the secure device.

40.-68 (Canceled)

REMARKS

Claims 11-14, 16-19, 27, 28, 31, 32 and 39 remain in the application. Claims 1-10, 21-26, 33-38 and 40-68 have been canceled.

The Examiner has indicated that claims 11-14, 16-19, 27, 28, 31, 32 and 39 are allowed. Applicants are canceling the rejected claims, 1-10, 21-26, 33-38 and 40-68, without prejudice, with the intent of filing a continuation application to pursue those claims. Accordingly, the application is now considered to be in condition for allowance. An early and favorable action thereon is therefore earnestly solicited.

Authorization is hereby given to charge our deposit account, No. 50-1133, for the fees corresponding to a Petition for Extension of Time (two-months) under 37 CFR § 1.136, and for any other fees that may be required for the prosecution of the subject application.

If a telephone conference will expedite prosecution of the application, the Examiner is invited to telephone the undersigned.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Date: 13 December 2007

Toby H. Kusmer, P.C.
Reg. No. 26,418
Attorney for Applicants
28 State Street
Boston, MA 02109-1775
DD Telephone: (617) 535-4065
Facsimile: (617)535-3800
e-mail: tkusmer@mwe.com

BST99 1557319-1.077580.0012

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of:	Edmund Colby Munger, et al.
Serial No.:	10/259,494
Filing Date:	September 30, 2002
Title:	Improvement To An Agile Network Protocol For Secure Communications With Assured System Availability
Docket No.:	77580-012 (VRNK-1CPDV3)

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

CHANGE OF CORRESPONDENCE ADDRESS AND REVOCATION OF POWER OF ATTORNEY AND APPOINTMENT OF NEW ATTORNEY

Dear Sir:

Applicants in the above-identified patent application hereby revoke all powers of attorney previously given in connection with the above-identified patent application and hereby appoint the following attorneys, with full power of substitution, to transact all business in the Patent and Trademark Office connected therewith:

All attorneys associated with CUSTOMER NUMBER 23630

It is requested that all correspondence regarding this patent application be directed to:

Toby H. Kusmer, P.C.
McDermott Will & Emery LLP
28 State Street
Boston, Massachusetts 02109-1775
Telephone: (617) 535-4065
Facsimile: (617) 535-3800
E-mail: tkusmer@mwe.com

Respectfully submitted,

Date: 12/11/07

By [signature]
Name: Kendall Larsen
Title: President

PTO/SB/96 (04-07)
Approved for use through 09/30/2007. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: Munger, et al.

Application No./Patent No.: 10/259,494 Filed/Issue Date: September 30, 2002

Entitled: IMPROVEMENT TO AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

VirnetX, Inc., a Corporation
(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that it is:

1. ☑ the assignee of the entire right, title, and interest; or

2. ☐ an assignee of less than the entire right, title and interest
(The extent (by percentage) of its ownership interest is ________ %)

in the patent application/patent identified above by virtue of either:

A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel ________, Frame ________, or for which a copy thereof is attached.

OR

B. ☑ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Munger et al. (Inventors) To: Science Applications International Corporation
The document was recorded in the United States Patent and Trademark Office at
Reel 015855, Frame 0419, or for which a copy thereof is attached.

2. From: Science Applications International Corporation To: VirnetX, Inc.
The document was recorded in the United States Patent and Trademark Office at
Reel 018757, Frame 0326, or for which a copy thereof is attached.

3. From: ________ To: ________
The document was recorded in the United States Patent and Trademark Office at
Reel ________, Frame ________, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet.

☐ As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.
[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

[signature] 12/11/07
Signature Date

Kendall Larsen 831.438.8200
Printed or Typed Name Telephone number

President
Title

McDermott
Will&Emery

RECEIVED
CENTRAL FAX CENTER
DEC 13 2007

Boston Brussels Chicago Düsseldorf London Los Angeles Miami Munich
New York Orange County Rome San Diego Silicon Valley Washington, D.C.
Strategic alliance with MWE China Law Offices (Shanghai)

FACSIMILE

Date: December 13, 2007 **Time Sent:**

To:	**Company:**	**Facsimile No:**	**Telephone No:**
Commissioner for Patents – Group Art Unit: 2153	U.S. Patent and Trademark Office	1.571.273.8300	

From:	Toby H. Kusmer, P.C.	*Direct Phone:*	617.535.4065
E-Mail:	tkusmer@mwe.com	*Direct Fax:*	617.535.3800
Sent By:	Cynthia Joseph	*Direct Phone:*	617.535.4111
Client/Matter/Tkpr:	77580-012/5496	*Original to Follow by Mail:*	No
		Number of Pages, Including Cover:	12

Re: In re Application of: Edmund Colby Munger, et al.
Serial No.: 10/259,494
Filing Date: September 30, 2002
Title: Establishment Of A Secure Communication Link Based On A Domain Name Service (DNS) Request (As amended)
Docket No.: 77580-012 (VRNK-1CPDV3)

Message:

Please enter the attached Amendment "D" and attachments.

Main Facsimile: 617.535.3800 Facsimile Operator: 617.535.4000

U.S. practice conducted through McDermott Will & Emery LLP.
28 State Street Boston, Massachusetts 02109-1775 Telephone: 617.535.4000

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of:	Edmund Colby Munger, et al.
Serial No.:	10/259,494
Filing Date:	September 30, 2002
Title:	Establishment Of A Secure Communication Link Based On A Domain Name Service (DNS) Request (As amended)
Group Art Unit:	2153
Confirmation No.:	5257
Docket No.:	77580-012 (VRNK-1CPDV3)

MAIL STOP Amendment
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER

Applicants transmit herewith the following documents in the above-identified application:

1. Amendment "D";
2. Combined Amendment & Petition For Extension of Time Under 37CFR 1.136(a);
3. Change Of Correspondence Address And Revocation Of Power Of Attorney And Appointment Of New Power; and
4. Statement Under 37 CFR 3.73(b).

The Commissioner is authorized to charge any filing fee or credit any overpayment to Deposit Account No. 50-1133.

Respectfully submitted,

Toby H. Kusmer, P.C.
Reg. No. 26,418
McDermott Will & Emery LLP
28 State Street
Boston, MA 02109-1775
Telephone: 617.535.4065
Facsimile: 617.535.3800
e-mail: tkusmer@mwe.com

CERTIFICATE OF FACSIMILE TRANSMISSION

I hereby certify that this correspondence is being facsimile transmitted, via Facsimile No. 571.273.8300, to the U.S. Patent and Trademark Office on the date indicated below and is addressed to Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450.

December 13 2007
Date

Cynthia Joseph

COMBINED AMENDMENT & PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) (Large Entity)	Docket No. 77580-012 (VRNK-1CPDV3)

In Re Application Of: **Edmund Colby Munger, et al.**

Application No.	Filing Date	Examiner	Customer No.	Group Art Unit	Confirmation No.
10/259,494	September 30, 2002	Krisna Lim	23630	2153	5257

Invention: **ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST (As amended)**

COMMISSIONER FOR PATENTS:

This is a combined amendment and petition under the provisions of 37 CFR 1.136(a) to extend the period for filing a response to the Office Action of June 15, 2007 (*Date*) in the above-identified application.

The requested extension is as follows (check time period desired):

☐ One month ☐ Two months ☒ Three months ☐ Four months ☐ Five months

from: September 15, 2007 (*Date*) until: December 13, 2007 (*Date*)

The fee for the amendment and extension of time has been calculated as shown below:

CLAIMS AS AMENDED

	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST # PREV. PAID FOR		NUMBER EXTRA CLAIMS PRESENT		RATE	ADDITIONAL FEE
TOTAL CLAIMS	13	-	68	=	0	x	$50.00	$0.00
INDEP. CLAIMS	2	-	11	=	0	x	$210.00	$0.00
							FEE FOR AMENDMENT	$0.00
							FEE FOR EXTENSION OF TIME	$1,050.00
							TOTAL FEE FOR AMENDMENT AND EXTENSION OF TIME	$1,050.00

12/14/2007 PCHOMP 00000019 501133 10259494
01 FC:1253 1050.00 DA

P28LARGE/REV06

RECEIVED
CENTRAL FAX CENTER
DEC 13 2007

COMBINED AMENDMENT & PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) (Large Entity)	Docket No. 77580-012 (VRNK-1CPDV3)

The fee for the amendment and extension of time is to be paid as follows:

☐ A check in the amount of   for the amendment and extension of time is enclosed.

☒ Please charge Deposit Account No. **501133** in the amount of **$1,050.00**

☒ The Director is hereby authorized to charge payment of the following fees associated with this communication or credit any overpayment to Deposit Account No. **501133**

- ☒ Any additional filing fees required under 37 C.F.R. 1.16.
- ☒ Any patent application processing fees under 37 CFR 1.17.

☒ If an additional extension of time is required, please consider this a petition therefor and charge any additional fees which may be required to Deposit Account No. **501133**.

☐ Payment by credit card. Form PTO-2038 is attached.
WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

Signature

Dated: 12/12/07

Toby H. Kusmer, P.C.
Reg. No. 26,418
McDermott Will & Emery LLP
28 State Street
Boston, MA 02109
Telephone: 617.535.4065
Facsimile: 617.535.3800
e-mail: tkusmer@mwe.com

cc:

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to the "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on 12/13/07 (Date) or by facsimile

Signature of Person Mailing Correspondence

Cynthia Joseph

Typed or Printed Name of Person Mailing Correspondence

P28LARGE/REV06

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number	Filing Date	
	10/259,494	09/30/2002	☐ To be Mailed

APPLICATION AS FILED – PART I

FOR	NUMBER FILED (Column 1)	NUMBER EXTRA (Column 2)	SMALL ENTITY ☐ RATE ($)	SMALL ENTITY FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
			TOTAL			TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

AMENDMENT	12/13/2007	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 13	Minus	** 65	= 0	X $ =		OR	X $50=	0
	Independent (37 CFR 1.16(h))	* 2	Minus	***10	= 0	X $ =		OR	X $210=	0
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	**0**

AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	SMALL ENTITY RATE ($)	SMALL ENTITY ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
SHARON C. COTTON

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

22907 7590 01/11/2008

BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

EXAMINER	
LIM, KRISNA	
ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 01/11/2008

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

TITLE OF INVENTION: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1440	$300	$0	$1740	04/11/2008

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: **Mail** **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

22907 7590 01/11/2008

BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

TITLE OF INVENTION: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS WITH ASSURED SYSTEM AVAILABILITY

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1440	$300	$0	$1740	04/11/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-225000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 ____________
2 ____________
3 ____________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:

❑ Issue Fee
❑ Publication Fee (No small entity discount permitted)
❑ Advance Order - # of Copies ____________

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**

❑ A check is enclosed.
❑ Payment by credit card. Form PTO-2038 is attached.
❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number ____________ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

❑ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
❑ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ____________________ Date ____________________

Typed or printed name ____________________ Registration No. ____________________

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

22907 7590 01/11/2008

BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

EXAMINER	
LIM, KRISNA	
ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 01/11/2008

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 308 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 308 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

Notice of Allowability	Application No.	Applicant(s)	
	10/259,494	MUNGER ET AL.	
	Examiner	Art Unit	
	Krisna Lim	2153	

– The MAILING DATE of this communication appears on the cover sheet with the correspondence address–

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *the amendment filed 12/13/07*.

2. ☒ The allowed claim(s) is/are *11-14,16-19,27,28,31 and 32*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____.
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☒ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____.
7. ☒ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____

Examiner's Amendment

An Examiner's Amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 C.F.R. 1.312. To ensure consideration of such an amendment, it **MUST** be submitted no later than the payment of the Issue Fee.

