UTILITY PATENT APPLICATION TRANSMITTAL

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

Attorney Docket No.	000479.00111
First Inventor	Victor Larson
Title	An Agile Network Protocol For Secure Communications Using Secure Domain Names
Express Mail Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Commissioner for Patents
Mail Stop Patent Application
P.O. Box 1450
Alexandria VA 22313-1450

10/714849

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 83]
   (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 40]
   ☒ Formal ☐ Informal
5. Oath or Declaration [Total Sheets 6]
   a. ☒ Newly executed (original or copy)
   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 ☐ Power of Attorney *(when there is an assignee)*
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 / 558,210

Prior application information: Examiner Krisna Lim   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 <u>can only</u> be relied upon when a portion has been inadvertently omitted from the submitted application parts.

19. 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	November 18, 2003

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
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.

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 ($) 1036

Complete if Known	
Application Number	TBA
Filing Date	November 18, 2003
First Named Inventor	Victor Larson
Examiner Name	TBA
Art Unit	TBA
Attorney Docket No.	000479.00111

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		Small Entity			
Fee Code	Fee ($)	Fee Code	Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	770
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 filling fee	

SUBTOTAL (1) ($) 770

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

		Extra Claims		Fee from below		Fee Paid
Total Claims	23	-20 ** =	3	X	18	= 54
Independent Claims	5	-3 ** =	2	X	86	= 172
Multiple Dependent				X		= 0

Large Entity		Small Entity		
Fee Code	Fee ($)	Fee Code	Fee ($)	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) ($) 226

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

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity		Small Entity			
Fee Code	Fee ($)	Fee Code	Fee ($)	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)	40
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)** ($) 40

SUBMITTED BY

Complete (if applicable)

Name (Print/Type)	Ross A. Dannenberg	Registration No. (Attorney/Agent)	49,024	Telephone	(202) 824-3153
Signature	*[signature]*			Date	November 18, 2003

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.

PTO/SB/05 (05-03)
Approved for use through 04/30/2003. 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.

UTILITY PATENT APPLICATION TRANSMITTAL

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

Attorney Docket No.	000479.00111
First Inventor	Victor Larson
Title	An Agile Network Protocol For Secure Communications Using Secure Domain Names
Express Mail Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: Commissioner for Patents
Mail Stop Patent Application
P.O. Box 1450
Alexandria VA 22313-1450

10/714849

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 ⌊83⌋]
   (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 ⌊40⌋]
   ☒ Formal ☐ Informal
5. Oath or Declaration [Total Sheets ⌊6⌋]
   a. ☒ Newly executed (original or copy)
   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 ☐ Power of (when there is an assignee) Attorney
11. ☐ English Translation Document (if applicable)
12. ☒ Information Disclosure ☐ Copies of IDS Statement (IDS)/PTO-1449 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 / 558,210

Prior application information: Examiner Krisna Lim 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.

19. 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		Date	November 18, 2003

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 ($) 1036

Complete if Known

Application Number	TBA
Filing Date	November 18, 2003
First Named Inventor	Victor Larson
Examiner Name	TBA
Art Unit	TBA
Attorney Docket No.	000479.00111

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		Small Entity			
Fee Code	Fee ($)	Fee Code	Fee ($)	Fee Description	Fee Paid
1001	770	2001	385	Utility filing fee	770
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 filling fee	

SUBTOTAL (1) ($) 770

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

		Extra Claims		Fee from below		Fee Paid
Total Claims	23	-20 ** =	3	X	18 =	54
Independent Claims	5	-3 ** =	2	X	86 =	172
Multiple Dependent				X	=	0

Large Entity		Small Entity			
Fee Code	Fee ($)	Fee Code	Fee ($)	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) ($) 226

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

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity		Small Entity			
Fee Code	Fee ($)	Fee Code	Fee ($)	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)	40
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) ($) 40

SUBMITTED BY

Complete (if applicable)

Name (Print/Type)	Ross A. Dannenberg	Registration No. (Attorney/Agent)	49,024	Telephone	(202) 824-3153
Signature				Date	November 18, 2003

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.

AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from and is a continuation patent application of co-pending U.S. application serial number 09/558,210, filed April 26, 2000, which is a continuation-in-part patent application of previously-filed U.S. application serial number 09/504,783, filed on 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 patent application of previously-filed U.S. application serial number 09/429,643, filed on October 29, 1999. The subject matter of U.S. application serial number 09/429,643, 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). The present application is also related to U.S. application serial number (Atty Docket No. 479.86838), filed April 26, 2000, and which is incorporated by reference herein.

BACKGROUND OF THE INVENTION

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.

1

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.

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).

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.

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.

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.

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

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

3

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.

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.

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.

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

4

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.

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.

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.

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.

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

5

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.

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.

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.

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.

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.

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.

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.

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.

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

The present invention provides key technologies for implementing a secure virtual Internet by using a new agile network protocol that is built on top of the existing Internet protocol (IP). The secure virtual Internet works over the existing Internet infrastructure, and interfaces with client applications the same way as the existing Internet. The key technologies provided by the present invention that support the secure virtual Internet include a "one-click" and "no-click" technique to become part of the secure virtual Internet, a secure domain name service (SDNS) for the secure virtual Internet, and a new approach for interfacing specific client applications onto the secure virtual Internet. According to the invention, the secure domain name service interfaces with existing applications, in addition to providing a way to register and serve domain names and addresses.

According to one aspect of the present invention, a user can conveniently establish a VPN using a "one-click" or a "no-click" technique without being required to enter user identification information, a password and/or an encryption key for establishing a VPN. The advantages of the present invention are provided by a method for establishing a secure communication link between a first computer and a second computer over a computer network, such as the Internet. In one embodiment, a secure communication mode is enabled at a first computer without a user entering any cryptographic information for establishing the secure communication mode of communication, preferably by merely selecting an icon displayed on the first computer. Alternatively, the secure communication mode of communication can be enabled by entering a command into the first computer. Then, a secure communication link is established between the first computer and a second computer over a computer network based on

the enabled secure communication mode of communication. According to the invention, it is determined whether a secure communication software module is stored on the first computer in response to the step of enabling the secure communication mode of communication. A predetermined computer network address is then accessed for loading the secure communication software module when the software module is not stored on the first computer. Subsequently, the proxy software module is stored in the first computer. The secure communication link is a virtual private network communication link over the computer network. Preferably, the virtual private network can be based on inserting into each data packet one or more data values that vary according to a pseudo-random sequence. Alternatively, the virtual private network can be based on a computer network address hopping regime that is used to pseudorandomly change computer network addresses or other data values in packets transmitted between the first computer and the second computer, such that the second computer compares the data values in each data packet transmitted between the first computer and the second computer to a moving window of valid values. Yet another alternative provides that the virtual private network can be based on a comparison between a discriminator field in each data packet to a table of valid discriminator fields maintained for the first computer.

According to another aspect of the invention, a command is entered to define a setup parameter associated with the secure communication link mode of communication. Consequently, the secure communication mode is automatically established when a communication link is established over the computer network.

The present invention also provides a computer system having a communication link to a computer network, and a display showing a hyperlink for establishing a virtual private network through the computer network. When the hyperlink for establishing the virtual private network is selected, a virtual private network is established over the computer network. A non-standard top-level domain name is then sent over the virtual private network communication to a predetermined computer network address, such as a computer network address for a secure domain name service (SDNS).

The present invention provides a domain name service that provides secure computer network addresses for secure, non-standard top-level domain names. The advantages of the present invention are provided by a secure domain name service for a computer network that includes a portal connected to a computer network, such as the Internet, and a domain name

9

database connected to the computer network through the portal. According to the invention, the portal authenticates a query for a secure computer network address, and the domain name database stores secure computer network addresses for the computer network. Each secure computer network address is based on a non-standard top-level domain name, such as .scom, .sorg, .snet, .snet, .sedu, .smil and .sint.

The present invention provides a way to encapsulate existing application network traffic at the application layer of a client computer so that the client application can securely communicate with a server protected by an agile network protocol. The advantages of the present invention are provided by a method for communicating using a private communication link between a client computer and a server computer over a computer network, such as the Internet. According to the invention, an information packet is sent from the client computer to the server computer over the computer network. The information packet contains data that is inserted into the payload portion of the packet at the application layer of the client computer and is used for forming a virtual private connection between the client computer and the server computer. The modified information packet can be sent through a firewall before being sent over the computer network to the server computer and by working on top of existing protocols (i.e., UDP, ICMP and TCP), the present invention more easily penetrates the firewall. The information packet is received at a kernel layer of an operating system on the server side. It is then determined at the kernel layer of the operating system on the host computer whether the information packet contains the data that is used for forming the virtual private connection. The server side replies by sending an information packet to the client computer that has been modified at the kernel layer to containing virtual private connection information in the payload portion of the reply information packet. Preferably, the information packet from the client computer and the reply information packet from the server side are each a UDP protocol information packet. Alternative, both information packets could be a TCP/IP protocol information packet, or an ICMP protocol information packet.

BRIEF DESCRIPTION OF THE DRAWINGS

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

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

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

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

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

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

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

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

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

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

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

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.

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

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

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

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

FIG. 15 shows further details of the checkpoint scheme of FIG. 14.

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

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

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

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

FIG. 20 shows a system employing distributed transmission paths.

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

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

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

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

FIG. 24 shows an example using the system of FIG. 23.

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

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

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

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

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

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

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

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

FIG. 33 shows a system block diagram of a computer network in which the "one-click" secure communication link of the present invention is suitable for use.

FIG. 34 shows a flow diagram for installing and establishing a "one-click" secure communication link over a computer network according to the present invention.

FIG. 35 shows a flow diagram for registering a secure domain name according to the present invention.

FIG. 36 shows a system block diagram of a computer network in which a private connection according to the present invention can be configured to more easily traverse a firewall between two computer networks.

FIG. 37 shows a flow diagram for establishing a virtual private connection that is encapsulated using an existing network protocol.

DETAILED DESCRIPTION OF THE INVENTION

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.

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.

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.

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.

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.

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.

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.

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*.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

19

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.

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.

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.

21

- 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.

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.

22

- 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

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.

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.

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

23

nodes, irrespective of the session or end points being transferred between the directly communicating pair of nodes.

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.

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.

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.

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.

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

25

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.

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.

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.

26

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.

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.

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.

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.

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

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.

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

28

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.

<div align="center">A. Hardware Address Hopping</div>

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.

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.

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

<div align="center">29</div>

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.

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.

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.

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.

30

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.

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.

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.

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.

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.

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.

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.

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

33

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.

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).

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.

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.)

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

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.

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

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.

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.

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

36

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.

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.

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.

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,

37

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.

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.

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.

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.

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.

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.

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.

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.

<u>D. Other Synchronization Schemes</u>

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.

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.

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

40

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).

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.

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.

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).

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 will actually help to re-establish synchronization.

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

41

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

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.

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 ... 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:

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

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).

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 jth 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).

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.

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

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)

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:

$((91 (X_i30+4)-4)/30)\bmod 15$ (7).

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

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.

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.

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

44

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 xth 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."

For example, suppose one wanted to represent the number 135 using a presence vector. The 135th 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 135th 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.

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).

A presence vector will have a 1 in the yth 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.

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.

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. Further Synchronization Enhancements

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 ($1 \leq OoO \leq$ 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.

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.

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.

<div align="center">J.   Distributed Transmission Path Variant</div>

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.

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

<div align="center">47</div>

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

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

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.

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).

48

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.

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.

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.

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.

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.

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.

49

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.

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.

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).

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.

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.

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).

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.

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.

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.

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.

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.

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.

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.

52

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:

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

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

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

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).

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

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

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.

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.

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.

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).

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.

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

55

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.

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.

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.

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.

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.

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.

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.

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.

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

57

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.

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.

58

C. Large Link to Small Link Bandwidth Management

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.

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.

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.

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).

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

59

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. 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).

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.

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

60

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.

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.

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

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

61

control the rate at which its hopping window moves by delaying "SYNC ACK" responses to "SYNC_REQ" messages.

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.

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.

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.

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.

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.

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.

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).

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.

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 thetransmitter 3101. Transmitter 3101 then processes the SYNC_REQ in the normal manner.

E. Signaling Synchronizer

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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.

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

67

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.

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.

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.

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.

F. One-Click Secure On-line Communications and Secure Domain Name Service

The present invention provides a technique for establishing a secure communication link between a first computer and a second computer over a computer network. Preferably, a user

enables a secure communication link using a single click of a mouse, or a corresponding minimal input from another input device, such as a keystroke entered on a keyboard or a click entered through a trackball. Alternatively, the secure link is automatically established as a default setting at boot-up of the computer (i.e., no click). FIG. 33 shows a system block diagram 3300 of a computer network in which the one-click secure communication method of the present invention is suitable. In FIG. 33, a computer terminal or client computer 3301, such as a personal computer (PC), is connected to a computer network 3302, such as the Internet, through an ISP 3303. Alternatively, computer 3301 can be connected to computer network 3302 through an edge router. Computer 3301 includes an input device, such as a keyboard and/or mouse, and a display device, such as a monitor. Computer 3301 can communicate conventionally with another computer 3304 connected to computer network 3302 over a communication link 3305 using a browser 3306 that is installed and operates on computer 3301 in a well-known manner.