In the claims:

Cancel claim 39 (note: this claim depends on a non-exist claim or a canceled claim).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Krisna Lim whose telephone number is 571-272-3956 The examiner can normally be reached on Monday to Friday from 9:30 AM to 6:00 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Glenton Burgess, can be reached on 571-272-3949. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Kl

January 6, 2008

KRISNA LIM
PRIMARY EXAMINER

10/259,494

INFORMATION DISCLOSURE CITATION IN AN APPLICATION (PTO-1449)	ATTY. DOCKET NO. 077580-0012	SERIAL NO. 10/259,494
	APPLICANT Munger et al.	
	FILING DATE Sep. 20, 2002	GROUP ~~1633~~ 2153

U.S. PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Document Number Number-Kind Code2 (if known)		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
/K/	A1	US	4,933,846 A	6/12/1990	Humphrey et al.	
	A2	US	4,988,990 A	1/29/1991	Warrior	
	A3	US	5,276,735 A	1/4/1994	Boebert et al	
	A4	US	5,311,593 A	5/10/1994	Carmi	
	A5	US	5,329,521 A	7/12/1994	Walsh et al.	
	A6	US	5,341,426 A	8/23/1994	Barney et al.	
	A7	US	5,367,643 A	11/22/1994	Chang et al	
	A8	US	5,559,883 A	9/24/1996	Williams	
	A9	US	5,561,669 A	10/1/1996	Lenney et al	
	A10	US	5,588,060 A	12/24/1996	Aziz	
	A11	US	5,625,626 A	4/29/1997	Umekita	
	A12	US	5,654,695 A	8/5/1997	Olnowich et al	
	A13	US	5,682,480 A	10/28/1997	Nakagawa	
	A14	US	5,689,566 A	11/18/1997	Nguyen	
	A15	US	5,740,375 A	4/14/1998	Dunne et al.	
	A16	US	5,774,660 A	6/30/1998	Brendel et al	
	A17	US	5,787,172 A	7/28/1998	Arnold	
	A18	US	5,796,942 A	8/18/1998	Esbensen	
	A19	US	5,805,801 A	9/8/1998	Holloway et al.	
	A20	US	5,842,040 A	11/24/1998	Hughes et al.	
	A21	US	5,845,091 A	12/1/1998	Dunne et al.	
	A22	US	5,867,650 A	2/2/1998	Osterman	
	A23	US	5,870,610 A	2/9/1999	Beyda et al.	
	A24	US	5,878,231 A	5/2/1999	Baehr et al	
	A25	US	5,892,903 A	4/6/1999	Klaus	
	A26	US	5,898,830 A	4/27/1999	Wesinger, Jr. et al.	
	A27	US	5,905,859 A	5/18/1999	Holloway et al.	
	A28	US	5,918,019 A	6/29/1999	Valencia	
	A29	US	5,996,016 A	11/30/1999	Thalheimer et al.	
	A30	US	6,006,259 A	12/21/1999	Adelman et al.	
	A31	US	6,006,272 A	12/21/1999	Aravamudan et al	

EXAMINER	DATE CONSIDERED
KRISNA LIM	1/6/08

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

10/259,494

SHEET 2 OF 4

INFORMATION DISCLOSURE CITATION IN AN APPLICATION (PTO-1449)	ATTY. DOCKET NO. 077580-0012	SERIAL NO. 10/259,494
	APPLICANT **Munger et al.**	
	FILING DATE **Sep. 20, 2002**	GROUP ~~1633~~ 2153

U.S. PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Document Number Number-Kind Code2 (if known)		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
KL	A32	US	6,016,318 A	1/18/2000	Tomoike	
	A33	US	6,016,512	1/18/2000	Huitema	
	A34	US	6,041,342 A	3/21/2000	Yamaguchi	
	A35	US	6,052,788 A	4/18/2000	Wesinger, Jr. et al.	
	A36	US	6,055,574 A	4/25/2000	Smorodinsky et al.	
	A37	US	6,061,736 A	5/9/2000	Rochberger et al	
	A38	US	6,079,020 A	6/20/2000	Liu	
	A39	US	6,092,200 A	7/18/2000	Muniyappa et al.	
	A40	US	6,119,171 A	9/12/2000	Alkhatib	
	A41	US	6,119,234 A	9/12/2000	Aziz et al.	
	A42	US	6,147,976 A	11/14/2000	Shand et al.	
	A43	US	6,157,957 A	12/5/2000	Berthaud	
	A44	US	6,158,011 A	12/5/2000	Chen et al.	
	A45	US	6,168,409 B1	1/2/2001	Fare	
	A46	US	6,175,867 B1	1/16/2001	Taghadoss	
	A47	US	6,178,409 B1	1/23/2001	Weber et al.	
	A48	US	6,178,505 B1	1/23/2001	Schneider et al	
	A49	US	6,179,102 B1	1/30/2001	Weber, et al.	
	A50	US	6,222,842 B1	4/24/2001	Sasyan et al.	
	A51	US	6,226,751 B1	5/1/2001	Arrow et al	
	A52	US	6,233,618 B1	5/15/2001	Shannon	
	A53	US	6,243,360 B1	6/5/2001	Basilico	
	A54	US	6,243,749 B1	6/5/2001	Sitaraman et al.	
	A55	US	6,243,754 B1	6/5/2001	Guerin et al	
	A56	US	6,256,671 B1	7/3/2001	Strentzsch et al.	
	A57	US	6,263,445 B1	7/17/2001	Blumenau	
	A58	US	6,286,047 B1	9/4/2001	Ramanathan et al	
	A59	US	6,301,223 B1	10/9/2001	Hrastar et al	
	A60	US	6,308,274 B1	10/23/2001	Swift	
	A61	US	6,311,207 B1	10/30/2001	Mighdoll et al	
	A62	US	6,324,161 B1	11/27/2001	Kirch	
	A63	US	6,330,562 B1	12/11/2001	Boden et al.	

EXAMINER	DATE CONSIDERED
Krisna Lim	1/6/08

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

10/259,494

SHEET 3 OF 4

INFORMATION DISCLOSURE CITATION IN AN APPLICATION (PTO-1449)	ATTY. DOCKET NO. 077580-0012	SERIAL NO. 10/259,494
	APPLICANT Munger et al.	
	FILING DATE Sep. 20, 2002	GROUP ~~1633~~ 2153

U.S. PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Document Number Number-Kind Code2 (if known)		Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines, Where Relevant Passages or Relevant Figures Appear
KL	A64	US	6,332,158 B1	12/18/2001	Risley et al.	
	A65	US	6,353,614 B1	3/5/2002	Borella et al.	
	A66	US	6,430,155 B1	8/6/2002	Davie et al	
	A67	US	6,430,610 B1	8/6/2002	Carter	
	A68	US	6,487,598 Bi	11/26/2002	Valencia	
	A69	US	6,502,135 B1	12/31/2002	Munger et al	
	A70	US	6,505,232 B1	1/7/2003	Mighdoll et al	
	A71	US	6,510,154 B1	1/21/2003	Mayes et al	
	A72	US	6,549,516 B1	4/15/2003	Albert et al	
	A73	US	6,557,037 B1	4/29/2007	Provino	
	A74	US	6,571,296 B1	5/27/2002	Dillon	
	A75	US	6,571,338 B1	5/27/2003	Shaio et al.	
	A76	US	6,581,166 B1	7/17/2003	Hirst et al.	
	A77	US	6,618,761 B2	9/9/2003	Munger et al.	
	A78	US	6,671,702 B2	12/30/2003	Kruglikov et al	
	A79	US	6,687,551 B1	2/3/2004	Steindl	
	A80	US	6,714,970 B1	3/30/2004	Fiveash et al.	
	A81	US	6,717,949 B1	4/6/2004	Boden et al.	
	A82	US	6,760,766 B1	7/6/2004	Sahlqvist	
	A83	US	6,826,616 B2	11/30/2004	Larson et al.	
	A84	US	6,839,759 B2	1/4/2005	Larson et al.	
	A85	US	7,010,604 B1	3/7/2006	Munger et al.	
	A86	US	7,133,930 B2	11/7/2006	Munger et al.	
	A87	US	7,188,180 B2	3/6/2007	Larson et al.	
	A88	US	7,197,563 B2	3/27/2007	Sheymov et al.	
	A89	US	2002/0004898 A1	1/10/2002	Droge	
	A90	US	2005/0055306 A1	3/10/2005	Miller et al.	

FOREIGN PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Foreign Patent Document Country Code3-Number4-Kind Code5 (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines Where Relevant Figures Appear	Translation Yes	Translation No
KL	B1	GB 2334181 A	8/11/1999	Noblet			
KL	B2	EP 836306A1	4/15/1998	Sasyan et al.			

EXAMINER	DATE CONSIDERED
Krisna Lim	1/6/08

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

10/259,494

SHEET 4 OF 4

INFORMATION DISCLOSURE CITATION IN AN APPLICATION (PTO-1449)	ATTY. DOCKET NO. 077580-0012	SERIAL NO. 10/259,494
	APPLICANT Munger et al.	
	FILING DATE Sep. 20, 2002	GROUP ~~1633~~ 2153

FOREIGN PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Foreign Patent Document Country Code-Number-Kind Codes (if known)	Publication Date MM-DD-YYYY	Name of Patentee or Applicant of Cited Document	Pages, Columns, Lines Where Relevant Figures Appear	Translation Yes	No
KL	B3	WO 9827783 A	6/25/1998	Tello et al.			
	B4	WO 00/17775	3/30/2000	Miller et al.			
	B5	WO 00/70458	11/23/2000	Sheymov			
	B6	WO 98 55930	12/10/1998	Tang			

OTHER ART (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER'S INITIALS	CITE NO.	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc.), date, page(s), volume-issue number(s), publisher, city and/or country where published.	
KL	C1	Alan 0. Frier et al., "The SSL Protocol Version 3.0", Nov. 18, 1996, printed from http://www.netscape.com/eng/ss13/ draft302.txt on Feb. 4, 2002, 56 pages.	
	C2	Davila J et al, "Implementation of Virtual Private Networks at the Transport Layer", Information Security, Second International Work-shop, ISW'99. Proceedings (Lecture Springer-Verlag Berlin, Germany, [Online] 1999, pp. 85-102, XP002399276, ISBN 3-540-66695-B, retrieved from the Internet: URL: http://www. springerlink.com/content/4uac0tb0hecoma89/fulltext.pdf>	
	C3	Dolev, Shlomi and Ostrovsky, Rafil, "Efficient Anonymous Multicast and Reception" (Extended Abstract), 16 pages.	
	C4	F. Halsall, "Data Communications, Computer Networks and Open Systems", Chapter 4, Protocol Basics, 1996, pp. 198–203.	
	C5	Glossary for the Linux FreeS/WAN project, printed from http://liberty.freeswan.org/ freeswan_trees/freeswan-1.3/ doc/glossary.html on Feb. 21, 2002, 25 pages.	
	C6	J. Gilmore, "Swan: Securing the Internet against Wiretapping", printed from http:// liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/rationale.html on Feb. 21, 2002, 4 pages.	
	C7	Linux FreeS/WAN Index File, printed from http://liberty.freewan.org/freeswan trees/freeswan-1.3/doc/ on Feb. 21, 2002, 3 Pages.	
	C8	Reiter, Michael K. and Rubin, Aviel D. (AT&T Labs—Research), "Crowds: Anonymity for Web Transactions", pp. 1-23.	
	C9	RFC 2401-Security Architecture for the Internet Protocol (RTP)	
	C10	RFC 2543-SIP: Session Initiation Protocol (SIP or SIPS)	
	C11	Rubin, Aviel D., Geer, Daniel, and Ranum, Marcus J. (Wiley Computer Publishing), "Web Security Sourcebook", pp. 82-94.	
	C12	Search Report, IPER (dataed Nov. 13, 2002), International Application No. PCT/US01/04340.	
	C13	Search Report, IPER (dated Feb. 06, 2002), International Application No. PCT/US01/13261.	
	C14	Search Report, IPER (dated Jan. 14, 2003), International Application No. PCT/US01/13260.	
	C15	Shankar, A.U. "A verified sliding window protocol with variable flow control". Proceedings of ACM SIGCOMM conference on Communications architectures & protocols. pp. 84-91, ACM Press, NY,NY 1986.	

EXAMINER	DATE CONSIDERED
Krisna Lim	1/6/08

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.
1 Applicant's unique citation designation number (optional). 2 Applicant is to place a check mark here if English language Translation is attached.

Issue Classification	Application/Control No. 10/259,494	Applicant(s)/Patent under Reexamination MUNGER ET AL.
	Examiner Krisna Lim	Art Unit 2153

ISSUE CLASSIFICATION

ORIGINAL		INTERNATIONAL CLASSIFICATION					
CLASS	SUBCLASS	CLAIMED					NON-CLAIMED
709	225	G	06	F	15	/173	/
						/	/
						/	/
						/	/
						/	/
						/	/
						/	/

CROSS REFERENCES				
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)			
709	229			

(Assistant Examiner) (Date)	KRISNA LIM PRIMARY EXAMINER 1/6/08 (Primary Examiner) (Date)	Total Claims Allowed: 12	
(Legal Instruments Examiner) (Date)		O.G. Print Claim(s) 1	O.G. Print Fig. 26

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
	1	11	31		61		91		121		151		181
	2	12	32		62		92		122		152		182
	3		33		63		93		123		153		183
	4		34		64		94		124		154		184
	5		35		65		95		125		155		185
	6		36		66		96		126		156		186
	7		37		67		97		127		157		187
	8		38		68		98		128		158		188
	9		39		69		99		129		159		189
	10		40		70		100		130		160		190
1	11		41		71		101		131		161		191
2	12		42		72		102		132		162		192
3	13		43		73		103		133		163		193
4	14		44		74		104		134		164		194
	15		45		75		105		135		165		195
7	16		46		76		106		136		166		196
8	17		47		77		107		137		167		197
9	18		48		78		108		138		168		198
10	19		49		79		109		139		169		199
	20		50		80		110		140		170		200
	21		51		81		111		141		171		201
	22		52		82		112		142		172		202
	23		53		83		113		143		173		203
	24		54		84		114		144		174		204
	25		55		85		115		145		175		205
	26		56		86		116		146		176		206
5	27		57		87		117		147		177		207
6	28		58		88		118		148		178		208
	29		59		89		119		149		179		209
	30		60		90		120		150		180		210

U.S. Patent and Trademark Office Part of Paper No. 20080106

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 5257

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/259,494	09/30/2002 RULE	709	2143	000479.00082

APPLICANTS

Edmund Colby Munger, Crownsville, MD;
Robert Dunham Short III, Leesburg, VA;
Victor Larson, Fairfax, VA;
Michael Williamson, South Riding, VA;

** **CONTINUING DATA** *************************

This application is a DIV of 09/504,783 02/15/2000 PAT 6,502,135 now U.S. Patent No 6,502,135
which is a CIP of 09/429,643 10/29/1999
which claims benefit of 60/106,261 10/30/1998
and claims benefit of 60/137,704 06/07/1999

** **FOREIGN APPLICATIONS** ********************

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 10/31/2002

Foreign Priority claimed / 35 USC 119 (a-d) conditions met / Verified and Acknowledged	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ☐ yes ☑ no 35 USC 119 (a-d) conditions met ☐ yes ☑ no ☐ Met after Allowance Verified and Acknowledged ______ Examiner's Signature ______ Initials	MD	35	20	6

ADDRESS

22907

TITLE

Agile network protocol for secure communications with assured system availability

FILING FEE RECEIVED		
992	FEES: Authority has been given in Paper No. _________ to charge/credit DEPOSIT ACCOUNT No. _________ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other ________________ ☐ Credit

Search Notes	Application/Control No. 10/259,494	Applicant(s)/Patent under Reexamination MUNGER ET AL.	
	Examiner Krisna Lim	Art Unit 2153	

SEARCHED			
Class	Subclass	Date	Examiner
709	217- 225 229	1/6/08	KL
713	201		

INTERFERENCE SEARCHED			
Class	Subclass	Date	Examiner
709	225	1/6/08	KL
709	229	1/6/08	KL

SEARCH NOTES (INCLUDING SEARCH STRATEGY)	DATE	EXMR
EAST TEXT SEARCH	10/16/2006	KL
Inventor Name Search	10/16/2006	KL

U.S. Patent and Trademark Office

Part of Paper No. 20061016

Index of Claims	Application/Control No.	Applicant(s)/Patent under Reexamination
	10/259,494	MUNGER ET AL.
	Examiner	**Art Unit**
	Krisna Lim	2153

√	**Rejected**	–	**(Through numeral) Cancelled**	N	**Non-Elected**	A	**Appeal**
=	**Allowed**	+	**Restricted**	I	**Interference**	O	**Objected**

Claim		Date								
Final	Original	1/6/08								
	~~1~~									
	~~2~~									
	~~3~~									
	~~4~~									
	~~5~~									
	~~6~~									
	~~7~~									
	~~8~~									
	~~9~~									
	~~10~~									
1	11	=								
2	12	=								
3	13	=								
4	14	=								
	~~15~~									
7	16	=								
8	17	=								
9	18	=								
10	19	=								
	~~20~~									
	~~21~~									
	~~22~~									
	~~23~~									
	~~24~~									
	~~25~~									
	~~26~~									
5	27	=								
6	28	=								
	~~29~~									
	~~30~~									
11	31	=								
12	32	=								
	33									
	34									
	35									
	36									
	37									
	38									
	39									
	40									
	41									
	42									
	43									
	44									
	45									
	46									
	47									
	48									
	49									
	50									

Claim		Date								
Final	Original									
	51									
	52									
	53									
	54									
	55									
	56									
	57									
	58									
	59									
	60									
	61									
	62									
	63									
	64									
	65									
	66									
	67									
	68									
	69									
	70									
	71									
	72									
	73									
	74									
	75									
	76									
	77									
	78									
	79									
	80									
	81									
	82									
	83									
	84									
	85									
	86									
	87									
	88									
	89									
	90									
	91									
	92									
	93									
	94									
	95									
	96									
	97									
	98									
	99									
	100									

Claim		Date								
Final	Original									
	101									
	102									
	103									
	104									
	105									
	106									
	107									
	108									
	109									
	110									
	111									
	112									
	113									
	114									
	115									
	116									
	117									
	118									
	119									
	120									
	121									
	122									
	123									
	124									
	125									
	126									
	127									
	128									
	129									
	130									
	131									
	132									
	133									
	134									
	135									
	136									
	137									
	138									
	139									
	140									
	141									
	142									
	143									
	144									
	145									
	146									
	147									
	148									
	149									
	150									

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 5257

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/259,494	09/30/2002 RULE	709	2153	000479.00082

APPLICANTS

Edmund Colby Munger, Crownsville, MD;
Robert Dunham Short III, Leesburg, VA;
Victor Larson, Fairfax, VA;
Michael Williamson, South Riding, VA;

** **CONTINUING DATA** *************************

This application is a DIV of 09/504,783 02/15/2000 PAT 6,502,135
which is a CIP of 09/429,643 10/29/1999 PAT 7,010,604
which claims benefit of 60/106,261 10/30/1998
and claims benefit of 60/137,704 06/07/1999

** **FOREIGN APPLICATIONS** ********************

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 10/31/2002

Foreign Priority claimed ☐ yes ☐ no 35 USC 119 (a-d) conditions met ☐ yes ☐ no ☐ Met after Allowance Verified and Acknowledged ______ Examiner's Signature ______ Initials	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
	MD	35	20	6

ADDRESS
22907

TITLE
ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

DG 2.8

FILING FEE RECEIVED	FEES	
3394	FEES: Authority has been given in Paper No. ________ to charge/credit DEPOSIT ACCOUNT No. ________ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other ________ ☐ Credit

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 5257

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/259,494	09/30/2002 RULE	709	2153	000479.00082

APPLICANTS

Edmund Colby Munger, Crownsville, MD;
Robert Dunham Short III, Leesburg, VA;
Victor Larson, Fairfax, VA;
Michael Williamson, South Riding, VA;

** **CONTINUING DATA** *************************

This application is a DIV of 09/504,783 02/15/2000 PAT 6,502,135
which is a CIP of 09/429,643 10/29/1999 PAT 7,010,604
which claims benefit of 60/106,261 10/30/1998
and claims benefit of 60/137,704 06/07/1999

** **FOREIGN APPLICATIONS** ********************

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 10/31/2002

Foreign Priority claimed ☐ yes ☐ no 35 USC 119 (a-d) conditions met ☐ yes ☐ no ☐ Met after Allowance Verified and Acknowledged ______ Examiner's Signature Initials	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
	MD	35	20	6

ADDRESS
22907

TITLE
ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

FILING FEE RECEIVED	FEES: Authority has been given in Paper	
3394	No. _________ to charge/credit DEPOSIT ACCOUNT No. _________ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other ____________ ☐ Credit

Request for Continued Examination (RCE) Transmittal

Address to:
Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Application Number	10/259,494
Filing Date	September 30, 2002
First Named Inventor	Edmund C. Munger
Art Unit	2153
Examiner Name	Lim, Krisna
Attorney Docket Number	77580-012 (VRNK-1CPDV3)

This is a Request for Continued Examination (RCE) under 37 CFR 1.114 of the above-identified application.
Request for Continued Education (RCE) practice under 37 CFR 1.114 does not apply to any utility or plant application filed prior to June 8, 1995, or to any design application. See Instruction Sheet for RCEs (not to be submitted to the USPTO) on page 2.

1. Submission required under 37 CFR 1.114 Note: If the RCE is proper, any previously filed unentered amendments and amendments enclosed with the RCE will be entered in which they were filed unless applicant instructs otherwise. If applicant does not wish to have any previously filed unentered amendment(s) entered, applicant must request non-entry of such amendment(s).
   - a. ☐ Previously submitted If a final Office action is outstanding, any amendments filed after the final Office action may be considered as a submission even if this box is not checked.
     - i. ☐ Consider the arguments in the Appeal Brief or Reply Brief previously filed on ______
     - ii. ☐ Other ______
   - b. ☒ Enclosed
     - i. ☒ Amendment/Reply
     - ii. ☐ Affidavit(s)/Declaration(s)
     - iii. ☐ Information Disclosure Statement (IDS)
     - iv. ☐ Other ______
2. Miscellaneous
   - a. ☒ Suspension of action of the above-identified application is requested under 37 CFR 1.103(c) for a period of 1 month. (Period of suspension shall not exceed 3 months; Fee under 37 CFR 1.17(i) required)
   - b. ☐ Other ______
3. Fees The RCE fee under 37 CFR 1.17(e) is required by 37 CFR 1.114 when the RCE is filed.
   - a. ☒ The Director is hereby authorized to charge the following fees, or credit any overpayments, to Deposit Account No. 50-1133.
     - i. ☒ RCE fee required under 37 CFR 1.17(e) $810.00
     - ii. ☒ Extension of time fee (37 CFR 1.136 and 1.17)
     - iii. ☐ Other ______
   - b. ☐ Check in the amount of $ ______ Enclosed
   - c. ☐ Payment by credit card (Form PTO-2038 enclosed)

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

Signature	[signature]	Date	April 11, 2008
Name (Print/Type)	Toby H. Kusmer	Registration No	26,418

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop RCE, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450 or facsimile transmitted to the U.S. Patent and Trademark Office on the date shown below.

Signature	[signature]		
Name(Print/Type)	Jacqueline Andreu	Date	April 11, 2008

BST99 1569194-1 077580.0012

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Edmund Colby Munger et al.
Serial Number:	10/259,494
Filing Date:	30 September 2002
Title:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST
Examiner:	Krisna Lim
Art Unit:	2153
Docket No.:	77580-012 (VRNK- 1CPDV3)
Confirmation No.:	5257

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Mail Stop RCE, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted (571-273-8300) or filed via EFS-Web to the USPTO, on the date indicated below.

Date: April 11, 2008

Jacqueline Andreu
Jacqueline Andreu

Mail Stop RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT "E"

This paper is filed concurrently with the filing of an RCE for the above-identified application.

The Applicants appreciate the Examiner's thorough examination of the subject application, and request further examination in view of the following:

Amendments to the Claims, in the claim listing beginning on page 2 of this paper; and

Remarks, beginning on page 6 of this paper.

AMENDMENTS TO THE CLAIMS

Please cancel claims 1-10, 21-26, 33-38-68, without prejudice, amend claims 17 and 18, and add new claims 69-72. This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1.-10. (Canceled)

11. (Original) A data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client and, for each intercepted DNS request, performs the steps of:

(i) determining whether the intercepted DNS request corresponds to a secure server;

(ii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer, and

(iii) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

12. (Original) The data processing device of claim 11, wherein step (iii) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

13. (Original) The data processing device of claim 12, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

14. (Original) The data processing device of claim 13, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

15. (Canceled)

16. (Original) A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) intercepting a DNS request sent by a client;

(ii) determining whether the intercepted DNS request corresponds to a secure server;

(iii) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iv) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.

17. (Presently Amended) The computer readable medium of claim 16, wherein step ~~(iii)~~(iv) comprises the steps of:

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.

18. (Presently Amended) The computer readable medium of claim 17, wherein step ~~(iii)~~(iv) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

19. (Original) The computer readable medium of claim 18, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

20.-26. (Cancelled)

27. (Previously Presented) The data processing device of claim 11, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

28. (Previously Presented) The data processing device of claim 11, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

29.-30. (Canceled)

31. (Previously Presented) The computer readable medium of claim 16, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

32. (Previously Presented) The computer readable medium of claim 16, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

33.-68. (Canceled)

69. (New) A computer readable medium storing a domain name server (DNS) module comprised of computer readable instructions that, when executed, cause a data processing device to perform the steps of:

(i) determining whether a DNS request sent by a client corresponds to a secure server;

(ii) when the DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer; and

(iii) when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel between the client and the secure server.