Computer 3304 can be, for example, a server computer that is used for conducting e-commerce. In the situation when computer network 3302 is the Internet, computer 3304 typically will have a standard top-level domain name such as .com, .net, .org, .edu, .mil or .gov.

FIG. 34 shows a flow diagram 3400 for installing and establishing a "one-click" secure communication link over a computer network according to the present invention. At step 3401, computer 3301 is connected to server computer 3304 over a non-VPN communication link 3305. Web browser 3306 displays a web page associated with server 3304 in a well-known manner. According to one variation of the invention, the display of computer 3301 contains a hyperlink, or an icon representing a hyperlink, for selecting a virtual private network (VPN) communication link ("go secure" hyperlink) through computer network 3302 between terminal 3301 and server 3304. Preferably, the "go secure" hyperlink is displayed as part of the web page downloaded from server computer 3304, thereby indicating that the entity providing server 3304 also provides VPN capability.

By displaying the "go secure" hyperlink, a user at computer 3301 is informed that the current communication link between computer 3301 and server computer 3304 is a non-secure, non-VPN communication link. At step 3402, it is determined whether a user of computer 3301 has selected the "go secure" hyperlink. If not, processing resumes using a non-secure (conventional) communication method (not shown). If, at step 3402, it is determined that the user has selected the "go secure" hyperlink, flow continues to step 3403 where an object

associated with the hyperlink determines whether a VPN communication software module has already been installed on computer 3301. Alternatively, a user can enter a command into computer 3301 to "go secure."

If, at step 3403, the object determines that the software module has been installed, flow continues to step 3407. If, at step 3403, the object determines that the software module has not been installed, flow continues to step 3404 where a non-VPN communication link 3307 is launched between computer 3301 and a website 3308 over computer network 3302 in a well-known manner. Website 3308 is accessible by all computer terminals connected to computer network 3302 through a non-VPN communication link. Once connected to website 3308, a software module for establishing a secure communication link over computer network 3302 can be downloaded and installed. Flow continues to step 3405 where, after computer 3301 connects to website 3308, the software module for establishing a communication link is downloaded and installed in a well-known manner on computer terminal 3301 as software module 3309. At step 3405, a user can optionally select parameters for the software module, such as enabling a secure communication link mode of communication for all communication links over computer network 3302. At step 3406, the -communication link between computer 3301 and website 3308 is then terminated in a well-known manner.

By clicking on the "go secure" hyperlink, a user at computer 3301 has enabled a secure communication mode of communication between computer 3301 and server computer 3304. According to one variation of the invention, the user is not required to do anything more than merely click the "go secure" hyperlink. The user does not need to enter any user identification information, passwords or encryption keys for establishing a secure communication link. All procedures required for establishing a secure communication link between computer 3301 and server computer 3304 are performed transparently to a user at computer 3301.

At step 3407, a secure VPN communications mode of operation has been enabled and software module 3309 begins to establish a VPN communication link. In one embodiment, software module 3309 automatically replaces the top-level domain name for server 3304 within browser 3406 with a secure top-level domain name for server computer 3304. For example, if the top-level domain name for server 3304 is .com, software module 3309 replaces the .com top-level domain name with a .scom top-level domain name, where the "s" stands for secure. Alternatively, software module 3409 can replace the top-level domain name of server 3304 with

any other non-standard top-level domain name.

Because the secure top-level domain name is a non-standard domain name, a query to a standard domain name service (DNS) will return a message indicating that the universal resource locator (URL) is unknown. According to the invention, software module 3409 contains the URL for querying a secure domain name service (SDNS) for obtaining the URL for a secure top-level domain name. In this regard, software module 3309 accesses a secure portal 3310 that interfaces a secure network 3311 to computer network 3302. Secure network 3311 includes an internal router 3312, a secure domain name service (SDNS) 3313, a VPN gatekeeper 3314 and a secure proxy 3315. The secure network can include other network services, such as e-mail 3316, a plurality of chatrooms (of which only one chatroom 3317 is shown), and a standard domain name service (STD DNS) 3318. Of course, secure network 3311 can include other resources and services that are not shown in FIG. 33.

When software module 3309 replaces the standard top-level domain name for server 3304 with the secure top-level domain name, software module 3309 sends a query to SDNS 3313 at step 3408 through secure portal 3310 preferably using an administrative VPN communication link 3319. In this configuration, secure portal 3310 can only be accessed using a VPN communication link. Preferably, such a VPN communication link can be based on a technique of inserting a source and destination IP address pair into each data packet that is selected according to a pseudo-random sequence; an IP address hopping regime that pseudorandomly changes IP addresses in packets transmitted between a client computer and a secure target computer; periodically changing at least one field in a series of data packets according to a known sequence; an Internet Protocol (IP) address in a header of each data packet that is compared to a table of valid IP addresses maintained in a table in the second computer; and/or a comparison of the IP address in the header of each data packet to a moving window of valid IP addresses, and rejecting data packets having IP addresses that do not fall within the moving window. Other types of VPNs can alternatively be used. Secure portal 3310 authenticates the query from software module 3309 based on the particular information hopping technique used for VPN communication link 3319.

SDNS 3313 contains a cross-reference database of secure domain names and corresponding secure network addresses. That is, for each secure domain name, SDNS 3313 stores a computer network address corresponding to the secure domain name. An entity can

71

register a secure domain name in SDNS 3313 so that a user who desires a secure communication link to the website of the entity can automatically obtain the secure computer network address for the secure website. Moreover, an entity can register several secure domain names, with each respective secure domain name representing a different priority level of access in a hierarchy of access levels to a secure website. For example, a securities trading website can provide users secure access so that a denial of service attack on the website will be ineffectual with respect to users subscribing to the secure website service. Different levels of subscription can be arranged based on, for example, an escalating fee, so that a user can select a desired level of guarantee for connecting to the secure securities trading website. When a user queries SDNS 3313 for the secure computer network address for the securities trading website, SDNS 3313 determines the particular secure computer network address based on the user's identity and the user's subscription level.

At step 3409, SDNS 3313 accesses VPN gatekeeper 3314 for establishing a VPN communication link between software module 3309 and secure server 3320. Server 3320 can only be accessed through a VPN communication link. VPN gatekeeper 3314 provisions computer 3301 and secure web server computer 3320, or a secure edge router for server computer 3320, thereby creating the VPN. Secure server computer 3320 can be a separate server computer from server computer 3304, or can be the same server computer having both non-VPN and VPN communication link capability, such as shown by server computer 3322. Returning to FIG. 34, in step 3410, SDNS 3313 returns a secure URL to software module 3309 for the .scom server address for a secure server 3320 corresponding to server 3304.

Alternatively, SDNS 3313 can be accessed through secure portal 3310 "in the clear", that is, without using an administrative VPN communication link. In this situation, secure portal 3310 preferably authenticates the query using any well-known technique, such as a cryptographic technique, before allowing the query to proceed to SDNS 3319. Because the initial communication link in this situation is not a VPN communication link, the reply to the query can be "in the clear." The querying computer can use the clear reply for establishing a VPN link to the desired domain name. Alternatively, the query to SDNS 3313 can be in the clear, and SDNS 3313 and gatekeeper 3314 can operate to establish a VPN communication link to the querying computer for sending the reply.

At step 3411, software module 3309 accesses secure server 3320 through VPN

communication link 3321 based on the VPN resources allocated by VPN gatekeeper 3314. At step 3412, web browser 3306 displays a secure icon indicating that the current communication link to server 3320 is a secure VPN communication link. Further communication between computers 3301 and 3320 occurs via the VPN, e.g., using a "hopping" regime as discussed above. When VPN link 3321 is terminated at step 3413, flow continues to step 3414 where software module 3309 automatically replaces the secure top-level domain name with the corresponding non-secure top-level domain name for server 3304. Browser 3306 accesses a standard DNS 3325 for obtaining the non-secure URL for server 3304. Browser 3306 then connects to server 3304 in a well-known manner. At step 3415, browser 3306 displays the "go secure" hyperlink or icon for selecting a VPN communication link between terminal 3301 and server 3304. By again displaying the "go secure" hyperlink, a user is informed that the current communication link is a non-secure, non-VPN communication link.

When software module 3309 is being installed or when the user is off-line, the user can optionally specify that all communication links established over computer network 3302 are secure communication links. Thus, anytime that a communication link is established, the link is a VPN link. Consequently, software module 3309 transparently accesses SDNS 3313 for obtaining the URL for a selected secure website. In other words, in one embodiment, the user need not "click" on the secure option each time secure communication is to be effected.

Additionally, a user at computer 3301 can optionally select a secure communication link through proxy computer 3315. Accordingly, computer 3301 can establish a VPN communication link 3323 with secure server computer 3320 through proxy computer 3315. Alternatively, computer 3301 can establish a non-VPN communication link 3324 to a non-secure website, such as non-secure server computer 3304.

FIG. 35 shows a flow diagram 3500 for registering a secure domain name according to the present invention. At step 3501, a requester accesses website 3308 and logs into a secure domain name registry service that is available through website 3308. At step 3502, the requestor completes an online registration form for registering a secure domain name having a top-level domain name, such as .com, .net, .org, .edu, .mil or .gov. Of course, other secure top-level domain names can also be used. Preferably, the requestor must have previously registered a non-secure domain name corresponding to the equivalent secure domain name that is being

requested. For example, a requestor attempting to register secure domain name "website.scom" must have previously registered the corresponding non-secure domain name "website.com".

At step 3503, the secure domain name registry service at website 3308 queries a non-secure domain name server database, such as standard DNS 3322, using, for example, a whois query, for determining ownership information relating to the non-secure domain name corresponding to the requested secure domain name. At step 3504, the secure domain name registry service at website 3308 receives a reply from standard DNS 3322 and at step 3505 determines whether there is conflicting ownership information for the corresponding non-secure domain name. If there is no conflicting ownership information, flow continues to step 3507, otherwise flow continues to step 3506 where the requestor is informed of the conflicting ownership information. Flow returns to step 3502.

When there is no conflicting ownership information at step 3505, the secure domain name registry service (website 3308) informs the requestor that there is no conflicting ownership information and prompts the requestor to verify the information entered into the online form and select an approved form of payment. After confirmation of the entered information and appropriate payment information, flow continues to step 3508 where the newly registered secure domain name sent to SDNS 3313 over communication link 3326.

If, at step 3505, the requested secure domain name does not have a corresponding equivalent non-secure domain name, the present invention informs the requestor of the situation and prompts the requestor for acquiring the corresponding equivalent non-secure domain name for an increased fee. By accepting the offer, the present invention automatically registers the corresponding equivalent non-secure domain name with standard DNS 3325 in a well-known manner. Flow then continues to step 3508.

G. Tunneling Secure Address Hopping Protocol Through Existing Protocol Using Web Proxy

The present invention also provides a technique for implementing the field hopping schemes described above in an application program on the client side of a firewall between two computer networks, and in the network stack on the server side of the firewall. The present invention uses a new secure connectionless protocol that provides good denial of service rejection capabilities by layering the new protocol on top of an existing IP protocol, such as the ICMP, UDP or TCP protocols. Thus, this aspect of the present invention does not require

74

changes in the Internet infrastructure.

According to the invention, communications are protected by a client-side proxy application program that accepts unencrypted, unprotected communication packets from a local browser application. The client-side proxy application program tunnels the unencrypted, unprotected communication packets through a new protocol, thereby protecting the communications from a denial of service at the server side. Of course, the unencrypted, unprotected communication packets can be encrypted prior to tunneling.

The client-side proxy application program is not an operating system extension and does not involve any modifications to the operating system network stack and drivers. Consequently, the client is easier to install, remove and support in comparison to a VPN. Moreover, the client-side proxy application can be allowed through a corporate firewall using a much smaller "hole" in the firewall and is less of a security risk in comparison to allowing a protocol layer VPN through a corporate firewall.

The server-side implementation of the present invention authenticates valid field-hopped packets as valid or invalid very early in the server packet processing, similar to a standard virtual private network, for greatly minimizing the impact of a denial of service attempt in comparison to normal TCP/IP and HTTP communications, thereby protecting the server from invalid communications.

FIG. 36 shows a system block diagram of a computer network 3600 in which a virtual private connection according to the present invention can be configured to more easily traverse a firewall between two computer networks. FIG. 37 shows a flow diagram 3700 for establishing a virtual private connection that is encapsulated using an existing network protocol.

In FIG. 36 a local area network (LAN) 3601 is connected to another computer network 3602, such as the Internet, through a firewall arrangement 3603. Firewall arrangement operates in a well-known manner to interface LAN 3601 to computer network 3602 and to protect LAN 3601 from attacks initiated outside of LAN 3601.