70. (New) The computer readable medium of claim 69, wherein step (iii) comprises the steps of

(a) determining whether the client is authorized to access the secure server; and

(b) when the client is authorized to access the secure server, sending a request to the secure server to establish a secure channel between the secure server and the client.

71. (New) The computer readable medium of claim 70, wherein step (iii) further comprises the step of:

(c) when the client is not authorized to access the secure server, returning a host unknown error message to the client.

72. (New) The computer readable medium of claim 71, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

REMARKS

Claims 11-14, 16-19, 27, 28, 31, 32 and 69-72 remain in the application. Claims 1-10, 21-26, 33-68 have been canceled, claims 17 and 18 amended and claims 69-72 have been added.

The Examiner has indicated that claims 11-14, 16-19, 27, 28, 31 and 32 are allowed. Claims 17 and 18 have been amended to correct obvious errors. New claims 69-72 are believed to be allowable. Applicants are canceling the rejected claims, 1-10, 21-26, 33-68, without prejudice, with the intent of filing a continuation application to pursue those claims. Accordingly, the application is now considered to be in condition for allowance. An early and favorable action thereon is therefore earnestly solicited.

Authorization is hereby given to charge our deposit account, No. 50-1133, for the fees corresponding to a Petition for Extension of Time under 37 CFR § 1.136, and for any other fees that may be required for the prosecution of the subject application.

If a telephone conference will expedite prosecution of the application, the Examiner is invited to telephone the undersigned.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Date: April 11, 2008

Toby H. Kusmer, P.C.
Reg. No. 26,418
Attorney for Applicants
28 State Street
Boston, MA 02109-1775
DD Telephone: (617) 535-4065
Facsimile: (617)535-3800
e-mail: tkusmer@mwe.com

BST99 1569210-1 077580 0012

PTO/SB/80 (01-06)
Approved for use through 12/31/2008. OMB 0651-0035
U S. Patent and Trademark Office; U S DEPARTMENT OF COMMERCE

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(b).

I hereby appoint:

☒ Practitioners associated with the Customer Number: 23,630

OR

☐ Practitioner(s) named below (if more then ten practitioners are to be named, then a customer number must be used):

Name	Registration Number	Name	Registration Number

as attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(b).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(b) to:

☒ The address associated with Customer Number: 23,630

OR

☒ Firm or Individual Name	McDermott Will & Emery LLP		
Address	28 State Street		
City	Boston	State MA	Zip 02109
Country	U.S.A.		
Telephone	(617) 535-4065	Email tkusmer@mwe.com	

Assignee Name and Address:
VIRNETX, INC.
5615 SCOTTS VALLEY DRIVE, SUITE 110
SCOTTS VALLEY, CALIFORNIA 95066

A copy of this form, together with a statement under 37 CFR 3.73(b) (Form PTO/SB/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(b) may be completed by one of the practitioners appointed in this form if the appointed practitioner is authorized to act on behalf of the assignee, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee

Signature	[signature]	Date	6/19/07
Name	Kendall Larsen	Telephone	831.608.5698
Title	President		

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/96 (12-07)
Approved for use through 12/31/2007. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: **VIRNETX, INC.**

Application No./Patent No.: **10/259,494** Filed/Issue Date: **September 30, 2002**

Entitled: **ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICES (DNS) REQUEST**

VIRNETX, INC. , a **CORPORATION**
(Name of Assignee) (Type of Assignee, e g , corporation, partnership, university, government agency, etc)

states that it is:

1. [√] the assignee of the entire right, title, and interest; or

2. [ ] an assignee of less than the entire right, title and interest
(The extent (by percentage) of its ownership interest is ________ %)

in the patent application/patent identified above by virtue of either:

A. [ ] An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel ________, Frame ________, or for which a copy thereof is attatched.

OR

B. [√] A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

1. From: Edmund C. Munger, et al. To: Science Applications International Corporation
The document was recorded in the United States Patent and Trademark Office at
Reel 015855, Frame 0419, or for which a copy thereof is attached.

2. From: Science Applications International Corporation To: VirnetX, Inc.
The document was recorded in the United States Patent and Trademark Office at
Reel 018757, Frame 0326, or for which a copy thereof is attached.

3. From: N/A To: ________
The document was recorded in the United States Patent and Trademark Office at
Reel ________, Frame ________, or for which a copy thereof is attached.

[ ] Additional documents in the chain of title are listed on a supplemental sheet.

[ ] As required by 37 CFR 3.73(b)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.
[NOTE: A separate copy (*i.e.*, a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Toby H Kusmer (signature) — Signature | 4.11.08 — Date

TOBY KUSMER — Printed or Typed Name | 617.535.4065 — Telephone number

ATTORNEY IN FACT — Title

Electronic Patent Application Fee Transmittal

Application Number:	10259494
Filing Date:	30-Sep-2002
Title of Invention:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST
First Named Inventor/Applicant Name:	Edmund Colby Munger
Filer:	Toby H. Kusmer./Jacqueline Andreu
Attorney Docket Number:	000479.00082

Filed as Large Entity

Utility Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Miscellaneous:				
Request for continued examination	1801	1	810	810
Processing Fee, except for Provis. apps	1808	1	130	130
		Total in USD ($)		**940**

Electronic Acknowledgement Receipt

EFS ID:	3141021
Application Number:	10259494
International Application Number:	
Confirmation Number:	5257
Title of Invention:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST
First Named Inventor/Applicant Name:	Edmund Colby Munger
Customer Number:	22907
Filer:	Toby H. Kusmer./Jacqueline Andreu
Filer Authorized By:	Toby H. Kusmer.
Attorney Docket Number:	000479.00082
Receipt Date:	11-APR-2008
Filing Date:	30-SEP-2002
Time Stamp:	15:05:32
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$940
RAM confirmation Number	327
Deposit Account	501133
Authorized User	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)

1	Request for Continued Examination (RCE)	RCE.pdf	162351 c428ccfd204385c5333096eb6f7f7f5780a7ca29	no	1
Warnings:					
This is not a USPTO supplied RCE SB30 form.					
Information:					
2	Amendment after Notice of Allowance (Rule 312)	Amendment.pdf	439250 6a2ac83610fef8ae489369bafe2bf4da5c23d9ba	no	6
Warnings:					
Information:					
3	Power of Attorney	POA.pdf	180187 412c368258cb703b6da7bfebd3d330e892a34a83	no	1
Warnings:					
Information:					
4	CRF Statement Paper and CRF are the same	Statement37CFR.pdf	138758 d6fb7e12fc8fecf81827a9b32b0127ce317583ab	no	1
Warnings:					
Information:					
5	Fee Worksheet (PTO-06)	fee-info.pdf	8370 41bc36464aeea4b717aed39be9e0bb444ac0f74c	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	928916		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

PTO/SB/06 (07-06)
Approved for use through 1/31/2007. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number	Filing Date	
10/259,494	09/30/2002	☐ To be Mailed

APPLICATION AS FILED – PART I

FOR	NUMBER FILED (Column 1)	NUMBER EXTRA (Column 2)	SMALL ENTITY ☐ RATE ($)	FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*	X $ =		OR	X $ =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*	X $ =			X $ =	
☐ APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							
			TOTAL			TOTAL	

* If the difference in column 1 is less than zero, enter "0" in column 2.

APPLICATION AS AMENDED – PART II

AMENDMENT	04/11/2008	CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)	OR	OTHER THAN SMALL ENTITY RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 16	Minus	** 65	= 0	X $ =		OR	X $50=	0
	Independent (37 CFR 1.16(h))	* 3	Minus	***10	= 0	X $ =		OR	X $210=	0
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	0

AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT (Column 1)		HIGHEST NUMBER PREVIOUSLY PAID FOR (Column 2)	PRESENT EXTRA (Column 3)	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=	X $ =		OR	X $ =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	X $ =		OR	X $ =	
	☐ Application Size Fee (37 CFR 1.16(s))									
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.
** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".
*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".
The "Highest Number Previously Paid For" (Total or Independent) is the highest number found in the appropriate box in column 1.

Legal Instrument Examiner:
/SUSAN K. FORD/

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082

CONFIRMATION NO. 5257

POA ACCEPTANCE LETTER

OC000000029389884

23630
MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

Date Mailed: 04/14/2008

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 04/11/2008.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/nhorne/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082

CONFIRMATION NO. 5257

POWER OF ATTORNEY NOTICE

OC000000029389862

22907
BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

Date Mailed: 04/14/2008

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 04/11/2008.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/nhorne/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

23630 7590 04/21/2008

MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

EXAMINER	
LIM, KRISNA	
ART UNIT	**PAPER NUMBER**
2153	

DATE MAILED: 04/21/2008

Please find below and/or attached an Office communication concerning this application or proceeding.

The request for deferral/suspension of action under 37 CFR 1.103 has been approved.

PTO-90C (Rev. 10/03)

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Edmund Colby Munger et al.
Serial Number:	10/259,494
Filing Date:	30 September 2002
Title:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST
Examiner:	Krisna Lim
Art Unit:	2153
Docket No.:	77580-012 (VRNK- 1CPDV3)
Confirmation No.:	5257

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the U.S. Postal Service as first class mail in an envelope addressed to: Mail Stop RCE, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted (571-273-8300) or filed via EFS-Web to the USPTO, on the date indicated below.

Date: June 12, 2008

Jacqueline J. Andreu

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

SUPPLEMENTAL AMENDMENT "F"

This paper is filed as a supplement to the response filed on April 11, 2008 concurrently with the filing of an RCE for the above-identified application.

The Applicants appreciate the Examiner's thorough examination of the subject application, and request further examination in view of the following:

Amendment to the Specification, on page 2 of this paper; and

Remarks, beginning on page 3 of this paper.

AMENDMENT TO THE SPECIFICATION

On page 1 of the specification, following paragraph [01], please insert the following title and paragraph:

GOVERNMENT CONTRACT RIGHTS

This invention was made with Government support under Contract No. 360000-1999-000000-QC-000-000 awarded by the Central Intelligence Agency. The Government has certain rights in the invention.

REMARKS

Claims 11-14, 16-19, 27, 28, 31, 32 and 69-72 remain in the application. Please see the response filed April 11, 2008.

The specification has been amended to refer to certain contract rights retained by the Central Intelligence Agency of the United States Government. No new matter has been added. An early and favorable action thereon is therefore earnestly solicited.

Authorization is hereby given to charge our deposit account, No. 50-1133, for the fees corresponding to a Petition for Extension of Time under 37 CFR § 1.136, and for any other fees that may be required for the prosecution of the subject application.

If a telephone conference will expedite prosecution of the application, the Examiner is invited to telephone the undersigned.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Date: June 12, 2008

Toby H. Kusmer, P.C.
Reg. No. 26,418
Attorney for Applicants
28 State Street
Boston, MA 02109-1775
DD Telephone: (617) 535-4065
Facsimile: (617)535-3800
e-mail: tkusmer@mwe.com

BST99 1574513-1 077580 0012

Electronic Acknowledgement Receipt

EFS ID:	3446362
Application Number:	10259494
International Application Number:	
Confirmation Number:	5257
Title of Invention:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST
First Named Inventor/Applicant Name:	Edmund Colby Munger
Customer Number:	23630
Filer:	Toby H. Kusmer./Jacqueline Andreu
Filer Authorized By:	Toby H. Kusmer.
Attorney Docket Number:	000479.00082
Receipt Date:	12-JUN-2008
Filing Date:	30-SEP-2002
Time Stamp:	15:23:54
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1	Supplemental Response or Supplemental Amendment	SupplementalAmendmentF.pdf	168187 e1cfce507a193d0d96f9244fab57b7631fbbc794	no	3

Warnings:

Information:

Total Files Size (in bytes):	168187

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

23630 7590 07/01/2008

MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

EXAMINER	
LIM, KRISNA	
ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 07/01/2008

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

TITLE OF INVENTION: ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1440	$300	$0	$1740	10/01/2008

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

23630 7590 07/01/2008

MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

TITLE OF INVENTION: ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1440	$300	$0	$1740	10/01/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-225000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).
   - ❑ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.
   - ❑ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
   (1) the names of up to 3 registered patent attorneys or agents OR, alternatively,
   (2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.
   1__________________________
   2__________________________
   3__________________________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. The following fee(s) are submitted:
- ❑ Issue Fee
- ❑ Publication Fee (No small entity discount permitted)
- ❑ Advance Order - # of Copies __________________

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ❑ A check is enclosed.
- ❑ Payment by credit card. Form PTO-2038 is attached.
- ❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number ____________ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)
- ❑ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ❑ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ______________________________ Date __________________________

Typed or printed name ______________________________ Registration No. ________________________

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	000479.00082	5257

23630 7590 07/01/2008

MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

EXAMINER	
LIM, KRISNA	
ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 07/01/2008

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 308 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 308 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.