A client computer 3604 is connected to LAN 3601 in a well-known manner. Client computer 3604 includes an operating system 3605 and a web browser 3606. Operating system 3605 provides kernel mode functions for operating client computer 3604. Browser 3606 is an application program for accessing computer network resources connected to LAN 3601 and computer network 3602 in a well-known manner. According to the present invention, a proxy

75

application 3607 is also stored on client computer 3604 and operates at an application layer in conjunction with browser 3606. Proxy application 3607 operates at the application layer within client computer 3604 and when enabled, modifies unprotected, unencrypted message packets generated by browser 3606 by inserting data into the message packets that are used for forming a virtual private connection between client computer 3604 and a server computer connected to LAN 3601 or computer network 3602. According to the invention, a virtual private connection does not provide the same level of security to the client computer as a virtual private network. A virtual private connection can be conveniently authenticated so that, for example, a denial of service attack can be rapidly rejected, thereby providing different levels of service that can be subscribed to by a user.

Proxy application 3607 is conveniently installed and uninstalled by a user because proxy application 3607 operates at the application layer within client computer 3604. On installation, proxy application 3607 preferably configures browser 3606 to use proxy application for all web communications. That is, the payload portion of all message packets is modified with the data for forming a virtual private connection between client computer 3604 and a server computer. Preferably, the data for forming the virtual private connection contains field-hopping data, such as described above in connection with VPNs. Also, the modified message packets preferably conform to the UDP protocol. Alternatively, the modified message packets can conform to the TCP/IP protocol or the ICMP protocol. Alternatively, proxy application 3606 can be selected and enabled through, for example, an option provided by browser 3606. Additionally, proxy application 3607 can be enabled so that only the payload portion of specially designated message packets is modified with the data for forming a virtual private connection between client computer 3604 and a designated host computer. Specially designated message packets can be, for example, selected predetermined domain names.

Referring to FIG. 37, at step 3701, unprotected and unencrypted message packets are generated by browser 3606. At step 3702, proxy application 3607 modifies the payload portion of all message packets by tunneling the data for forming a virtual private connection between client computer 3604 and a destination server computer into the payload portion. At step, 3703, the modified message packets are sent from client computer 3604 to, for example, website (server computer) 3608 over computer network 3602.

Website 3608 includes a VPN guard portion 3609, a server proxy portion 3610 and a web server portion 3611. VPN guard portion 3609 is embedded within the kernel layer of the operating system of website 3608 so that large bandwidth attacks on website 3608 are rapidly rejected. When client computer 3604 initiates an authenticated connection to website 3608, VPN guard portion 3609 is keyed with the hopping sequence contained in the message packets from client computer 3604, thereby performing a strong authentication of the client packet streams entering website 3608 at step 3704. VPN guard portion 3609 can be configured for providing different levels of authentication and, hence, quality of service, depending upon a subscribed level of service. That is, VPN guard portion 3609 can be configured to let all message packets through until a denial of service attack is detected, in which case VPN guard portion 3609 would allow only client packet streams conforming to a keyed hopping sequence, such as that of the present invention.

Server proxy portion 3610 also operates at the kernel layer within website 3608 and catches incoming message packets from client computer 3604 at the VPN level. At step 3705, server proxy portion 3610 authenticates the message packets at the kernel level within host computer 3604 using the destination IP address, UDP ports and discriminator fields. The authenticated message packets are then forwarded to the authenticated message packets to web server portion 3611 as normal TCP web transactions.

At step 3705, web server portion 3611 responds to message packets received from client computer 3604 in accordance with the particular nature of the message packets by generating reply message packets. For example, when a client computer requests a webpage, web server portion 3611 generates message packets corresponding to the requested webpage. At step 3706, the reply message packets pass through server proxy portion 3610, which inserts data into the payload portion of the message packets that are used for forming the virtual private connection between host computer 3608 and client computer 3604 over computer network 3602. Preferably, the data for forming the virtual private connection is contains field-hopping data, such as described above in connection with VPNs. Server proxy portion 3610 operates at the kernel layer within host computer 3608 to insert the virtual private connection data into the payload portion of the reply message packets. Preferably, the modified message packets sent by host computer 3608 to client computer 3604 conform to the UDP protocol. Alternatively, the modified message packets can conform to the TCP/IP protocol or the ICMP protocol.

At step 3707, the modified packets are sent from host computer 3608 over computer network 3602 and pass through firewall 3603. Once through firewall 3603, the modified packets are directed to client computer 3604 over LAN 3601 and are received at step 3708 by proxy application 3607 at the application layer within client computer 3604. Proxy application 3607 operates to rapidly evaluate the modified message packets for determining whether the received packets should be accepted or dropped. If the virtual private connection data inserted into the received information packets conforms to expected virtual private connection data, then the received packets are accepted. Otherwise, the received packets are dropped.

While the present invention has been described in connection with the illustrated embodiments, it will be appreciated and understood that modifications may be made without departing from the true spirit and scope of the invention.

CLAIMS

What is claimed is:

1. A system for providing a secure domain name service over a computer network, comprising:

a portal connected to a computer network, the portal authenticating a query for a secure computer network address; and

a domain name database connected to the computer network through the portal, the domain name database storing secure computer network addresses for the computer network.

2. The system of claim 1, wherein each secure computer network address is based on a non-standard top-level domain name.

3. The system of claim 2, wherein the non-standard top-level domain name is one of .scom, .sorg, .snet, .sgov, .sedu, .smil and .sint.

4. The system of claim 1, wherein the computer network includes the Internet.

5. The system of claim 1, wherein the secure portal comprises an edge router.

6. The system of claim 1, wherein the portal authenticates the query using a cryptographic technique.

7. The system of claim 1, wherein the portal is connectable to a virtual private network link through the computer network.

8. The system of claim 7, wherein the secure communication link is one of a plurality of secure communication links in a hierarchy of secure communication links.

9. The system of claim 7, wherein the virtual private network is based on inserting into each data packet one or more data values that vary according to a pseudo-random sequence.

10. The system of claim 7, wherein the virtual private network is based on a computer network address hopping regime that is used to pseudorandomly change computer network addresses in packets transmitted between a first computer and a second computer.

11. The system of claim 7, wherein the virtual private network is based on comparing a value in each data packet transmitted between a first computer and a second computer to a moving window of valid values.

12. The system of claim 7, wherein the virtual private network is based on a comparison of a discriminator field in a header of each data packet to a table of valid discriminator fields maintained for a first computer.

13. A method for registering a secure domain name, comprising steps of:

receiving a request for registering a secure domain name;

verifying ownership information for an equivalent non-secure domain name corresponding to the secure domain name;

registering the secure domain name in a secure domain name service when the ownership information for the equivalent non-secure domain name is consistent with ownership information for the secure domain name.

14. The method according to claim 13, wherein the step of verifying ownership information includes steps of:

determining whether the equivalent non-secure domain name corresponding to the secure domain name has been registered in a non-secure domain name service; and

querying whether the equivalent non-secure domain name should be registered in the non-secure domain name service when the equivalent non-secure domain name has not been registered in the non-secure domain name service.

15. A computer-readable storage medium, comprising:

a storage area; and

computer-readable instructions for a method for registering a secure domain name, the method comprising steps of:

receiving a request for registering a secure domain name;

verifying ownership information for an equivalent non-secure domain name corresponding to the secure domain name;

registering the secure domain name in a secure domain name service when the ownership information for the equivalent non-secure domain name is consistent with ownership information for the secure domain name.

16. The computer-readable medium according to claim 15, wherein the step of verifying ownership information includes steps of:

determining whether the equivalent non-secure domain name corresponding to the secure domain name has been registered in a non-secure domain name service; and

querying whether the equivalent non-secure domain name should be registered in the non-secure domain name service when the equivalent non-secure domain name has not been registered in the non-secure domain name service.

17. A method for registering a domain name, comprising steps of:

(i)   receiving a request for registering a first domain name;

(ii)   verifying ownership information for a second domain name corresponding to the first domain name; and

(iii)   registering the first domain name when the ownership information for the second domain name is consistent with ownership information for the first domain name.

18. The method of claim 17, wherein the first domain name comprises a non-standard top-level domain and the second domain name comprises a standard top-level domain.

19. The method of claim 17, further comprising the step of storing information corresponding to the registration performed in step (iii) in a database separate from a database storing information for standard domain name registrations.

20. The method according to claim 17, wherein the step of verifying ownership information includes steps of:

(a)   determining whether the second domain name has been registered in a domain name service; and

(b)   querying whether the second domain name should be registered in the domain name service when the second domain name has not been registered in the domain name service.

21. A computer-readable medium, comprising computer-readable instructions for a method for registering a domain name, the method comprising steps of:

(i)   receiving a request for registering a first domain name;

(ii)   verifying ownership information for a second domain name corresponding to the first domain name; and

(iii)   registering the first domain name when the ownership information for the second domain name is consistent with ownership information for the first domain name.

22. The computer readable medium of claim 21, wherein the first domain name comprises a non-standard top-level domain and the second domain name comprises a standard top-level domain.

23. The computer-readable medium of claim 21, wherein the step of verifying ownership information includes steps of:

(a)   determining whether the second domain name has been registered in a domain name service; and
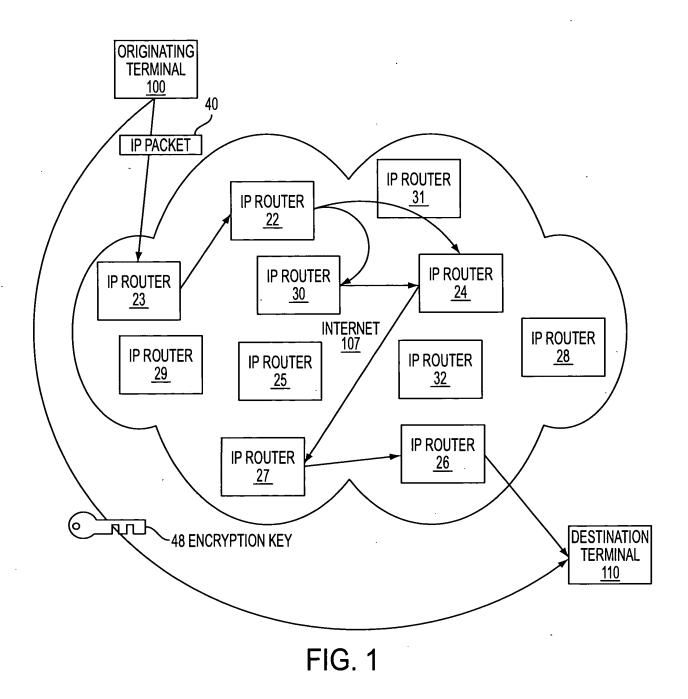
(b)   querying whether the second domain name should be registered in the domain name service when the second domain name has not been registered in the domain name service.

ABSTRACT
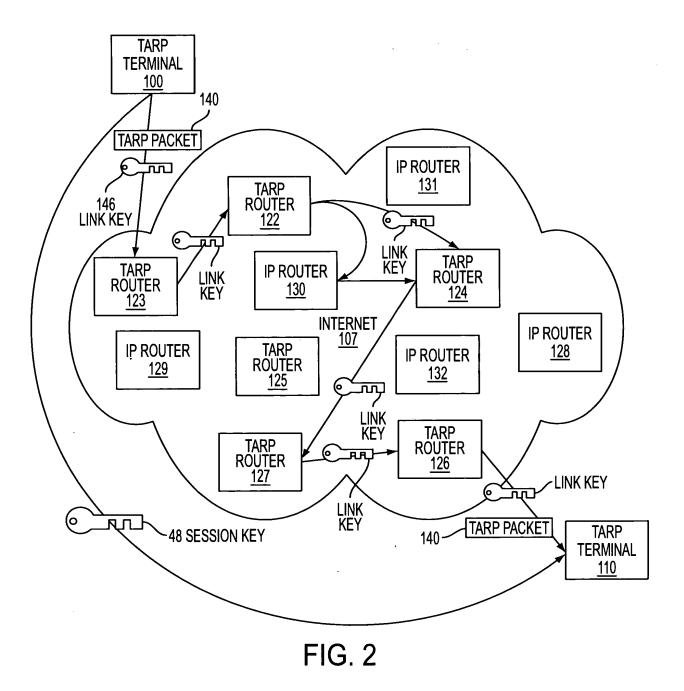
A secure domain name service for a computer network is disclosed that includes a portal connected to a computer network, such as the Internet, and a domain name database connected to the computer network through the portal. The portal authenticates a query for a secure computer network address, and the domain name database stores secure computer network addresses for the computer network. Each secure computer network address is based on a non-standard top-level domain name, such as .scom, .sorg, .snet, .snet, .sedu, .smil and .sint.