Notice of Allowability	**Application No.** 10/259,494	**Applicant(s)** MUNGER ET AL.	
	Examiner Krisna Lim	**Art Unit** 2153	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to *the RCE filed 4/11/08*.

2. ☒ The allowed claim(s) is/are *11-14, 16-19, 27-28, 31-32 and 69-72*.

3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☐ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☐ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

/Krisna Lim/
Primary Examiner, Art Unit 2153

Index of Claims	Application/Control No. 10259494	Applicant(s)/Patent Under Reexamination MUNGER ET AL.
	Examiner Krisna Lim	Art Unit 2153

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	06/20/2008								
	1	-								
	2	-								
	3	-								
	4	-								
	5	-								
	6	-								
	7	-								
	8	-								
	9	-								
	10	-								
1	11	=								
2	12	=								
3	13	=								
4	14	=								
	15	-								
7	16	=								
8	17	=								
9	18	=								
10	19	=								
	20	-								
	21	-								
	22	-								
	23	-								
	24	-								
	25	-								
	26	-								
5	27	=								
6	28	=								
	29	-								
	30	-								
11	31	=								
12	32	=								
	33	-								
	34	-								
	35	-								
	36	-								

Index of Claims	Application/Control No. 10259494	Applicant(s)/Patent Under Reexamination MUNGER ET AL.
	Examiner Krisna Lim	Art Unit 2153

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☐ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	06/20/2008								
	37	-								
	38	-								
	39	-								
	40	-								
	41	-								
	42	-								
	43	-								
	44	-								
	45	-								
	46	-								
	47	-								
	48	-								
	49	-								
	50	-								
	51	-								
	52	-								
	53	-								
	54	-								
	55	-								
	56	-								
	57	-								
	58	-								
	59	-								
	60	-								
	61	-								
	62	-								
	63	-								
	64	-								
	65	-								
	66	-								
	67	-								
	68	-								
13	69	=								
14	70	=								
15	71	=								
16	72	=								

U.S. Patent and Trademark Office Part of Paper No. : 20080620

Index of Claims	Application/Control No. 10259494	Applicant(s)/Patent Under Reexamination MUNGER ET AL.
	Examiner Krisna Lim	**Art Unit** 2153

✓	**Rejected**	-	**Cancelled**	N	**Non-Elected**	A	**Appeal**
=	**Allowed**	÷	**Restricted**	I	**Interference**	O	**Objected**

Issue Classification	Application/Control No. 10259494	Applicant(s)/Patent Under Reexamination MUNGER ET AL.
	Examiner Krisna Lim	Art Unit 2153

ORIGINAL		INTERNATIONAL CLASSIFICATION	
CLASS	SUBCLASS	CLAIMED	NON-CLAIMED
709	225	G 0 6 F 15 / 173 (2006.01.01)	

CROSS REFERENCE(S)

CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)				
709	229				

NONE (Assistant Examiner) (Date)		Total Claims Allowed: 16
/Krisna Lim/ 6/20/08 (Primary Examiner) (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 1

U.S. Patent and Trademark Office Part of Paper No. 20080620

Search Notes	Application/Control No. 10259494	Applicant(s)/Patent Under Reexamination MUNGER ET AL.
	Examiner Krisna Lim	**Art Unit** 2153

SEARCHED

Class	Subclass	Date	Examiner
709	225, 229	6/20/08	KL

SEARCH NOTES

Search Notes	Date	Examiner
EAST, INVENTORS	6/20/08	KL

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner
709	225, 229	6/20/08	KL

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	2446154	(DNS encrypted channel)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 16:07
L2	191	(establish$4 or creat$3) adj4 (encrypt$3 adj4 channel)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 16:10
L3	191	l1 and l2	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 16:16
L4	12	l3 and @ad<="19981030"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 16:16
L5	2163841	secure server	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 16:32
L6	461453	l1 and l5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 16:33
L7	145870	l6 and (DNS encrypted channel).ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 16:35
L8	63736	l7 and (secure server cient).ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 16:35

L9	34841	l8 and (domain name server)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 16:36
L10	20759	l9 and (DNS request)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 16:38
L11	36984	709/217-229.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 17:07
L12	2400	l11 and l10	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 17:07
L13	2400	l12 and (encrypted channel client server)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 17:08
L14	2395	l12 and (encrypted channel client server).ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 17:08
L15	2395	l14 and l11	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 17:08
L16	298	l15 and @ad<="19981030"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 17:09
L17	223	l16 and (DNS request).ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 17:10

L18	223	l17 and (secure server).ti,ab, clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 17:10
L19	216	l18 and (client server).ti,ab, clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/06/20 17:10

6/ 20/ 08 5:11:57 PM
C:\ Program Files\ USPTO\ EAST\ Bin\ default.wsp

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or Fax (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

23630 7590 07/01/2008

MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	09/30/2002	Edmund Colby Munger	77580-012 (VRNK-1CPDV3)	5257

TITLE OF INVENTION: ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

APPLN. TYPE	SMALL ENTITY	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1440	$300	$0	$1740	10/01/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-225000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 McDermott Will & Emery
2 ______
3 ______

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: VIRNETX INC.

(B) RESIDENCE: (CITY and STATE OR COUNTRY): Scotts Valley Drive, CA

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☑ Corporation or other private group entity ☐ Government

4a. The following fee(s) are submitted:
- ☑ Issue Fee
- ☑ Publication Fee (No small entity discount permitted)
- ☐ Advance Order - # of Copies ______

4b. Payment of Fee(s): **(Please first reapply any previously paid issue fee shown above)**
- ☐ A check is enclosed.
- ☐ Payment by credit card. Form PTO-2038 is attached.
- ☑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment, to Deposit Account Number 501133.

5. **Change in Entity Status** (from status indicated above)
- ☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.
- ☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature /Toby H. Kusmer/ Date August 1, 2008

Typed or printed name Toby H. Kusmer Registration No. 26,418

Electronic Patent Application Fee Transmittal

Application Number:	10259494
Filing Date:	30-Sep-2002
Title of Invention:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST
First Named Inventor/Applicant Name:	Edmund Colby Munger
Filer:	Toby H. Kusmer./Kelly Ciarmataro
Attorney Docket Number:	77580-012 (VRNK-1CPDV3)

Filed as Large Entity

Utility Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl issue fee	1501	1	1440	1440
Publ. Fee- early, voluntary, or normal	1504	1	300	300

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**1740**

Electronic Acknowledgement Receipt

EFS ID:	3716272
Application Number:	10259494
International Application Number:	
Confirmation Number:	5257
Title of Invention:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST
First Named Inventor/Applicant Name:	Edmund Colby Munger
Customer Number:	23630
Filer:	Toby H. Kusmer./Kelly Ciarmataro
Filer Authorized By:	Toby H. Kusmer.
Attorney Docket Number:	77580-012 (VRNK-1CPDV3)
Receipt Date:	01-AUG-2008
Filing Date:	30-SEP-2002
Time Stamp:	13:56:12
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	Deposit Account
Payment was successfully received in RAM	$1740
RAM confirmation Number	7624
Deposit Account	501133
Authorized User	

The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:

Charge any Additional Fees required under 37 C.F.R. Section 1.20 (Post Issuance fees)

Charge any Additional Fees required under 37 C.F.R. Section 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	PartB_PTOL_85_IssueFeeTrans.pdf	100484 726ab61da712c6a499d66afc3bb81ab601d2a932	no	1
Warnings:					
Information:					
2	Fee Worksheet (PTO-06)	fee-info.pdf	8363 c874efee0a2f8ebe0eed0d90db5a899eb44e49e0	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	108847		

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

<u>New Applications Under 35 U.S.C. 111</u>
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

<u>National Stage of an International Application under 35 U.S.C. 371</u>
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

<u>New International Application Filed with the USPTO as a Receiving Office</u>
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/259,494	02/10/2009	7490151	77580-012 (VRNK-1CPDV3)	5257

23630 7590 01/21/2009
MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

ISSUE NOTIFICATION

The projected patent number and issue date are specified above.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 818 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (571)-272-4200.

APPLICANT(s) (Please see PAIR WEB site http://pair.uspto.gov for additional applicants):

Edmund Colby Munger, Crownsville, MD;
Robert Dunham Short III, Leesburg, VA;
Victor Larson, Fairfax, VA;
Michael Williamson, South Riding, VA;

IR103 (Rev. 11/05)

AO 120 (Rev. 3/04)

TO: Mail Stop 8 Director of the U.S. Patent and Trademark Office P.O. Box 1450 Alexandria, VA 22313-1450	REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas on the following ☑ Patents or ☐ Trademarks:

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
6:10-cv-417	8/11/2010	Eastern District of Texas

PLAINTIFF	DEFENDANT
VirnetX Inc.,	Aastra USA, Inc., Aastra Technologies Ltd., Apple, Inc., Cisco Systems, Inc., NEC Corporation, and NEC Corporation of America

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 6,502,135	12/31/2002	VirnetX Inc.
2 6,839,759	1/4/2005	VirnetX Inc.
3 7,188,180	3/6/2007	VirnetX Inc.
4 7,418,504	8/26/2008	VirnetX Inc.
5 7,490,151	2/10/2009	VirnetX Inc.

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy

OPAP JAN 1 8 2011 IAP6 PATENT & TRADEMARK OFFICE

#10259494

IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

In re patent of
VirnetX Inc.
Patent No. 7,490,151
Issued: February 10, 2009
For: Establishment of a secure communication link based on a domain name service (DNS) request

Submission of Prior Art Under 37 CFR 1.501

Hon. Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The undersigned herewith submits in the above-identified patent the following prior art which is pertinent and applicable to the patent and is believed to have a bearing on the patentability of at least claim 1 thereof:

Valencia U.S. 6,308,213 October 23, 2001

The reference discloses a method for creating a secure dial-up session from a remote client to a local network through an internet service provider strikingly similar to the device of VirnetX Inc. It is believed that the reference has a bearing on the patentability of at least claim 1 of the VirnetX Inc. patent.

Insofar as claim 1 is concerned, the reference clearly anticipates the claimed subject matter under 35 U.S.C. 102.

Below is a list of other references which affect one or more of the claims in the patent.

US6377993	US6330602	US6308213	US6263368	US6208656	US6185619	US6173411	US6173311
US6154775	US6130892	US6088796	US6032184	US6023724	US6006264	US6003084	US5983350
US5982891	US5974453	US5968176	US5968133	US5950195	US5948054	US5944783	US5935245
US5918018	US5892900	US5884025	US5864683	US5757924	US5727146	US5651002	US5638448
US5615340	US5550984	US5508997	US5400334	US7861166	US7853723	US57849393	US7849105
US7844743	US7844706	US7837101	US7836481	US7835989	US7835344	US7831823	US7831722
US7831477	US7830860	US7827291	US7821926	US7818371	US7817619	US7814533	US7814169
US7813332	US7809847	US7809644	US7805399	US7804816	US7802718	US7797423	US7793830
US7788182	US7778396	US7774501	US7770196	US7761585	US7756190	US7752649	US7734789
US7730299	US7730190	US7720076	US7716349	US7715371	US7706332	US7702540	US7699220
US7698567	US7694024	US7680879	US7673072	US7664871	US7664097	US7647243	US7631193
US7631188	US7630861	US7627684	US7627001	US7624180	US7623932	US7620726	US7620605
US7617973	US7617527	US7613659	US7613633	US7606401	US7602782	US7600677	US7597251
US7597248	US7593999	US7593453	US7591420	US7586939	US7584260	US7583971	US7583668