ORIGINATING
TERMINAL
100

40

IP PACKET

IP ROUTER
22

IP ROUTER
31

IP ROUTER
23

IP ROUTER
30

IP ROUTER
24

INTERNET
107

IP ROUTER
29

IP ROUTER
25

IP ROUTER
32

IP ROUTER
28

IP ROUTER
27

IP ROUTER
26

48 ENCRYPTION KEY

DESTINATION
TERMINAL
110

FIG. 1

FIG. 2

FIG. 3A

FIG. 3B

TARP TRANSCEIVER
405

NETWORK (IP) LAYER
410

IP   415

TARP LAYER
420

IP_C   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)

IP_C   A

450
DATA LINK
PROTOCOL WRAPPER

FIG. 4

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

```
┌─────────────────────────┐
│ BACKGROUND LOOP - DECOY │
│       GENERATION        │──S40
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│ AUTHENTICATE TARP PACKET│
│        RECEIVED         │──S42
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│    DECRYPT OUTER LAYER   │
│ ENCRYPTION WITH LINK KEY │──S43
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│  INCREMENT PERISHABLE   │
│    COUNTER IF DECOY     │──S44
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│ THROW AWAY DECOY OR KEEP│
│  IN RESPONSE TO ALGORITHM│──S45
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│ CACHE TARP PACKETS UNTIL│
│   WINDOW IS ASSEMBLED   │──S46
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│   DEINTERLEAVE PACKETS  │
│     FORMING WINDOW      │──S47
└─────────────────────────┘
            │
            ▼
┌─────────────────────────┐
│      DECRYPT BLOCK      │──S48
└─────────────────────────┘
```

DIVIDE BLOCK INTO PACKETS USING WINDOW SEQUENCE DATA, ADD CLEAR IP HEADERS GENERATED FROM TARP HEADERS ──S49

HAND COMPLETED IP PACKETS TO IP LAYER PROCESS ──S50

FIG. 7

CLIENT
TERMINAL
801

SSYN
PACKET
821

SSYN ACK
PACKET
822

SSYN ACK
ACK PACKET
823

TARP
ROUTER
811

825
SECURE SESSION
INITIATION ACK

824
SECURE SESSION
INITIATION

FIG. 8

CLIENT 1
901

TARP
ROUTER
911

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

FIG. 10

FIG. 11

1160

ETHERNET FRAME HEADER	1104
SRC. HW ADDRESS: 53	1104A
DEST. HW ADDRESS: 88	1104B
IP PACKET HEADER	1105
SOURCE IP ADDRESS: 71	1105A
DEST. IP ADDRESS: 91	1105B
DISCRIM FIELD: 45	1105C
PAYLOAD #3	1113

IP3

1150

ETHERNET FRAME HEADER	1101
SRC. HW ADDRESS: 53	1101A
DEST. HW ADDRESS: 88	1101B
IP PACKET HEADER	1102
SOURCE IP ADDRESS: 10	1102A
DEST. IP ADDRESS: 14	1102B
DISCRIM FIELD: 77	1102C
PAYLOAD #1	1110

IP1

IP PACKET HEADER	1103
SOURCE IP ADDRESS: 13	1103A
DEST. IP ADDRESS: 15	1103B
DISCRIM FIELD: 13	1103C
PAYLOAD #2	1112

IP2

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

FIG. 13

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

FIG. 16

FIG. 17

FIG. 18

FIG. 19

FIG. 20

AD TABLE		2101
IP1	IP2	
IP3	IP4	
⋮	⋮	

AE TABLE — 2102

AF TABLE — 2103

BD TABLE — 2104

BE TABLE

LINK DOWN ⟶ 2105

BF TABLE — 2106

2100

CD TABLE — 2107

CE TABLE — 2108

CF TABLE — 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 / NO

PATH X WEIGHT > MIN.? — 2207
YES / NO

SET WEIGHT TO MIN. VALUE — 2209

DECREASE WEIGHT FOR PATH X — 2208

PATH X WEIGHT LESS THAN STEADY STATE VALUE? — 2204
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

FIG. 22B

FIG. 23

FIG. 24

FIG. 25
(PRIOR ART)

FIG. 26

FIG. 27

FIG. 28

FIG. 29

FIG. 30

FIG. 31

CLIENT   SERVER

SEND DATA PACKET
USING ckpt_n   DATA
CKPT_O=ckpt_n   PASS DATA UP STACK
GENERATE NEW ckpt_n   ckpt_o=ckpt_n
START TIMER, SHUT TRANSMITTER   GENERATE NEW ckpt_n
OFF   SYNC_ACK   GENERATE NEW ckpt_r FOR
                                                                TRANSMITTER SIDE
IF CKPT_O IN SYNC_ACK   TRANSMIT SYNC_ACK
MATCHES TRANSMITTER'S   CONTAINING ckpt_o
ckpt_o
UPDATE RECEIVER'S
ckpt_r
KILL TIMER, TURN
TRANSMITTER ON

SEND DATA PACKET
USING ckpt_n   DATA   X
ckpt_o=ckpt_n
GENERATE NEW ckpt_n
START TIMER, SHUT TRANSMITTER
OFF

 WHEN TIMER EXPIRES   SYNC_REQ
 TRANSMIT SYNC_REQ   ckpt_o=ckpt_n
 USING TRANSMITTERS   GENERATE NEW ckpt_n
 ckpt_o, START TIMER   GENERATE NEW ckpt_r FOR
                                                                TRANSMITTER SIDE
                                                TRANSMIT SYNC_ACK
IF ckpt_o IN SYNC_ACK   SYNC_ACK   CONTAINING ckpt_o
MATCHES TRANSMITTER'S
ckpt_o
UPDATE RECEIVER'S
ckpt_r
KILL TIMER, TURN
TRANSMITTER ON

FIG. 32

FIG. 33

FIG. 34

3500

REQUESTOR ACCESSES WEBSITE
AND LOGS INTO SECURE
DOMAIN NAME REGISTRY SERVICE — 3501

REQUESTER COMPLETES ONLINE
REGISTRATION FORM — 3502

QUERY STANDARD DOMAIN NAME
SERVICE REGARDING OWNERSHIP
OF EQUIVALENT NON-SECURE
DOMAIN NAME — 3503

RECEIVE REPLY FROM STANDARD
DOMAIN NAME REGISTRY — 3504

CONFLICT
? — 3505

YES → INFORM REQUESTOR
OF CONFLICT — 3506

NO

VERIFY INFORMATION AND
ENTER PAYMENT INFORMATION — 3507

REGISTER SECURE DOMAIN NAME — 3508

FIG. 35

FIG. 36

```
3700
```

| GENERATE MESSAGE PACKETS | 3701 |

MODIFY MESSAGE PACKETS WITH PRIVATE CONNECTION DATA AT AN APPLICATION LAYER — 3702

SEND TO HOST COMPUTER THROUGH FIREWALL — 3703

RECEIVE PACKETS AND AUTHENTICATE AT KERNEL LAYER OF HOST COMPUTER — 3704

RESPOND TO RECEIVED MESSAGE PACKETS AND GENERATE REPLY MESSAGE PACKETS — 3705

MODIFY REPLY MESSAGE PACKETS WITH PRIVATE CONNECTION DATA AT A KERNEL LAYER — 3706

SEND PACKETS TO CLIENT COMPUTER THROUGH FIREWIRE — 3707

RECEIVE PACKETS AT CLIENT COMPUTER AND AUTHENTICATE AT APPLICATION LAYER — 3708

FIG. 37

Banner & Witcoff Ref. No.   000479.00111
Client Ref. No.   10007-Cont.

JOINT DECLARATION FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residence, post office address and citizenship 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 AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES, the specification of which

☒   is attached hereto.

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

☐   was filed under the Patent Cooperation Treaty (PCT) and accorded International Application No. _____, filed _____, and amended on _____ (if any).

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

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate 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 No.	Date of Filing (day month year)	Date of Issue (day month year)	Priority Claimed Under 35 U.S.C. §119

Prior United States Provisional Application(s)

We hereby claim priority benefits under Title 35, United States Code, §119(e)(1) of any U.S. provisional application listed below:

U.S. Provisional Application No.	Date of Filing (day month year)	Priority Claimed Under 35 U.S.C. §119(e)(1)
60/106,261	30 October 1998	Yes
60/137,704	7 June 1999	Yes

Prior United States Application(s)

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(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

BANNER & WITCOFF, LTD.   Rev 1.1 10-09-2001

Page 1 of 2

Banner & Witcoff Ref. No.   000479.000111
Client Ref. No.   10007-Cont.

Application Serial No.	Date of Filing (Day, Month, Year)	Status — Patented, Pending, Abandoned
09/558,210	26 April 2000	Pending
09/504,783	15 February 2000	Patented
09/429,643	29 October 1999	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 to transact all business in the Patent and Trademark Office connected herewith the practitioners at:

Customer Number: 22907   (WDC)

Please address all correspondence and telephone communications to the address and telephone number for this Customer Number.

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 Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature _Vita Larson_   Date _11/10/2003_
Full Name of First Inventor _Larson_   _Victor_
                          Family Name   First Given Name   Second Given Name
Residence _Fairfax, Virginia_   Citizenship _USA_
Post Office Address _12026 Lisa Marie Court, Fairfax, Virginia 22033_

Signature _____
Full Name of Second Inventor _Short, III_   _Robert_   _Dunham_
                          Family Name   First Given Name   Second Given Name
Residence _Leesburg, Virginia_   Citizenship _USA_
Post Office Address _38710 Goose Creek Lane, Leesburg, Virginia 20175_

Signature _____   Date _____
Full Name of Third Inventor _Munger_   _Edmund_   _Colby_
                          Family Name   First Given Name   Second Given Name
Residence _Crownsville, Maryland_   Citizenship _USA_
Post Office Address _1101 Opaca Court, Crownsville, Maryland 21032_

Signature _____   Date _Nov 10 2003_
Full Name of Fourth Inventor _Williamson_   _Michael_
                          Family Name   First Given Name   Second Given Name
Residence _South Riding, Virginia_   Citizenship _USA_
Post Office Address _26203 Ocala Circle, South Riding, Virginia 20152_

BANNER & WITCOFF, LTD.   Rev 1.1 10-09-2001

Page 2 of 2

JOINT DECLARATION FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residence, post office address and citizenship 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 <u>AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES</u>, the specification of which

☒   is attached hereto.

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

☐   was filed under the Patent Cooperation Treaty (PCT) and accorded International Application No. _____, filed _____, and amended on _____ (if any).

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

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate 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 No.	Date of Filing (day month year)	Date of Issue (day month year)	Priority Claimed Under 35 U.S.C. §119

Prior United States Provisional Application(s)

We hereby claim priority benefits under Title 35, United States Code, §119(e)(1) of any U.S. provisional application listed below:

U.S. Provisional Application No.	Date of Filing (day month year)	Priority Claimed Under 35 U.S.C. §119(e)(1)
60/106,261	30 October 1998	Yes
60/137,704	7 June 1999	Yes

Prior United States Application(s)

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(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Banner & Witcoff Ref. No.   000479.000111
Client Ref. No.   10007-Cont.

Application Serial No.	Date of Filing (Day, Month, Year)	Status — Patented, Pending, Abandoned
09/558,210	26 April 2000	Pending
09/504,783	15 February 2000	Patented
09/429,643	29 October 1999	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 to transact all business in the Patent and Trademark Office connected herewith the practitioners at:

Customer Number: 22907   (WDC)

Please address all correspondence and telephone communications to the address and telephone number for this Customer Number.

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 Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature_____   Date_____
Full Name of First Inventor_____Larson_____Victor_____
                                    Family Name   First Given Name   Second Given Name
Residence__Fairfax, Virginia_____   Citizenship_____USA___
Post Office Address____12026 Lisa Marie Court, Fairfax, Virginia 22033_____

Signature_____   Date 11/7/03_____
Full Name of Second Inventor_____Short, III_____Robert_____Dunham_____
                                    Family Name   First Given Name   Second Given Name
Residence__Leesburg, Virginia_____   Citizenship_USA_____
Post Office Address____38710 Goose Creek Lane, Leesburg, Virginia 20175_____

Signature_____   Date_____
Full Name of Third Inventor_____Munger_____Edmund_____Colby____
                                    Family Name   First Given Name   Second Given Name
Residence__Crownsville, Maryland_____   Citizenship_USA_____
Post Office Address____1101 Opaca Court, Crownsville, Maryland 21032_____

Signature_____   Date_____
Full Name of Fourth Inventor_____Williamson_____Michael_____
                                    Family Name   First Given Name   Second Given Name
Residence__South Riding, Virginia_____   Citizenship_USA_____
Post Office Address____26203 Ocala Circle, South Riding, Virginia 20152_____

BANNER & WITCOFF, LTD.   Rev 1.1 10-09-2001

Page 2 of 2

Banner & Witcoff Ref. No.   000479.00111
Client Ref. No.   10007-Cont.

JOINT DECLARATION FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residence, post office address and citizenship 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 <u>AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES</u>, the specification of which

☒ is attached hereto.

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

☐ was filed under the Patent Cooperation Treaty (PCT) and accorded International Application No. _____, filed _____, and amended on _____ (if any).