US7583665	US7580919	US7580715	US7575158	US7571850	US7558407	US7554995	US7546251
US7539942	US7526644	US7523072	US7522542	US7515712	US7509360	US7509270	US7502869
US7502406	US7496198	US7486660	US7484004	US7483417	US7477900	US7475156	US7475137
US7472156	US7462746	US7461160	US7454709	US7451193	US7443858	US7424737	US7415617
US7404014	US7401286	US7392395	US7389270	US7386880	US7380273	US7366900	US7360244
US7359881	US7359137	US7342581	US7337233	US7336788	US7333485	US7325063	US7315893
US7307956	US7302473	US7299501	US7298851	US7296283	US7287271	US7281133	US7277248
US7272625	US7269577	US7269576	US7266529	US7265927	US7260556	US7260518	US7254518
US7251784	US7249378	US7249376	US7248719	US7248588	US7246148	US7236486	US7236320
US7225249	US7225160	US7224798	US7218625	US7210035	US7209902	US7200574	US7187686
US7181613	US7165174	US7152045	US7149775	US7149771	US7149208	US7146417	US7145898
US7143438	US7143290	US7139736	US7136903	US7136359	US7133940	US7133846	US7133845
US7124302	US7123600	US7120802	US7120800	US7116669	US7114083	US7113912	US7113508
US7103007	US7100199	US7099313	US7099308	US7095854	US7092394	US7088871	US7088727
US7085610	US7076652	US7073056	US7072948	US7072380	US7069451	US7065505	US7062781
US7062500	US7061900	US7058954	US7058720	US7058696	US7058606	US7058600	US7051212
US7047415	US7043453	US7042900	US7039802	US7039679	US7039065	US7039008	US7031964
US7031471	US7031309	US7028088	US7024392	US7023875	US7020700	US7020111	US7017046
US7010702	US7003587	US6983327	US6983009	US6978017	US6971008	US6968571	US6963923
US6957427	US6956845	US6945457	US6944657	US6944600	US6940859	US6940840	US6938263
US6937729	US6937703	US6925644	US6922412	US6917966	US6915329	US6912222	US6901509
US6891553	US6889181	US6886035	US6883034	US6874090	US6874026	US6865551	US6859783
US6850484	US6839770	US6836548	US6832252	US6832223	US6831923	US6829231	US6823412
US6823140	US6819663	US6816966	US6816875	US6816473	US6804743	US6801522	US6798873
US6798776	US6792471	US6791993	US6791979	US6788662	US6782427	US6775692	US6772332
US6766454	US6766416	US6763376	US6760747	US6760736	US6757286	US6754601	US6754212
US6754181	US6748082	US6745229	US6744892	US6742040	US6741909	US6738981	US6732179
US6731625	US6728737	US6728242	US6721301	US6718319	US6714979	US6714921	US6711241
US6711171	US6708262	US6708171	US6708157	US6707805	US6704756	US6701377	US6701370
US6697836	US6697350	US6690654	US6687731	US6683876	US6681213	US6678284	US6677968
US6675195	US6672775	US6671741	US6668278	US6665733	US6662205	US6661779	US6658568
US6657956	US6654793	US6654697	US6650631	US6647418	US6640243	US6633572	US6631402
US6618484	US6618353	US6611528	US6606378	US6603857	US6603763	US6601005	US6598081
US6594257	US6590894	US6587474	US6584094	US6578088	US6578079	US6577734	US6571290
US6564240	US6560340	US6560236	US6553002	US6546011	US6542897	US6542830	US6542569
US6539267	US6539015	US6538988	US6535517	US6532540	US6532392	US6529517	US6526508
US6519675	US6519615	US6519546	US6519224	US6516412	US6515996	US6515968	US6513116
US6513061	US6512766	US6510519	US6510461	US6510151	US6505302	US6505301	US6504844
US6501767	US6501753	US6487195	US6484244	US6480893	US6473803	US6473407	US6473406
US6470386	US6470340	US6467091	US6466780	US6463475	US6463443	US6463057	US6457039
US6456716	US6456608	US6456594	US6453350	US6453345	US6452925	US6449648	US6449574
US6449367	US6449344	US6446164	US6445710	US6445704	US6445703	US6442689	US6442170
US6438127	US6434504	US6427140	US6424992	US6424717	US6424627	US6418466	US6415335
US6414952	US6412074	US6408367	US6408341	US6405272	US6405253	US6400371	US6397114
US6396831	US6393526	US6393486	US6393270	US6389480	US6389402	US6389004	US6385644
US6381650	US6378028	US6377992	US6377859	US6377691	US6370627	US6370580	US6370448
US6370137	US6367016	US6366958	US6366558	US6363488	US6360256	US6356934	US6356530
US6353856	US6349323	US6349274	US6347085	US6345303	US6345300	US6345291	US6343072
US6339830	US6339596	US6336141	US6335927	US6332195	US6330618	US6330608	US6330599
US6330240	US6327596	US6327266	US6327258	US6327251	US6326969	US6324582	US6324287
US6324267	US6324177	US6320878	US6317729	US6314520	US6314456	US6314435	US6314409
US6314102	US6311219	US6311218	US6308148	US6307837	US6304912	US6304908	US6304637
US6304574	US6304546	US6298382	US6298120	US6295551	US6292827	US6292569	US6292568
US6292481	US6289388	US6286106	US6286058	US6286050	US6286045	US6286029	US6285675
US6285658	US6282669	US6282208	US6278992	US6278705	US6278704	US6275941	US6272632
US6272523	US6272150	US6272110	US6269481	US6268789	US6266809	US6266704	US6263444
US6263394	US6260145	US6260073	US6256675	US6256305	US6253239	US6253193	US6252964
US6249787	US6249523	US6247132	US6247129	US6247054	US6247026	US6246767	US6246680
US6243812	US6243751	US6243716	US6243667	US6243394	US6243379	US6243378	US6240513
US6240461	US6240450	US6240185	US6240091	US6240084	US6237786	US6237099	US6237009
US6236660	US6233246	US6233234	US6233232	US6230245	US6230203	US6230194	US6226748
US6223209	US6222822	US6219803	US6219726	US6219715	US6219707	US6219699	US6219421
US6212558	US6212506	US6212192	US6212183	US6212182	US6209098	US6205483	US6205473

US6205456	US6202153	US6202097	US6201962	US6199115	US6199104	US6198724	US6192034
US6189037	US6189032	US6188684	US6185683	US6185601	US6185218	US6185204	US6182224
US6182139	US6182138	US6182116	US6182113	US6182111	US6178455	US6175875	US6175874
US6173399	US6173322	US6172615	US6170057	US6170012	US6163844	US6163843	US6163779
US6163543	US6161180	US6161145	US6161128	US6161126	US6161102	US6157721	US6157641
US6154744	US6151679	US6151628	US6151325	US6148400	US6147991	US6147987	US6145004
US6145001	US6144934	US6144667	US6141749	US6141351	US6138119	US6137869	US6134678
US6134658	US6134591	US6134235	US6131169	US6131116	US6130933	US6128666	US6128664
US6128657	US6128642	US6128298	US6128296	US6122272	US6122258	US6119196	US6119109
US6118760	US6115780	US6115737	US6115458	US6115376	US6115040	US6112242	US6112239
US6112238	US6111883	US6108787	US6108786	US6108782	US6108700	US6108692	US6108655
US6108300	US6105132	US6105131	US6105065	US6104716	US6104704	US6101616	US6101552
US6101189	US6098172	US6098123	US6098111	US6098108	US6097719	US6097718	US6097705
US6096094	US6094715	US6094655	US6094435	US6091951	US6091808	US6091725	US6088728
US6088717	US6088451	US6088368	US6088356	US6084892	US6081807	US6081533	US6081522
US6081512	US6078590	US6075860	US6075858	US6075783	US6073241	US6073202	US6073185
US6073181	US6073178	US6073176	US6073122	US6073105	US6072942	US6072781	US6070243
US6070242	US6070191	US6070184	US6069890	US6067573	US6067561	US6065059	US6065046
US6065002	US6064671	US6064667	US6063129	US6061797	US6061794	US6061734	US6061721
US6061362	US6061357	US6061346	US6061330	US6058478	US6058423	US6058412	US6058381
US6058170	US6058106	US6055638	US6055562	US6055224	US6052819	US6052738	US6052718
US6052629	US6052450	US6049872	US6049825	US6049602	US6049528	US6047376	US6047338
US6046988	US6044418	US6044402	US6044362	US6044224	US6044155	US6044144	US6044087
US6041379	US6041357	US6041356	US6041355	US6041325	US6041041	US6038664	US6038596
US6035414	US6035360	US6035332	US6035324	US6035281	US6035105	US6034680	US6032266
US6031904	US6031836	US6031528	US6029203	US6029182	US6028860	US6028848	US6026499
US6026435	US6026352	US6026293	US6023762	US6023722	US6021496	US6021409	US6021126
US6018768	US6018764	US6018515	US6016388	US6016317	US6016310	US6016305	US6015348
US6014702	US6014698	US6014686	US6014647	US6014441	US6014380	US6012090	US6012066
US6011910	US6011847	US6011797	US6011782	US6009475	US6009467	US6009274	US6009177
US6009173	US6009081	US6006330	US6006268	US6006248	US6003090	US6003087	US6003079
US6003049	US6002675	US5999972	US5999970	US5999965	US5999940	US5999629	US5999525
US5996077	US5996000	US5995726	US5995725	US5995605	US5995489	US5995021	US5991881
US5991876	US5991824	US5991814	US5991810	US5991809	US5991806	US5991746	US5991733
US5991406	US5991302	US5991292	US5991264	US5987523	US5987521	US5987498	US5987497
US5983327	US5983281	US5983275	US5983270	US5983233	US5983176	US5982997	US5982864
US5980078	US5978880	US5978853	US5978813	US5978594	US5978568	US5978360	US5974562
US5974460	US5974441	US5974396	US5970477	US5970467	US5970064	US5969632	US5969433
US5968177	US5968158	US5968116	US5966705	US5966695	US5966528	US5966509	US5966431
US5966163	US5964841	US5963925	US5963746	US5963745	US5963556	US5961644	US5961620
US5961608	US5961602	US5960411	US5960179	US5959990	US5959972	US5958053	US5958052
US5958016	US5958012	US5956714	US5956403	US5954798	US5953732	US5953507	US5953389
US5953005	US5951694	US5951652	US5951651	US5951643	US5949976	US5949883	US5949882
US5949876	US5949874	US5949784	US5949761	US5949753	US5948069	US5946464	US5946295
US5944824	US5944782	US5943422	US5943230	US5941957	US5941947	US5941944	US5940591
US5940396	US5940394	US5938736	US5938732	US5938729	US5937165	US5937163	US5937162
US5933827	US5933606	US5933600	US5933596	US5933591	US5933490	US5933428	US5933142
US5931900	US5930804	US5930786	US5930764	US5930257	US5928351	US5926463	US5926458
US5925109	US5923854	US5923853	US5923016	US5922073	US5922049	US5920861	US5920705
US5920695	US5920566	US5920542	US5919248	US5918021	US5918017	US5918016	US5917997
US5917912	US5917911	US5917821	US5917820	US5915101	US5915087	US5915019	US5915001
US5913041	US5913024	US5910987	US5910955	US5909682	US5909679	US5907717	US5907685
US5907681	US5907680	US5907597	US5905714	US5903651	US5903572	US5901332	US5898780
US5896508	US5894479	US5892922	US5892919	US5892910	US5892909	US5892812	US5889958
US5889953	US5889776	US5884312	US5884284	US5884246	US5884038	US5884033	US5884032
US5884024	US5883948	US5881243	US5881238	US5881237	US5878241	US5878212	US5877759
US5875343	US5875329	US5875236	US5872931	US5872847	US5872783	US5870631	US5870562
US5870558	US5870550	US5870545	US5870386	US5867667	US5867666	US5867665	US5867660
US5867647	US5867495	US5867484	US5864854	US5864843	US5864654	US5862490	US5862344
US5862339	US5862325	US5859852	US5856974	US5855020	US5854901	US5854900	US5854899
US5852812	US5852810	US5852724	US5852607	US5850517	US5850446	US5848399	US5848396
US5848258	US5848233	US5845267	US5845087	US5845080	US5845073	US5845070	US5845067
US5844896	US5844888	US5842043	US5842031	US5841773	US5841468	US5838924	US5838683