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

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate 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 No.	Date of Filing (day month year)	Date of Issue (day month year)	Priority Claimed Under 35 U.S.C. §119

Prior United States Provisional Application(s)

We hereby claim priority benefits under Title 35, United States Code, §119(e)(1) of any U.S. provisional application listed below:

U.S. Provisional Application No.	Date of Filing (day month year)	Priority Claimed Under 35 U.S.C. §119(e)(1)
60/106,261	30 October 1998	Yes
60/137,704	7 June 1999	Yes

Prior United States Application(s)

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(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

BANNER & WITCOFF, LTD.   Rev 1.1 10-09-2001

Banner & Witcoff Ref. No.   000479.000111
Client Ref. No.   10007-Cont.

Application Serial No.	Date of Filing (Day, Month, Year)	Status — Patented, Pending, Abandoned
09/558,210	26 April 2000	Pending
09/504,783	15 February 2000	Patented
09/429,643	29 October 1999	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 to transact all business in the Patent and Trademark Office connected herewith the practitioners at:

Customer Number: 22907   (WDC)

Please address all correspondence and telephone communications to the address and telephone number for this Customer Number.

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 Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature_____ Date_____
Full Name of First Inventor_____Larson_____Victor_____
                                        Family Name   First Given Name   Second Given Name
Residence___Fairfax, Virginia_____Citizenship_____USA_____
Post Office Address____12026 Lisa Marie Court, Fairfax, Virginia 22033_____

Signature_____ Date_____
Full Name of Second Inventor_____Short, III_____Robert_____Dunham_____
                                        Family Name   First Given Name   Second Given Name
Residence___Leesburg, Virginia_____Citizenship__USA_____
Post Office Address____38710 Goose Creek Lane, Leesburg, Virginia 20175_____

Signature___Edmund Colby Munger_____ Date__11 November 2003_____
Full Name of Third Inventor_____Munger_____Edmund_____Colby_____
                                        Family Name   First Given Name   Second Given Name
Residence___Crownsville, Maryland_____Citizenship__USA_____
Post Office Address____1101 Opaca Court, Crownsville, Maryland 21032_____

Signature_____ Date_____
Full Name of Fourth Inventor_____Williamson_____Michael_____
                                        Family Name   First Given Name   Second Given Name
Residence___South Riding, Virginia_____Citizenship__USA_____
Post Office Address____26203 Ocala Circle, South Riding, Virginia 20152_____

BANNER & WITCOFF, LTD.   Rev 1.1 10-09-2001

Page 2 of 2

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::   AN AGILE NETWORK PROTOCOL FOR
                                            SECURE COMMUNCIATIONS USING SECURE
                                            DOMAIN NAMES

Attorney Docket Number::   000479.00111

Request for Early Publication?::   NO

Request for Non-Publication?::   NO

Suggested Drawing Figure::   1

Total Drawing Sheets::   40

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::   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::   Robert
Middle Name::   Durham
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::	Edmund
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::	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 Ocla 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	Continuation of	09/558,210	04/26/00
	Continuation-in-Part of	09/504,783	02/15/00
	Continuation-in-Part of	09/429,643	10/29/99
	Provisional	60/106,261	10/30/98
	Provisional	60/137,704	06/07/99

Foreign Priority Information

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

Assignee Information

Assignee name::   Science Applications International Corporation

                                     10260 Campus Point Drive

4   Initial 11/18/03

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

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	)	
	)	
Victor Larson et al.	)	Group Art Unit: TBA
	)	
Serial No.: TBA	)	Examiner: TBA
(CONT of 09/558,210))		
	)	
Filed: Herewith	)	Atty. Docket No.: 00479.00111
	)	

For: AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

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 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: Ross A. Dannenberg
Registration No. 49,024

1001 G Street, N.W.
Eleventh Floor
Washington, D.C. 20001-4597
(202) 824-3000

Dated: November 18, 2003

PTO-1449 (Modified)	ATTY. DOCKET NO. 000479.00111	SERIAL NUMBER TBD
U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	APPLICANT Victor Larson et al.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
	6,119,171	9/2000	Alkhatib			
	5,588,060	12/24/96	Aziz			
	5,689,566	11/18/9	Nguyen			
	5,842,040	11/24/98	Hughes et al.			
	4,933,846	06/12/90	Humphrey et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
	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.)

	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
	FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security: Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER	DATE CONSIDERED

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. 000479.00111	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
	6,353,614	3/5/02	Borella et al.			
	6,332,158	12/18/01	Risley et al.			
	6,330,562	12/11/01	Boden et al.			
	6,286,047	9/4/01	Ramanathan et al.			
	6,243,749	6/5/01	Sitaraman et al.			
	6,226,751	5/1/01	Arrow et al.			
	6,178,505	1/23/01	Schneider et al.			
	6,119,171	9/12/00	Alkhatib			
	6,079,020	6/20/00	Liu			
	6,052,788	4/18/00	Wesinger, Jr. et al.			
	6,016,318	1/18/00	Tomoike			
	6,006,259	12/21/99	Adelman et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER	DATE CONSIDERED

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. 000479.00111	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
	5,905,859	5/18/99	Holloway et al.			
	5,898,830	4/27/99	Wesinger, Jr. et al.			
	5,892,903	4/6/99	Klaus			
	5,878,231	3/2/99	Baehr et al.			
	5,805,801	9/8/98	Holloway et al.			
	5,796,942	8/18/98	Esbensen			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
	WO 00/70458	11/23/00	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	Linux FreeS/WAN Index File, printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/ on February 21, 2002, 3 Pages
	J. Gilmore, "Swan: Securing the Internet against Wiretapping", printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/rationale.html on February 21, 2002, 4 pages
	Glossary for the Linux FreeS/WAN project, printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/glossary.html on February 21, 2002, 25 pages
	Alan O. Frier et al., "The SSL Protocol Version 3.0", November 18, 1996, printed from http://www.netscape.com/eng/ssl3/draft302.txt on February 4, 2002, 56 pages

EXAMINER	DATE CONSIDERED

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)	ATTY. DOCKET NO. 000479.00111	SERIAL NUMBER TBD
U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	APPLICANT Victor Larson et al.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT	FILING DATE Herewith	GROUP ART UNIT TBD

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	
	0 858 189	8/12/98	EPO				
	WO 01 50688	7/12/01	PCT				
	WO 98 59470	12/30/98	PCT				
	WO 99 48303	9/23/99	PCT				
	WO 99 38081	7/29/99	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	Search Report (dated 8/20/02), International Application No. PCT/US01/04340
	Search Report (dated 8/23/02), International Application No. PCT/US01/13260
	Shree Murthy et al., "Congestion-Oriented Shortest Multipath Routing", Proceedings of IEEE INFOCOM, 1996, pages 1028-1036
	Jim Jones et al., "Distributed Denial of Service Attacks: Defenses", Global Integrity Corporation, 2000, pages 1-14
	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	DATE CONSIDERED

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. 000479.00111	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
	6,016,512	1/20000	Christian Huitema			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
	WO 98 55930	12/10/98	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	Search Report (dated 10/7/02), International Application No. PCT/US01/13261
	F. Halsall, "Data Communications, Computer Networks And Open Systems", Chapter 4, Protocol Basics, 1996, pages 198-203
	Reiter, Michael K. and Rubin, Aviel D. (AT&T Labs – Research), "Crowds: Anonymity for Web Transmissions", pages 1-23
	Dolev, Shlomi and Ostrovsky, Rafil, "Efficient Anonymous Multicast and Reception" (Extended Abstract), 16 pages
	Rubin, Aviel D., Greer, Daniel, and Ranum, Marcus J. (Wiley Computer Publishing), "Web Security Sourcebook", pages 82-94
	FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security" Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER	DATE CONSIDERED

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 FEE DETERMINATION RECORD
Effective October 1, 2003

Application or Docket Number 10/714,849

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	23	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	23 minus 20=	* 3
INDEPENDENT CLAIMS	5 minus 3 =	* 2
MULTIPLE DEPENDENT CLAIM PRESENT		☐

SMALL ENTITY TYPE ☐		OR	OTHER THAN SMALL ENTITY	
RATE	FEE		RATE	FEE
BASIC FEE	385.00	OR	BASIC FEE	770.00
X$ 9=		OR	X$18=	54
X43=		OR	X86=	172
+145=		OR	+290=	
TOTAL		OR	TOTAL	996

* If the difference in column 1 is less than zero, enter "0" in column 2

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	*	Minus	**	=
	Independent	*	Minus	***	=
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	

1 13, 15 17 21

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=	
X43=		OR	X86=	
+145=		OR	+290=	
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=	
X43=		OR	X86=	
+145=		OR	+290=	
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.

PATENT APPLICATION SERIAL NO. _____

U.S. DEPARTMENT OF COMMERCE
PATENT AND TRADEMARK OFFICE
FEE RECORD SHEET

11/20/2003 SDIRETA1 00000016 190733   10714849

01 FC:1001   770.00 DA
02 FC:1202   54.00 DA
03 FC:1201   172.00 DA

PTO-1556
   (5/87)

JMP/Bддy/RAD

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

APPL NO.	FILING OR 371 (c) DATE	ART UNIT	FIL FEE REC'D	ATTY.DOCKET NO	DRAWINGS	TOT CLMS	IND CLMS
10/714,849	11/18/2003	2143	990	000479.00111	40	23	5

OIPE
MAR 19 2004
JG174
PATENT & TRADEMARK OFFICE

22907
BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

CONFIRMATION NO. 3154
FILING RECEIPT

OC000000011897740

Date Mailed: 02/13/2004

Receipt is acknowledged of this regular Patent Application. It will be considered in its order and you will be notified as to the results of the examination. Be sure to provide the U.S. APPLICATION NUMBER, FILING DATE, NAME OF APPLICANT, and TITLE OF INVENTION when inquiring about this application. Fees transmitted by check or draft are subject to collection. Please verify the accuracy of the data presented on this receipt. **If an error is noted on this Filing Receipt, please write to the Office of Initial Patent Examination's Filing Receipt Corrections, facsimile number 703-746-9195. Please provide a copy of this Filing Receipt with the changes noted thereon. If you received a "Notice to File Missing Parts" for this application, please submit any corrections to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections (if appropriate).**

Applicant(s)

Victor Larson, Fairfax, VA;
Robert ~~Durham~~ Short III, Leesburg, VA; DUNHAM
Edmund Colby Munger, Crownsville, MD;
Michael Williamson, South Riding, VA;

Assignment For Published Patent Application

Science Applications International Corporation, San Diego, CA;

Domestic Priority data as claimed by applicant
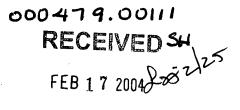
This application is a CON of 09/558,210 04/26/2000
which is a CIP of 09/504,783 02/15/2000 PAT 6,502,135
which is a CIP of 09/429,643 10/29/1999
which claims benefit of 60/106,261 10/30/1998
and claims benefit of 60/137,704 06/07/1999

Foreign Applications

If Required, Foreign Filing License Granted: 02/12/2004

Projected Publication Date: 05/20/2004

Non-Publication Request: No

000479.00111
RECEIVED SH
FEB 17 2004
BANNER & WITCOFF
Rcv'd off

Early Publication Request: No

Title

Agile network protocol for secure communications using secure domain names

Preliminary Class

709

LICENSE FOR FOREIGN FILING UNDER
Title 35, United States Code, Section 184
Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Office of Export Administration, Department of Commerce (15 CFR 370.10 (j)); the Office of Foreign Assets Control, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

re Application of:	)	Filed: November 18, 2003
	)	
Victor LARSON *et al*	)	Group Art Unit: 2143
	)	
Serial No. 10/714,849	)	Docket No. 000479.00111

For: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING
SECURE DOMAIN NAMES

REQUEST FOR CORRECTED FILING RECEIPT

Commissioner for Patents
Alexandria, VA 22313-1450

Dear Sir:

It is requested that a Corrected Filing Receipt be issued to correct errors found in the Official Filing Receipt. Specifically, in the Applicants section, please replace "Durham" with --Dunham--.

A copy of the Official Filing Receipt with the corrections marked in red, along with the original, is attached. Also enclosed is a Supplemental Application Data Sheet. No fee is believed to be associated with this paper since this error occurred through no fault of the Applicants. Nonetheless, should the USPTO determine that a fee is required, please charge such fee to our Deposit Account No. 19-0733.