US5835727	US5835726	US5835725	US5835720	US5835718	US5835714	US5835712	US5835710
US5835696	US5835084	US5832519	US5832510	US5832222	US5832216	US5832092	US5828894
US5828876	US5828833	US5828753	US5828666	US5828655	US5826269	US5826029	US5825890
US5825871	US5825774	US5825772	US5825769	US5825283	US5822608	US5822531	US5822435
US5822433	US5822319	US5822303	US5819271	US5819225	US5819045	US5819033	US5819028
US5818930	US5818845	US5815723	US5815665	US5815664	US5815080	US5812819	US5812784
US5812775	US5812771	US5812750	US5812671	US5812666	US5812654	US5812552	US5812533
US5809292	US5809235	US5809128	US5808886	US5808671	US5805915	US5805822	US5805820
US5805818	US5805803	US5805785	US5805596	US5805595	US5805591	US5802590	US5802554
US5802367	US5802320	US5802316	US5802306	US5802304	US5802302	US5802291	US5802290
US5802285	US5802283	US5802053	US5802047	US5802043	US5799154	US5799090	US5799016
US5799002	US5796952	US5796951	US5796944	US5796824	US5796718	US5796393	US5794059
US5793983	US5793978	US5793965	US5793964	US5793763	US5793762	US5793759	US5793694
US5790809	US5790806	US5790805	US5790800	US5790797	US5790789	US5790780	US5787412
US5787160	US5787070	US5784566	US5784547	US5784464	US5784003	US5781743	US5781632
US5781551	US5781550	US5781534	US5778419	US5778377	US5778178	US5778174	US5777989
US5774689	US5774670	US5774668	US5774640	US5771459	US5771353	US5771291	US5768528
US5768501	US5768280	US5768271	US5765015	US5765012	US5764935	US5764915	US5764909
US5764789	US5764756	US5761669	US5761523	US5758087	US5758083	US5757916	US5757900
US5754871	US5754830	US5754803	US5754774	US5754657	US5754656	US5754646	US5752260
US5752078	US5752067	US5751971	US5751954	US5751707	US5751706	US5751287	US5748901
US5748893	US5748871	US5748736	US5748633	US5748470	US5745754	US5745701	US5745676
US5745576	US5745573	US5745555	US5742905	US5742768	US5742763	US5742762	US5742760
US5742686	US5742682	US5740402	US5740362	US5740248	US5740158	US5737525	US5737331
US5737316	US5734921	US5734867	US5734865	US5734831	US5734709	US5734654	US5732406
US5732214	US5732213	US5732133	US5729689	US5729682	US5729452	US5727175	US5727155
US5727147	US5727129	US5727061	US5724510	US5724492	US5724355	US5721913	US5721908
US5721876	US5721780	US5720025	US5717944	US5717943	US5717686	US5715399	US5713037
US5712981	US5712903	US5712897	US5710935	US5710884	US5710883	US5710882	US5710814
US5708836	US5708780	US5708659	US5708655	US5708654	US5706507	US5706502	US5706437
US5706427	US5706277	US5701465	US5699532	US5699528	US5699521	US5699513	US5699500
US5699403	US5696906	US5696898	US5696825	US5696702	US5694546	US5694472	US5694335
US5692181	US5692126	US5692049	US5692030	US5689688	US5689645	US5689564	US5687235
US5685004	US5684950	US5684451	US5682553	US5682478	US5680461	US5680456	US5680437
US5678045	US5678041	US5678006	US5675782	US5675732	US5675723	US5673322	US5671365
US5671354	US5671279	US5668992	US5668952	US5668876	US5668857	US5666530	US5666486
US5666481	US5664199	US5664185	US5661803	US5661719	US5659601	US5659542	US5659350
US5657452	US5657450	US5657390	US5657320	US5657314	US5655077	US5654886	US5654746
US5654531	US5651066	US5650994	US5649182	US5649099	US5644751	US5644733	US5644720
US5644706	US5644571	US5642515	US5640504	US5638515	US5636371	US5636216	US5636210
US5634099	US5634074	US5634015	US5633869	US5633371	US5632029	US5632011	US5630162
US5630081	US5630066	US5625836	US5625622	US5623605	US5623601	US5623600	US5623492
US5621889	US5621727	US5619716	US5619574	US5619498	US5617577	US5617547	US5617540
US5617421	US5615277	US5614891	US5613204	US5613096	US5612959	US5612958	US5612897
US5612865	US5612730	US5611075	US5610981	US5610915	US5610910	US5608908	US5608786
US5608738	US5608726	US5608446	US5606668	US5604807	US5604803	US5604729	US5604528
US5603029	US5602918	US5600820	US5600644	US5598536	US5594918	US5594798	US5594490
US5592669	US5592470	US5590299	US5590285	US5590199	US5588152	US5588059	US5587726
US5586263	US5586260	US5586257	US5586121	US5586112	US5583996	US5583933	US5583868
US5581559	US5577209	US5577196	US5574859	US5572643	US5572640	US5572533	US5570466
US5570361	US5570360	US5568613	US5568487	US5568471	US5568301	US5566351	US5566170
US5564001	US5563805	US5561709	US5560013	US5559986	US5558339	US5557747	US5557742
US5555416	US5555304	US5555290	US5553287	US5553239	US5551052	US5551025	US5550816
US5550551	US5548756	US5548726	US5548724	US5548721	US5548649	US5548646	US5546389
US5544356	US5544340	US5544325	US5544322	US5542046	US5541919	US5539883	US5539734
US5538255	US5537611	US5537592	US5537535	US5537417	US5537099	US5535403	US5535336
US5535217	US5535206	US5534913	US5533108	US5533033	US5533029	US5530852	US5530758
US5530744	US5528763	US5528691	US5526489	US5526257	US5524238	US5524227	US5522077
US5521925	US5521923	US5519858	US5519704	US5517618	US5517567	US5517502	US5517488
US5515508	US5515418	US5515361	US5513346	US5513337	US5509120	US5509006	US5508998
US5506973	US5506961	US5506956	US5506893	US5504921	US5504866	US5504814	US5502766
US5499297	US5495580	US5495533	US5495426	US5493650	US5491800	US5491779	US5491752
US5491693	US5490252	US5490212	US5490208	US5490060	US5488715	US5488412	US5488411

US5485579	US5485577	US5485576	US5485510	US5485465	US5483661	US5483654	US5483631
US5483596	US5481613	US5481604	US5481542	US5479395	US5478993	US5477531	US5476259
US5475836	US5475823	US5475687	US5473692	US5473599	US5471532	US5471399	US5469554
US5469540	US5467452	US5465351	US5465330	US5465291	US5463772	US5463755	US5463752
US5463615	US5459304	US5457786	US5455948	US5455932	US5455865	US5455828	US5455407
US5454111	US5454093	US5453737	US5453601	US5452447	US5452446	US5452420	US5452352
US5452294	US5451923	US5450489	US5448723	US5448045	US5446880	US5446796	US5444705
US5444491	US5442771	US5442749	US5442633	US5442624	US5440723	US5440634	US5440555
US5438568	US5437024	US5437013	US5436909	US5434914	US5432907	US5432783	US5430727
US5430715	US5428615	US5426637	US5426427	US5425092	US5425026	US5423020	US5423006
US5423003	US5423002	US5421024	US5421019	US5420577	US5420405	US5418922	US5418854
US5416842	US5414833	US5414694	US5412717	US5410541	US5408506	US5406643	US5406628
US5406557	US5404562	US5404534	US5404505	US5404404	US5402415	US5402394	US5400335
US5398248	US5396494	US5394408	US5392357	US5392280	US5390336	US5390326	US5390247
US5390181	US5386548	US5386542	US5386413	US5384906	US5384777	US5381541	US5379289
US5379057	US5375219	US5375207	US5373559	US5371877	US5371852	US5371794	US5371494
US5369707	US5369688	US5369571	US5367517	US5363447	US5361259	US5361256	US5359717
US5359660	US5355476	US5355472	US5355453	US5351243	US5349693	US5349686	US5347642
US5347450	US5347304	US5343477	US5341499	US5341496	US5341477	US5341293	US5339356
US5337320	US5337309	US5333183	US5329619	US5327554	US5327486	US5327428	US5327426
US5325504	US5325423	US5325290	US5323146	US5321841	US5319642	US5315591	US5315093
US5313598	US5313465	US5309562	US5309437	US5307490	US5307347	US5305385	US5303234
US5301337	US5301329	US5301247	US5301232	US5299307	US5297242	US5295244	US5295140
US5293488	US5291609	US5291442	US5289585	US5287401	US5287351	US5287270	US5287103
US5285494	US5283856	US5280480	US5280479	US5280477	US5280475	US5278901	US5278833
US5278823	US5276901	US5276789	US5276678	US5276440	US5274631	US5272755	US5271041
US5271003	US5268962	US5263165	US5263158	US5263157	US5263084	US5262760	US5261102
US5261070	US5261064	US5261044	US5251324	US5251205	US5249290	US5247676	US5245606
US5245533	US5241677	US5241599	US5241594	US5241587	US5239648	US5239540	US5239537
US5237611	US5235642	US5235619	US5230020	US5228076	US5227782	US5227778	US5226172
US5226120	US5224163	US5224099	US5223699	US5222140	US5221838	US5220655	US5220604
US5220603	US5220516	US5220420	US5218699	US5218637	US5218600	US5216715	US5214767
US5214702	US5214701	US5214390	US5210710	US5208859	US5208858	US5208856	US5208665
US5207254	US5204966	US5201000	US5200949	US5199069	US5195092	US5195089	US5193151
US5193149	US5191611	US5187780	US5185860	US5185796	US5185795	US5182554	US5181107
US5179704	US5179591	US5178246	US5177788	US5175416	US5173938	US5166978	US5166931
US5166930	US5166678	US5164988	US5163154	US5163049	US5161192	US5159685	US5159592
US5157657	US5155590	US5153919	US5153874	US5150411	US5150408	US5150401	US5148479
US5146581	US5146574	US5146498	US5146497	US5144667	US5144664	US5142690	US5142622
US5140634	US5138712	US5136716	US5136707	US5136690	US5136643	US5136642	US5136580
US5134700	US5131041	US5131020	US5123011	US5122794	US5117422	US5115467	US5115466
US5113499	US5111504	US5109403	US5107492	US5105920	US5103476	US5101402	US5095494
US5093921	US5093860	US5091938	US5091851	US5088090	US5086499	US5086426	US5084867
US5084837	US5083265	US5081677	US5077790	US5075771	US5073852	US5065429	US5060263
US5058109	US5056090	US5054589	US5054067	US5051982	US5050213	US5048087	US5041972
US5033084	US5032979	US5031089	US5030806	US5029207	US5027315	US5025491	US5020673
US5018137	US5018096	US5016274	US5014265	US5014125	US5012469	US5012467	US5005011
US5003597	US5003593	US5003591	US5001753	US5001752	US5001628	US4995082	US4995074
US4993067	US4992646	US4991204	US4987571	US4983961	US4982430	US4982324	US4980913
US4980886	US4979100	US4975944	US4972504	US4968873	US4965804	US4965788	US4964120
US4962532	US4962531	US4962449	US4952930	US4949340	US4947390	US4942574	US4935962
US4933970	US4933938	US4933937	US4932056	US4932021	US4930159	US4927789	US4926479
US4926375	US4924303	US4922523	US4920484	US4916737	US4916704	US4914571	US4912762
US4912721	US4907224	US4906828	US4901348	US4899333	US4893248	US4891781	US4888801
US4885777	US4882751	US4881264	US4878218	US4877950	US4876716	US4873517	US4866421
US4864615	US4860201	US4859837	US4858224	US4853950	US4852151	US4851993	US4847837
US4846653	US4843026	US4825354	US4825050	US4823386	US4823338	US4817091	US4817050
US4811393	US4811337	US4811201	US4809265	US4807286	US4804248	US4803725	US4799156
US4799153	US4792973	US4787082	US4782326	US4779193	US4771461	US4771458	US4771391
US4769811	US4769810	US4766402	US4766293	US4757460	US4754395	US4748668	US4748560
US4734907	US4733391	US4722502	US4719616	US4713753	US4712238	US4710926	US4706081
US4694491	US4692918	US4691355	US4689478	US4680753	US4677670	US4672535	US4672533
US4670857	US4658093	US4654842	US4652993	US4652990	US4652698	US4649548	US4644532