Respectfully submitted,

Date: 3/18/04

By: _____
Ross A. Dannenberg
Registration No. 49,024

BANNER & WITCOFF, LTD.
1001 G Street, N.W., 11th Floor
Washington, D.C. 20001-4597
Telephone: 202-824-3000

Supplemental Application Data Sheet

Application Information

Application number::	10/714,849
Filing Date::	November 18, 2003
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::	AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNCIATIONS USING SECURE DOMAIN NAMES
Attorney Docket Number::	000479.00111
Request for Early Publication?::	NO
Request for Non-Publication?::	NO
Suggested Drawing Figure::	1
Total Drawing Sheets::	40
Small Entity?::	NO
Latin name::	
Variety denomination name::	
Petition included?::	NO
Petition Type::	
Licensed US Govt. Agency::	

1   Initial 11/18/03
           Revised 3/12/04

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::   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::   Robert

Middle Name::   ~~Durham~~ 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::	Edmund
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::	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 Ocla 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	Continuation of	09/558,210	04/26/00
	Continuation-in-Part of	09/504,783	02/15/00
	Continuation-in-Part of	09/429,643	10/29/99
	Provisional	60/106,261	10/30/98
	Provisional	60/137,704	06/07/99

Foreign Priority Information

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

<div align="center">

4   Initial 11/18/03
Revised 3/12/04

</div>

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

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

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 ($) 298

Complete if Known

Application Number	10/714,849
Filing Date	November 18, 2003
First Named Inventor	Victor LARSON
Examiner Name	TBA
Art Unit	2143
Attorney Docket No.	000479.00111

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	Fee ($)	Small Entity Fee Code	Fee ($)	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 filling fee	

SUBTOTAL (1) ($) 0

2. EXTRA CLAIM FEES FOR UTILITY AND REISSUE

		Extra Claims	Fee from below	Fee Paid
Total Claims	27	-20 ** = 7	X 18	= 126
Independent Claims	7	-5 ** = 2	X 86	= 172
Multiple Dependent	0		X 290	= 0

Large Entity Fee Code	Fee ($)	Small Entity Fee Code	Fee ($)	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) ($) 298

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

FEE CALCULATION (continued)

3. ADDITIONAL FEES

Large Entity Fee Code	Fee ($)	Small Entity Fee Code	Fee ($)	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)	Jordan N. Bodner	Registration No. (Attorney/Agent)	42,338	Telephone	202-824-3000
Signature				Date	May 18, 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.

TRANSMITTAL FORM

(to be used for all correspondence after initial filing)

Application Number	10/714,849
Filing Date	November 18, 2003
First Named Inventor	Victor LARSON
Art Unit	2143
Examiner Name	TBA
Attorney Docket Number	000479.00111

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):*
    Preliminary Amendment

Remarks

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm or Individual name	Jordan N. Bodner, Registration No. 42,338
Signature	
Date	May 18, 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

IFW
2143

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application Of)	
)	Group Art Unit: 2143
Victor Larson et al.)	
)	Examiner: TBD
Serial No.: 10/714,849)	
)	Atty. Dkt. No. 000479.00111
Filed: November 18, 2003)	

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

PRELIMINARY AMENDMENT

Honorable Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Prior to an examination on the merits, please amend the above-identified application as follows.

IN THE SPECIFICATION:

Please amend the first paragraph on p. 1 as follows:

This application claims priority from and is a continuation patent application of co-pending U.S. application serial number 09/558,210, filed April 26, 2000, which is a continuation-in-part patent application of previously-filed U.S. application serial number 09/504,783, filed on 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 patent application of previously-filed U.S. application serial number 09/429,643, filed on October 29, 1999. The subject matter of U.S. application serial number 09/429,643, 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). The present application is also related to U.S. application serial number (Atty Docket No. 479.86838)09/558,209, filed April 26, 2000, and which is incorporated by reference herein.

Please amend the paragraph beginning on p. 71, ln. 2, as follows:

Because the secure top-level domain name is a non-standard domain name, a query to a standard domain name service (DNS) will return a message indicating that the universal resource locator (URL) is unknown. According to the invention, software module 3409 3309 contains the URL for querying a secure domain name service (SDNS) for obtaining the URL for a secure top-level domain name. In this regard, software module 3309 accesses a secure portal 3310 that interfaces a secure network 3311 to computer network 3302. Secure network 3311 includes an internal router 3312, a secure domain name service (SDNS) 3313, a VPN gatekeeper 3314 and a secure proxy 3315. The secure network can include other network services, such as e-mail 3316, a plurality of chatrooms (of which only one chatroom 3317 is shown), and a standard domain name service (STD DNS) 3318. Of course, secure network 3311 can include other resources and services that are not shown in FIG. 33.

IN THE CLAIMS:

Please amend claims 1, 2, 5-7, 13, and 15, and add new claims 24-27, as follows:

1. (Presently Amended) A system for providing a secure domain name service over a computer network, comprising:

a ~~portal~~ server connected to a computer network, the ~~portal~~ server authenticating a query for a secure computer network address having a top-level domain reserved for secure network connections; and

a domain name database connected to the computer network through the ~~portal~~server, the domain name database storing secure computer network addresses for the computer network.

2. (Presently Amended) The system of claim 1, wherein ~~each secure computer network address is based on~~ the top-level domain name is a non-standard top-level domain name.

3. (Original) The system of claim 2, wherein the non-standard top-level domain name is one of .scom, .sorg, .snet, .sgov, .sedu, .smil and .sint.

4. (Original) The system of claim 1, wherein the computer network includes the Internet.

5. (Presently Amended) The system of claim 1, wherein the ~~secure portal~~server comprises an edge router.

6. (Presently Amended) The system of claim 1, wherein the ~~portal~~ server authenticates the query using a cryptographic technique.

7. (Presently Amended) The system of claim 1, wherein the ~~portal~~ server is connectable to a virtual private network link through the computer network.

8. (Original) The system of claim 7, wherein the secure communication link is one of a plurality of secure communication links in a hierarchy of secure communication links.

3

9. (Original) The system of claim 7, wherein the virtual private network is based on inserting into each data packet one or more data values that vary according to a pseudo-random sequence.

10. (Original) The system of claim 7, wherein the virtual private network is based on a computer network address hopping regime that is used to pseudorandomly change computer network addresses in packets transmitted between a first computer and a second computer.

11. (Original) The system of claim 7, wherein the virtual private network is based on comparing a value in each data packet transmitted between a first computer and a second computer to a moving window of valid values.

12. (Original) The system of claim 7, wherein the virtual private network is based on a comparison of a discriminator field in a header of each data packet to a table of valid discriminator fields maintained for a first computer.

13. (Presently Amended) A method for registering a secure domain name, comprising steps of:

    receiving a request for registering a secure domain name;

    verifying ownership information for an equivalent non-secure domain name corresponding to the secure domain name; and

    registering the secure domain name in a secure domain name service when the ownership information for the equivalent non-secure domain name is consistent with ownership information for the secure domain name.

14. (Original) The method according to claim 13, wherein the step of verifying ownership information includes steps of:

4

determining whether the equivalent non-secure domain name corresponding to the secure domain name has been registered in a non-secure domain name service; and

querying whether the equivalent non-secure domain name should be registered in the non-secure domain name service when the equivalent non-secure domain name has not been registered in the non-secure domain name service.

15. (Presently Amended) A computer-readable storage medium, comprising:

a storage area; and

computer-readable instructions for a method for registering a secure domain name, the method comprising steps of:

receiving a request for registering a secure domain name;

verifying ownership information for an equivalent non-secure domain name corresponding to the secure domain name; and

registering the secure domain name in a secure domain name service when the ownership information for the equivalent non-secure domain name is consistent with ownership information for the secure domain name.

16. (Original) The computer-readable medium according to claim 15, wherein the step of verifying ownership information includes steps of:

determining whether the equivalent non-secure domain name corresponding to the secure domain name has been registered in a non-secure domain name service; and

querying whether the equivalent non-secure domain name should be registered in the non-secure domain name service when the equivalent non-secure domain name has not been registered in the non-secure domain name service.

17. (Original) A method for registering a domain name, comprising steps of:

(i) receiving a request for registering a first domain name;

(ii) verifying ownership information for a second domain name corresponding to the first domain name; and

5

(iii) registering the first domain name when the ownership information for the second domain name is consistent with ownership information for the first domain name.

18. (Original) The method of claim 17, wherein the first domain name comprises a non-standard top-level domain and the second domain name comprises a standard top-level domain.

19. (Original) The method of claim 17, further comprising the step of storing information corresponding to the registration performed in step (iii) in a database separate from a database storing information for standard domain name registrations.

20. (Original) The method according to claim 17, wherein the step of verifying ownership information includes steps of:

(a)   determining whether the second domain name has been registered in a domain name service; and

(b)   querying whether the second domain name should be registered in the domain name service when the second domain name has not been registered in the domain name service.

21. (Original) A computer-readable medium, comprising computer-readable instructions for a method for registering a domain name, the method comprising steps of:

(i)   receiving a request for registering a first domain name;

(ii)   verifying ownership information for a second domain name corresponding to the first domain name; and

(iii)   registering the first domain name when the ownership information for the second domain name is consistent with ownership information for the first domain name.

22. (Original) The computer readable medium of claim 21, wherein the first domain name comprises a non-standard top-level domain and the second domain name comprises a standard top-level domain.

23. (Original) The computer-readable medium of claim 21, wherein the step of verifying ownership information includes steps of:

    (a)   determining whether the second domain name has been registered in a domain name service; and

    (b)   querying whether the second domain name should be registered in the domain name service when the second domain name has not been registered in the domain name service.

24.   (New) The method of claim 13, wherein the secure domain name has a top-level domain reserved for secure network connections.

25.   (New) The computer-readable storage medium of claim 15, wherein the secure domain name has a top-level domain reserved for secure network connections.

26.   (New) An apparatus configured to query a computer network address having a top-level domain reserved for secure network connections.

27.   (New) A method comprising querying a computer network address having a top-level domain reserved for secure network connections.

7

REMARKS

Consideration and allowance are respectfully requested. Claims 1-27 are now pending. By this Preliminary Amendment, claims 1, 2, 5-7, 13, and 15 are amended, and new claims 24-27 are added.

The present application is a continuation of U.S. Patent Application Serial No. 09/558,210. In that application, a non-final Office Action was mailed on October 1, 2003 (paper no. 12). The following remarks are made with reference to that Office Action.

The above-referenced Office Action rejected claims 1-12 under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,119,171 to Alkhatib ("Alkhatib"). Also, claims 13-17, 19-21, and 23 were rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,016,512 to Huitema ("Huitema"). Also, claims 18 and 22 were rejected under 35 U.S.C. 103(a) as being unpatentable over Huitema in view of Alkhatib.

Independent claim 1 as amended recites a server connected to a computer network, the server authenticating a query for a secure computer network address having a top-level domain reserved for secure network connections. Applicants do not concede that Alkhatib authenticates a query for a secure computer network address. In any event, Alkhatib fails to teach or suggest a top-level domain reserved for secure network connections, as amended in claim 1.

Claims 2-12, which depend from claim 1, are also allowable for at least those reasons set forth above with regard to claim 1, and further in view of the additional features recited therein.

For example, claim 2 recites that the top-level domain name is a non-standard top-level domain name. Alkhatib refers to top-level domains ".com", ".edu", ".gov", ".mil", ".net", and ".org". Alkhatib, col. 1, lns. 35-43. However, these were standard top-level domains at the time of

Applicants' invention, and thus are not non-standard top-level domains as required by the claim. The Examiner apparently also relies upon Alkhatib referring to the domain name "saturn.ttc.com". Alkhatib, col. 1, ln. 45. However, this also uses the standard top-level domain ".com", and not a non-standard top-level domain as required by the claim. The domain name "saturn" is a third-level domain name, the domain name "ttc" is a second level domain name, and the domain name ".com" is the standard top level domain name. Note that "saturn" and "ttc" are not part of the top-level domain.

Also, claim 3 recites that the non-standard top-level domain name is one of .scom, .sorg, .snet, .sgov, .sedu, .smil and .sint. The Examiner refers to Alkhatib at col. 1, lns. 39-43 to teach this feature. This is a baffling anticipation-based rejection, in that Alkhatib fails to mention even a single one of the top-level domain names .scom, .sorg, .snet, .sgov, .sedu, .smil, or .sint specifically listed in claim 3. Applicants request that the Examiner point out precisely where Alkhatib discloses even a single one of these recited top-level domain names.

Independent claim 13 recites verifying ownership information for an equivalent non-secure domain name corresponding to a secure domain name. Huitema fails to teach or suggest this feature. Indeed, Huitema fails to teach or suggest both a secure domain name and an equivalent non-secure domain name corresponding to the secure domain name. The Examiner refers to col. 4 of Huitema, but this merely discloses various data sets containing domain names. Huitema further discloses that an address such as "192.4.18.101" can be used to build the inverse name "101.18.4.192.in-addr.arpa.". However, this pair of addresses is not a secure domain name and a corresponding non-secure domain name as required by claim 13.

Claim 13 further recites registering the secure domain name in a secure domain name service

9

when the ownership information for the equivalent non-secure domain name is consistent with ownership information for the secure domain name. Huitema also fails to teach or suggest this feature. Huitema is unconcerned with whether any ownership information is consistent with anything else, much less whether ownership information for a non-secure domain name is consistent with ownership information for a secure domain name. Again, Huitema does not even disclose the secure and corresponding non-secure domain names as required by claim 13.

Independent claims 15, 17, and 21 are also allowable for at least similar reasons as those set forth above with regard to claim 13, and further in view of the various differing features recited therein.

Claims 14 and 24, which depend from claim 13, claims 16 and 25, which depend from claim 15, claims 18-20, which depend from claim 17, and claims 22 and 23, which depend from claim 21, are also allowable for at least those reasons set forth above with regard to their respective independent claims, and further in view of the additional features recited therein.

For example, claims 18 and 22 recite that the first domain name comprises a non-standard top-level domain and the second domain name comprises a standard top-level domain. The Examiner concedes that Huitema fails to teach or suggest this feature. Alkhatib similarly fails to teach or suggest this feature because, as previously discussed, Alkhatib fails to teach or suggest a non-standard top-level domain.

10

For at least the reasons set forth above, Applicants submit that the pending claims distinguish over the applied references, and are in condition for allowance. Should the Examiner feel that further discussion and/or amendment would be helpful in the prosecution of this application, the Examiner is invited to telephone the Applicants' undersigned representative at the number appearing below.
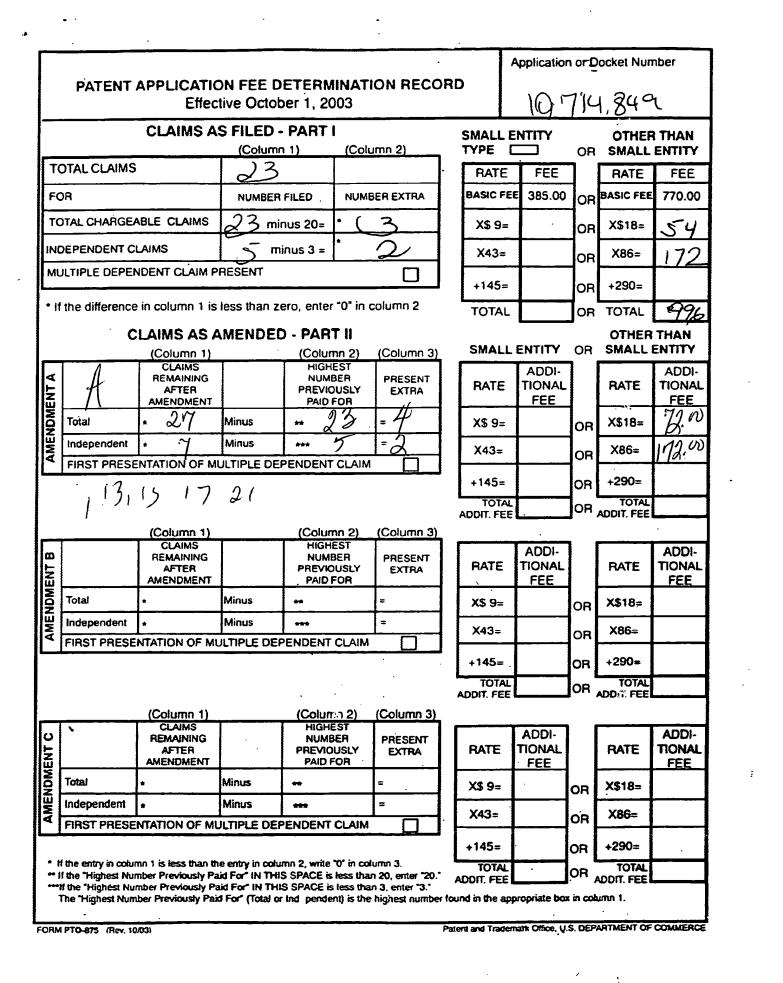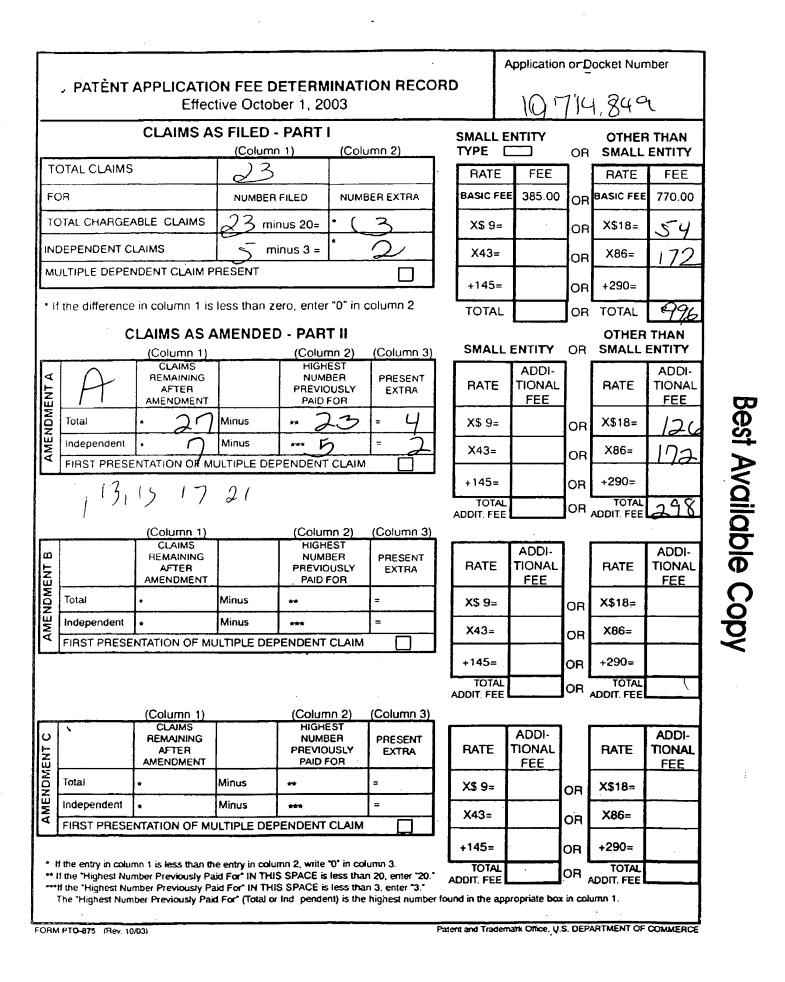
Respectfully Submitted,

By:_____

Jordan N. Bodner
Registration No. 42,338

BANNER & WITCOFF, LTD.
1001 G Street, N.W., 11th Floor
Washington, D.C. 20001
(202) 824-3000

Dated: May 18, 2004

11

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2003

Application or Docket Number: 10 714,849

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	23	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	23 minus 20=	* 3
INDEPENDENT CLAIMS	5 minus 3 =	* 2
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	385.00	OR	BASIC FEE	770.00
X$ 9=		OR	X$18=	54
X43=		OR	X86=	172
+145=		OR	+290=	
TOTAL		OR	TOTAL	996

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	* 27	Minus	** 23	= 4
Independent	* 7	Minus	*** 5	= 2
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

/ 13, 15 17 21

SMALL ENTITY OR **OTHER THAN SMALL ENTITY**

RATE	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
X$ 9=		OR	X$18=	72.00
X43=		OR	X86=	172.00
+145=		OR	+290=	
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	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
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	ADDI-TIONAL FEE		RATE	ADDI-TIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
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. 10/03)　　　　Patent and Trademark Office, U.S. DEPARTMENT OF COMMERCE

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2003

Application or Docket Number: 10/714,849

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	23	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	23 minus 20=	* 3
INDEPENDENT CLAIMS	5 minus 3 =	* 2
MULTIPLE DEPENDENT CLAIM PRESENT		☐

SMALL ENTITY TYPE ☐ **OR OTHER THAN SMALL ENTITY**

RATE	FEE	OR	RATE	FEE
BASIC FEE	385.00	OR	BASIC FEE	770.00
X$ 9=		OR	X$18=	54
X43=		OR	X86=	172
+145=		OR	+290=	
TOTAL		OR	TOTAL	996

* If the difference in column 1 is less than zero, enter "0" in column 2

CLAIMS AS AMENDED - PART II

OTHER THAN
SMALL ENTITY OR SMALL ENTITY

AMENDMENT A

A	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 27	Minus	** 23	= 4
Independent	* 7	Minus	*** 5	= 2
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

1 13, 15 17 21

RATE	ADDITIONAL FEE	OR	RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	126
X43=		OR	X86=	172
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	298

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	OR	RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
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	OR	RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
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.

Best Available Copy

PATENT APPLICATION

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:)	
)	
Victor LARSON, et al.)	Group Art Unit: 2143
)	
Serial No.: 10/714,849)	Examiner: TBD
)	
Filed: November 18, 2003)	
)	
For: IMPROVEMENTS TO AN AGILE)	
NETWORK PROTOCL FOR SECURE)	
COMMUNICATIONS WITH ASSURED)	
SYSTEM AVAILABILITY)	

INFORMATION DISCLOSURE STATEMENT

Commissioner for Patents
Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Sir:

   This Information Disclosure Statement is submitted for consideration by the United States Patent and Trademark Office in accordance with the duty of disclosure. The references listed in the PTO/SB/08a form were cited in a European Search Report from a related application. A copy of the Search Report is also submitted herewith.

   Transmitted herewith is:

[X] A PTO/SB/08a form

[ ] A Background Art Information Paper

[X] A copy of each item of information listed on the PTO 1449 form

   [ ] Concise explanation of non-English language information

      [ ] concise explanation of relevance:

      [ ] English translation of foreign language information

      [ ] English language version of foreign patent office report citing
   information

      [ ] English language abstract(s) of foreign patent document(s)

[ ] This application relies, under 35 U.S.C. §120, on the earlier filing date of U.S.
   Application Serial No. ****, filed ***. The information identified on the attached PTO

1449 form was submitted to and/or cited by the Patent and Trademark Office in this earlier application and, therefore, copies are not required to be provided in this application.

1.   [X]   This Information Disclosure Statement is filed <u>before</u> the later of:

- three months from the filing date of this national application;
- three months from the date of entry of the national stage as set forth in 37 CFR 1.491 in an international application; or
- the mailing date of a first office action on the merits.

In accordance with 37 CFR 1.97(b), no fee is required.

2.   [ ]   This Information Disclosure Statement is filed <u>after</u> the events noted in paragraph 1, but <u>before</u> either:

- a final action under 37 CFR 1.113, or an action that otherwise closes prosecution,
- a Notice of Allowance under 37 CFR 1.311.

In accordance with 37 CFR 1.97(c), also enclosed is

[ ]   a fee under 37 CFR 1.17(p) for $180.00; or

[ ]   a certification as stated below.

3.   [ ]   This Information Disclosure Statement is filed <u>before</u> payment of the issue fee, but <u>after</u> either

- a final action under 37 CFR 1.113, or an action that otherwise closes prosecution;
- a Notice of Allowance under 37 CFR 1.311.

In accordance with 37 CFR 1.97(d), also enclosed is:

[ ]   a certification, as stated below; and

[ ]   the IDS processing fee set forth in 37 CFR 1.17(p)(1) in the amount of $180.00.

<u>Certification under 1.97(e)1 or 1.97(e)2</u>

The undersigned states:

[ ]   Each item of Information contained in this Information Disclosure Statement was cited in a communication from a foreign Patent Office in a counterpart foreign application not more than three months prior to the filing date of this Information Disclosure Statement (e1 certification); or

2

[ ]   No item of information contained in this Information Disclosure Statement was first cited in any communication from a foreign Patent Office in a counterpart foreign application or, to the knowledge of the undersigned, having made reasonable inquiry, was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing date of this Information Disclosure Statement (e2 certification).

TOTAL FEES: $

[ ]   Please charge Deposit Account No. 19-0733 in the amount of $.

[X]   No fee is required.

[X]   Please charge or credit Deposit Account No. 19-0733 for any deficiency or overpayment associated with this Information Disclosure Statement.

A duplicate copy of this sheet is enclosed for accounting purposes.

It is respectfully requested that the Examiner fully consider each item of information, initial the enclosed Form PTO-1449 in the appropriate place to indicate that the information has been considered, and return a copy of the initialed form to the undersigned in accordance with MPEP Section 609.

Respectfully submitted,

BANNER & WITCOFF, LTD.   By:
Suite 1100   Jordan N. Bodner
1001 G Street, N.W.   Registration No. 42,338
Washington, D.C. 20001-4597
(202) 824-3000
Dated: June 3, 2005

3

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

Sheet	1	of	2

Complete if Known	
Application Number	10/714,849
Filing Date	November 18, 2003
First Named Inventor	Victor LARSON
Art Unit	2143
Examiner Name	TBA
Attorney Docket Number	000479.00111

U.S. PATENT DOCUMENTS

Examiner Initials *	Cite No.[1]	Document Number — Number - Kind Code[2] (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
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.[1]	Foreign Patent Document — Country Code[3] - Number[4] - Kind Code[5] (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	T[6]
		EP 0 838 930 A2	04/29/1998	Kenneth F. ALDEN, et al.		
		DE 199 24 575 A1	12/02/1999	Joseph E. PROVINO		

Examiner Signature		Date Considered	

PTO/SB/08b(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

Substitute for form 1449B/PTO	**INFORMATION DISCLOSURE STATEMENT BY APPLICANT** *(Use as many sheets as necessary)*		**Complete if Known**	
			Application Number	10/714,849
			Filing Date	November 18, 2003
			First Named Inventor	Victor LARSON
			Art Unit	2143
			Examiner Name	TBD
Sheet	2	of	2	Attorney Docket Number	000479.00111

NON PATENT LITERATURE DOCUMENTS

Examiner Initials *	Cite No.[1]	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.	T[2]
		Donald E. EASTLAKE, III, "Domain Name System Security Extensions", Internet Draft, April 1998.	
		P. SRISURESH, et al., "DNS Extensions to Network Address Translators', Internet Draft, July 1998.	
		D.B. CHAPMAN, et al., "Building Internet Firewalls, chapters 8 and 10 (parts)", pp. 278-296 and pp. 351-375.	