US4641304	US4638356	US4635254	US4635194	US4634808	US4633462	US4633434	US4633036
US4630201	US4629872	US4627052	US4627045	US4621362	US4614861	US4613935	US4613901
US4605820	US4601028	US4593282	US4587627	US4586175	US4578531	US4578530	US4577313
US4577224	US4575842	US4575579	US4574350	US4570261	US4567600	US4549308	US4536791
US4533948	US4531929	US4531021	US4529870	US4528589	US4520488	US4491983	US4484025
US4479228	US4475123	US4470114	US4455605	US4454414	US4449181	US4445833	US4440976
US4438493	US4424414	US4420068	US4417249	US4417245	US4413315	US4405829	US4403297
US4403282	US4400770	US4396984	US4376299	US4371929	US4361851	US4345315	US4332027
US4325116	US4309569	US4303904	US4282572	US4277837	US4264782	US4238854	US4228321
US4227253	US4223380	US4218582	US4207609	US4207431	US4200770	US4186380	US4164787
US4160129	US4139737	US4063220	US4034347	US3985962	US3963102	US3956615	US3749845
US3556635	US3540080	US3537552	US3129407	US2331428	US2299873	US20010003828	US1723740
US1485172	USRE40187	USRE39802	USRE36751	USH1641	EP1796013	EP1621961	EP1349043
EP1251653	EP1029292	EP1019833	EP1004076	EP0990206	EP0898777	EP0893796	EP0887979
EP0868050	EP0853406	EP0825784	EP0818007	EP0813133	EP0809387	EP0752674	EP0740871
EP0740439	EP0737907	EP0732654	EP0729256	EP0697662	EP0590861	EP0588415	EP0580350
EP0535863	EP0523386	EP0465016	EP0462540	EP0416943	EP0398645	EP0363122	EP0324277
EP0256768	EP0183273	EP0172670	EP0155762	EP0150688	WO9923538	WO9922485	WO9922278
WO9918515	WO9916209	WO9914907	WO9907007	WO9900737	WO9856135	WO9854871	WO9854644
WO9828924	WO9825372	WO9824207	WO9819472	WO9726735	WO9726734	WO9726731	WO9723988
WO9723972	WO9715008	WO9713340	WO9705727	WO9704410	WO9700471	WO9630840	WO9618269
WO9618160	WO9613774	WO9613113	WO9605549	WO9604741	WO9600485	WO9534970	WO9515526
WO9513583	WO9501023	WO9316538	WO9315581	WO9313481	WO9308545	WO9303562	WO9116691
WO8702155	WO8700373	GB2342020	GB2329499	GB2323258	GB2305747	GB2300544	GB2281793
GB2277181	GB2148563	GB2064920	GB0814589	GB0659112	FR2699700	DE4202852	DE3418234
DE2000776	DE19823666	DE19619886	DE0838930	JP63107254	JP09261265	AU0760742	AU0751942
AU0751212	AU0740012	AU0725712	AU0692872	AU0678937			

Respectfully submitted,

Ray Selig, Esq.
M·CAM, Inc.
210 Ridge-McIntire Road, Suite 300
Charlottesville, VA 22903

Certificate of Service

I hereby certify on this 14th day of January 2011, that a true and correct copy of the forgoing "Submission of Prior Art" was mailed by first-class mail, postage paid, to:

VirnetX Inc..
c/o McDermott Will & Emery
600 13th Street, NW
Washington DC 20005-3096

Ray Selig

PATENT
Customer No. 22,852
Attorney Docket No. 11798.0002-00000

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re *Inter Partes* Reexamination of:	)
Edmund C. MUNGER et al.	) Control No.: 95/001,714
U. S. Patent No. 7,490,151	) Group Art Unit: 3992
Issued: February 10, 2009	) Examiner: Michael J. Yigdall
For: ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST	) Confirmation No. 3428

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

**REVOCATION OF POWER OF ATTORNEY,
STATEMENT UNDER 37 C.F.R. § 3.73(b),
AND GRANT OF NEW POWER OF ATTORNEY
FOR REEXAMINATION CONTROL NO. 95/001,714 ONLY**

The undersigned, a representative authorized to sign on behalf of the assignee owning all of the interest in U.S. Patent No. 7,490,151 ("the '151 patent"), hereby revokes all previous powers of attorney or authorization of agent granted only in the above-mentioned reexamination proceeding, i.e., control no. 95/001,714.

As required by 37 C.F.R. § 3.73(b), the undersigned verifies that Virnetx Inc. is the assignee of the entire right, title, and interest in the '151 patent by virtue of assignments recorded in the U.S. Patent and Trademark Office at Reel/Frame 014878/0169 and 018757/0326.

The undersigned representative of the Assignee hereby grants its power of attorney to the patent practitioners associated with **FINNEGAN, HENDERSON, FARABOW, GARRETT &**

DUNNER, L.L.P., Customer Number 22,852, to prosecute only reexamination proceeding with control no. 95/001,714 and to transact all business in the Patent and Trademark Office connected therewith. Power of attorney and the mailing address for all other aspects of the '151 patent shall remain unchanged.

Please send all future correspondence concerning reexamination proceeding with control no. 95/001,714 to Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., Customer No. 22,852.

Dated: November 1, 2011

By: [signature]
Sameer Mathur
VP of Corporate Development
and Product Marketing
Virnetx Inc.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 5257

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/259,494	09/30/2002 **RULE**	709	2153	77580-012 (VRNK-1CPDV3)

APPLICANTS

Edmund Colby Munger, Crownsville, MD;
Robert Dunham Short III, Leesburg, VA;
Victor Larson, Fairfax, VA;
Michael Williamson, South Riding, VA;

** **CONTINUING DATA** *************************

This application is a DIV of 09/504,783 02/15/2000 PAT 6502135
which is a CIP of 09/429,643 10/29/1999 PAT 7010604
which claims benefit of 60/106,261 10/30/1998
and claims benefit of 60/137,704 06/07/1999

** **FOREIGN APPLICATIONS** ********************

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 10/31/2002

	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ☐ yes ☐ no 35 USC 119 (a-d) conditions met ☐ yes ☐ no ☐ Met after Allowance Verified and Acknowledged ____ Examiner's Signature ____ Initials	MD	35	20	6

ADDRESS
22852

TITLE
ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

FILING FEE RECEIVED		
3694	FEES: Authority has been given in Paper No. _________ to charge/credit DEPOSIT ACCOUNT No. _________ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other ______________ ☐ Credit

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/259,494	09/30/2002	Edmund Colby Munger	77580-012 (VRNK-1CPDV3)

CONFIRMATION NO. 5257

POA ACCEPTANCE LETTER

OC000000050799648

22852
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

Date Mailed: 11/03/2011

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 11/01/2011.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

/jawhitfield/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/259,494	09/30/2002	Edmund Colby Munger	77580-012 (VRNK-1CPDV3)

CONFIRMATION NO. 5257

POWER OF ATTORNEY NOTICE

OC000000050799645

23630
McDermott Will & Emery
600 13th Street, NW
Washington, DC 20005-3096

Date Mailed: 11/03/2011

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 11/01/2011.

- The Power of Attorney to you in this application has been revoked by the assignee who has intervened as provided by 37 CFR 3.71. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

/jawhitfield/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

PATENT
Customer No. 22,852
Attorney Docket Nos. 11798.0002; 11798.0010

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re *Inter Partes* Reexamination of:	)	
	)	
Edmund MUNGER et al.	)	Control Nos.: 95/001,714; 95/001,697
	)	
U. S. Patent No. 7,490,151	)	Group Art Unit: 3992
	)	
Issued: February 10, 2009	)	Examiner: Michael J. Yigdall
	)	
For: ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST	)))	Confirmation Nos. 3428; 2161

Mail Stop *Inter Partes* Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Commissioner:

REVOCATION OF POWER OF ATTORNEY,
STATEMENT UNDER 37 C.F.R. § 3.73(b),
AND GRANT OF NEW POWER OF ATTORNEY

The undersigned, a representative authorized to sign on behalf of the assignee owning all of the interest in U.S. Patent No. 7,490,151 ("the '151 patent"), hereby revokes all previous powers of attorney or authorization of agent granted in the '151 patent before the date of execution hereof.

In compliance with 37 C.F.R. § 3.73(b), the undersigned verifies that VirnetX Inc. is the assignee of the entire right, title, and interest in the '151 patent by virtue of an assignment recorded in the U.S. Patent and Trademark Office at Reel 018757, Frame 0326 on January 10, 2007.

The undersigned representative of the assignee hereby grants its power of attorney to the patent practitioners associated with **Finnegan, Henderson, Farabow, Garrett & Dunner,**

L.L.P., Customer Number 22,852, to transact all business in the Patent and Trademark Office connected with the '151 patent, including the reexamination proceedings assigned control nos. 95/001,714 and 95/001,697, and in any other proceedings involving the '151 patent.

Please also send all future correspondence concerning the '151 patent to the address associated with **Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., Customer Number 22,852.**

Dated: 11/30/12

By: [signature]
Sameer Mathur
Vice President, Corporate Development and Product Marketing
VirnetX Inc.

Electronic Acknowledgement Receipt

EFS ID:	14368557
Application Number:	10259494
International Application Number:	
Confirmation Number:	5257
Title of Invention:	ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST
First Named Inventor/Applicant Name:	Edmund Colby Munger
Customer Number:	23630
Filer:	Joseph Edwin Palys./connie sisk
Filer Authorized By:	Joseph Edwin Palys.
Attorney Docket Number:	77580-012 (VRNK-1CPDV3)
Receipt Date:	03-DEC-2012
Filing Date:	30-SEP-2002
Time Stamp:	15:55:57
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	POA_Patent_151.pdf	56406 8eb7a14fd0164150714f0d5c2051aa3e4c7d5352	no	2

Warnings:

Information:

Total Files Size (in bytes):	56406

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111
If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371
If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office
If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Bib Data Sheet

CONFIRMATION NO. 5257

SERIAL NUMBER	FILING OR 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
10/259,494	09/30/2002 RULE	709	2153	77580-012 (VRNK-1CPDV3)

APPLICANTS

Edmund Colby Munger, Crownsville, MD;
Robert Dunham Short III, Leesburg, VA;
Victor Larson, Fairfax, VA;
Michael Williamson, South Riding, VA;

** **CONTINUING DATA** *************************

This application is a DIV of 09/504,783 02/15/2000 PAT 6502135
which is a CIP of 09/429,643 10/29/1999 PAT 7010604
which claims benefit of 60/106,261 10/30/1998
and claims benefit of 60/137,704 06/07/1999

** **FOREIGN APPLICATIONS** ********************

IF REQUIRED, FOREIGN FILING LICENSE GRANTED
** 10/31/2002

	STATE OR COUNTRY	SHEETS DRAWING	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ☐ yes ☐ no 35 USC 119 (a-d) conditions met ☐ yes ☐ no ☐ Met after Allowance Verified and Acknowledged ______ Examiner's Signature Initials	MD	35	20	6

ADDRESS
22852

TITLE
ESTABLISHMENT OF A SECURE COMMUNICATION LINK BASED ON A DOMAIN NAME SERVICE (DNS) REQUEST

FILING FEE RECEIVED		
3694	FEES: Authority has been given in Paper No. ________ to charge/credit DEPOSIT ACCOUNT No. ________ for following:	☐ All Fees ☐ 1.16 Fees (Filing) ☐ 1.17 Fees (Processing Ext. of time) ☐ 1.18 Fees (Issue) ☐ Other ________ ☐ Credit

AO 120 (Rev. 08/10)

TO: **Mail Stop 8** **Director of the U.S. Patent and Trademark Office** **P.O. Box 1450** **Alexandria, VA 22313-1450**	**REPORT ON THE FILING OR DETERMINATION OF AN ACTION REGARDING A PATENT OR TRADEMARK**

In Compliance with 35 U.S.C. § 290 and/or 15 U.S.C. § 1116 you are hereby advised that a court action has been filed in the U.S. District Court Eastern District of Texas Tyler Division on the following ☐ Trademarks or ☑ Patents. (☐ the patent action involves 35 U.S.C. § 292.):

DOCKET NO.	DATE FILED	U.S. DISTRICT COURT
6:13-cv-00351	4/22/2013	Eastern District of Texas Tyler Division

PLAINTIFF	DEFENDANT
VirnetX Inc. and Science Applications International Corporation	Microsoft Corporation

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 6,502,135	12/31/2002	VirnetX Inc.
2 7,188,180	3/6/2007	VirnetX Inc.
3 7,418,504	8/26/2008	VirnetX Inc.
4 7,490,151	2/10/2009	VirnetX Inc.
5 7,921,211	4/5/2011	VirnetX Inc.

In the above—entitled case, the following patent(s)/ trademark(s) have been included:

DATE INCLUDED	INCLUDED BY ☐ Amendment ☐ Answer ☐ Cross Bill ☐ Other Pleading	
PATENT OR TRADEMARK NO.	**DATE OF PATENT OR TRADEMARK**	**HOLDER OF PATENT OR TRADEMARK**
1 7,987,274		
2		
3		
4		
5		

In the above—entitled case, the following decision has been rendered or judgement issued:

DECISION/JUDGEMENT

CLERK	(BY) DEPUTY CLERK	DATE

Copy 1—Upon initiation of action, mail this copy to Director Copy 3—Upon termination of action, mail this copy to Director
Copy 2—Upon filing document adding patent(s), mail this copy to Director Copy 4—Case file copy