Examiner Signature		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.
This collection of information is required by 37 CFR 1.98. 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 take 120 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.

(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.
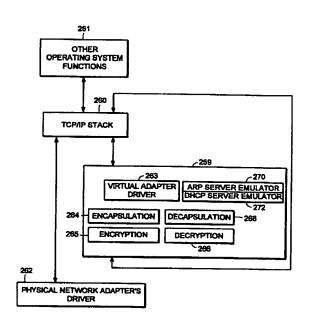
FIG. 15

Petitioner RPX Corporation - Ex. 1002, p. 176

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 exemplary embodiment of steps performed to perform session key management for a tunnel connection;

◊ Fig. 6 is a block diagram showing an exemplary embodiment of a relay frame;

◊ Fig. 7 is a block diagram showing an exemplary embodiment of a connection request frame;

◊ Fig. 8 is a block diagram showing an exemplary embodiment of a connection response frame;

◊ Fig. 9 is a block diagram showing an exemplary embodiment of a data frame;

◊ Fig. 10 is a block diagram showing an exemplary embodiment of a close connection frame;

◊ Fig. 11 is a state diagram showing an exemplary 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 exemplary embodiment of a state machine forming a tunnel connection in a server computer;

◊ Fig. 13 is a state diagram showing an exemplary embodiment of a state machine forming a tunnel connection in a relay node;

◊ Fig. 14 is a block diagram showing an exemplary 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 exemplary embodiment of a pseudo network adapter;

◊ Fig. 16 is a block diagram showing an exemplary embodiment of a pseudo network adapter;

◊ Fig. 17 is a flow chart showing steps performed by an exemplary embodiment of a pseudo network adapter during packet transmission;

◊ Fig. 18 is a flow chart showing steps performed by an exemplary embodiment of a pseudo network adapter during packet receipt;

◊ Fig. 19 is a data flow diagram showing data flow in an exemplary embodiment of a pseudo network adapter during packet transmission;

◊ Fig. 20 is a data flow diagram showing data flow in

an exemplary embodiment of a pseudo network adapter during packet receipt;

◊ Fig. 21 is a diagram showing the movement of encrypted and unencrypted data in an exemplary embodiment of a system including a pseudo network adapter;

◊ Fig. 22 is a diagram showing the movement of encrypted and unencrypted data in an exemplary embodiment of a system including a pseudo network adapter; and

◊ Fig. 23 is a flow chart showing steps initialization of an exemplary 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 Edi-

tion, 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 stand-alone 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 down-stream 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 discon-

nects 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 request 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 applica-

25   EP 0 838 930 A2   26

tion. 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.

Petitioner RPX Corporation - Ex. 1002, p. 189

14

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.

FIG. 1

FIG. 2

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

20

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
	— 138
STRING BUFFER	

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

FIG. 11

23

FIG. 12

FIG. 13

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

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
GENERATE ARP RESPONSE

324
ARP PACKET ? —YES→

NO

332
GENERATE DHCP RESPONSE

330
DHCP PACKET ? —YES→

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

FIG. 19

FIG. 20

466

TUNNEL
APPLICATION

450

USER
APPLICATION

·········· PLAINTEXT

— — — ENCRYPTED

WINSOCK — 452

TCP — 454

456
IP

458 — NDIS MAC

468 — PHYSICAL

460

VIRTUAL
DEVICE
DRIVER

ARP — 462

DHCP — 464

459

NETWORK — 470

FIG. 21

486

DAEMON

472

USER
APPLICATION

·········· PLAINTEXT

— — — ENCRYPTED

UNIX SOCKET
LAYER — 474

TCP — 476

478 — IP

488 — DATALINK

490 — PHYSICAL

482

VIRTUAL
DEVICE
DRIVER

DHCP — 484

480

NETWORK — 492

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

(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 A3

EUROPEAN SEARCH REPORT

European Patent Office

Application Number

EP 97 11 8556

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)
Y	BELLOVIN, S.M.: "Pseudo Network Drivers and Virtual Networks" PROCEEDINGS OF THE 1990 {USENIX} CONFERENCE, [Online] 1990, pages 1-15, XP002224471 Murray Hill, New Jersey Retrieved from the Internet: \<URL:www.research.att.com/~smb/papers/pnet.ext.pdf > [retrieved on 2002-12-10] * abstract * * page 1, paragraph 1 * * page 2, paragraph 2 * * page 4, line 3,4 * * page 5, paragraphs 4-8 * * page 6, paragraphs 1,7 * ---	1-15	H04L29/06 H04L29/12 H04L12/46
Y	XUN QU ET AL: "A practical method for achieving portable communications in the Internet context" GLOBAL TELECOMMUNICATIONS CONFERENCE, 1995. CONFERENCE RECORD. COMMUNICATION THEORY MINI-CONFERENCE, GLOBECOM '95., IEEE SINGAPORE 13-17 NOV. 1995, NEW YORK, NY, USA,IEEE, US, 13 November 1995 (1995-11-13), pages 1512-1516, XP010164473 ISBN: 0-7803-2509-5 * page 1514, column 1, line 1-10 * ---	1-4, 6-11, 13-15	
Y	BRUCE SCHNEIER: "Applied Cryptography, protocols, algorithms and source code in C, second edition" 18 October 1995 (1995-10-18) , JOHN WILEY & SONS , NEW YORK XP002224472 * page 29 - page 31 * * page 193 - page 206 * ---	5,12	**TECHNICAL FIELDS SEARCHED** (Int.Cl.6) H04L

-/--

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	11 December 2002	Hackl, A

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 97 11 8556

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)
L	ONLINE, [Online] XP002224551 Retrieved from the Internet: <URL:http://btobsearch.barnesandnoble.com/ booksearch/isbnInquiry.asp?userid=18E5DV2S QM&btob=Y&isbn=0471128457> [retrieved on 2002-12-11] Used to verify publication date of the Book "Applied Cryptography" -----		
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
MUNICH	11 December 2002	Hackl, A

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

EPC FORM 1503 03.82 (P04C01)

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

❑ **BLACK BORDERS**

❑ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

❑ **FADED TEXT OR DRAWING**

❑ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

❑ **SKEWED/SLANTED IMAGES**

❑ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

❑ **GRAY SCALE DOCUMENTS**
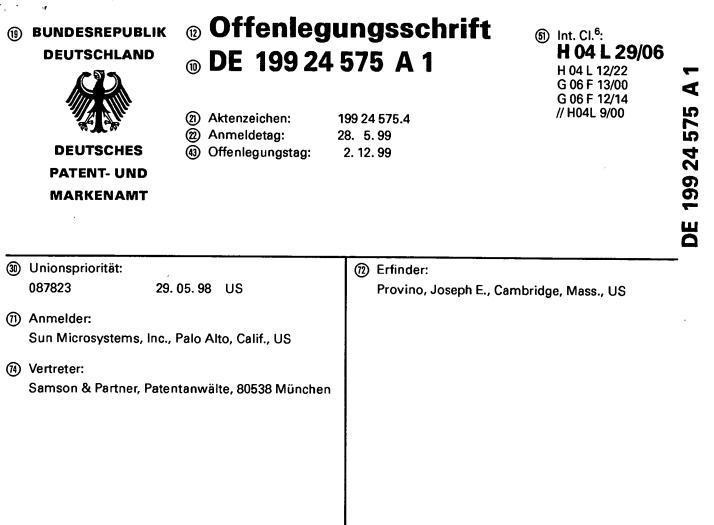
❑ **LINES OR MARKS ON ORIGINAL DOCUMENT**

❑ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

❑ **OTHER:** _____
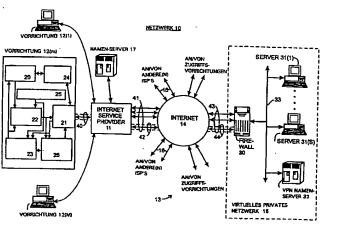
IMAGES ARE BEST AVAILABLE COPY.
As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

�554 Kommunikationssystem und -Verfahren

�557 Das erfindungsgemäße System umfaßt ein virtuelles privates Netzwerk (15) und eine externe Vorrichtung (12(m)), welche durch ein digitales Netzwerk (14) miteinander verbunden sind. Das virtuelle private Netzwerk (15) weist eine Firewall (30), wenigstens eine interne Vorrichtung (31(s)) und einen Namen-Server (32) auf, welche jeweils eine Netzwerkadresse besitzen. Die interne Vorrichtung (31(s)) besitzt auch eine Sekundäradresse, und der Namen-Server (32) ist derart konfiguriert, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt. In Reaktion auf eine Anfrage von der externen Vorrichtung (12(m)) zum Aufbau einer Verbindung zur Firewall (30) übermittelt die Firewall (30) der externen Vorrichtung (12(m)) die Netzwerkadresse des Namen-Servers (32). In Reaktion auf eine Anfrage von einem Bediener oder ähnlichem, welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält und einen Zugriff an die interne Vorrichtung (31(s)) anfordert, erzeugt die externe Vorrichtung (12(m)) eine Netzwerkadressen-Anfragenachricht zur Übertragung über die Verbindung an die Firewall (30), welche eine Auflösung der Netzwerkadresse, die der Sekundäradresse zugeordnet ist, anfordert. Die Firewall (30) übermittelt die Adressenauflösungsanfrage an den Namen-Server (32), und der Namen-Server (32) übermittelt die Netzwerkadresse, welche der Sekundäradresse zugeordnet ist, an die Firewall (30). Daraufhin stellt die Firewall (30) die Netzwerkadresse in einer ...

DE 199 24 575 A 1

BEST AVAILABLE COPY

Beschreibung

Die Erfindung betrifft allgemein das Gebiet der digitalen Kommunikationssysteme und -verfahren, und insbesondere Systeme und Verfahren zum Vereinfachen der Kommunikation zwischen Vorrichtungen, welche mit öffentlichen Netzwerken verbunden sind, z. B. dem Internet, und Vorrichtungen, welche mit privaten Netzwerken verbunden sind.

Digitale Netzwerke wurden entwickelt, um die Übertragung von Information, welche auch Daten und Programme umfaßt, über digitale Computersysteme und andere Digitalvorrichtungen zu ermöglichen. Es wurde eine Vielzahl von Arten von Netzwerken entwickelt und realisiert, einschließlich sog. Fernverbindungsnetze (Wide-Area Networks, nachfolgend "WAN" genannt) und lokale Netzwerke (Local Area Networks, nachfolgend "LAN" genannt), welche eine Information unter Verwendung verschiedener Informationsübertragungsmethoden übermitteln. Im allgemeinen werden LANs innerhalb kleiner geographischer Bereiche realisiert, z. B. innerhalb eines einzelnen Bürogebäudes oder ähnlichem, zum Übertragen von Information innerhalb eines bestimmten Büros, einer Firma oder einer ähnlichen Art von Organisationseinheit. Andererseits werden WANs im allgemeinen auf relativ großen geographischer Bereichen realisiert und können verwendet werden, um Information sowohl zwischen LANs als auch zwischen Vorrichtungen, welche nicht mit LANs verbunden sind, zu übertragen. Derartige WANs umfassen auch öffentliche Netzwerke, z. B. das Internet, welche zur Informationsübertragung zwischen einer Anzahl von Unternehmen verwendet werden können.

Es sind mehrere Probleme im Zusammenhang der Kommunikation über ein Netzwerk aufgetreten, insbesondere in einem großen öffentlichen WAN, wie es z. B. das Internet ist. Im allgemeinen werden Informationen über ein Netzwerk in Nachrichtenpaketen übertragen, welche ausgehend von einer Vorrichtung, als Quelle bzw. Quellenvorrichtung, zu einer anderen Vorrichtung, als Ziel bzw. Zielvorrichtung, über einen oder mehrere Router oder allgemein Schaltungsknoten im Netzwerk übertragen werden. Jedes Nachrichtenpaket enthält eine Zieladresse, welche von den Schaltungsknoten verwendet wird, um das jeweilige Nachrichtenpaket an die geeignete Zielvorrichtung zu leiten. Z.B. im Internet haben solche Adressen die Form von "n"-Bit Zahlen (wobei "n" 32 oder 128 sein kann), wobei solche Zahlenkolonnen für einen Benutzer schwierig sind zu merken und einzugeben, wenn die oder der Benutzer die Übertragung eines Nachrichtenpakets veranlassen möchte. Um einen Benutzer von der Notwendigkeit zu befreien, sich solche spezifische Zahlen-Internetadressen zu merken und einzugeben, stellt das Internet einen zweiten Adressierungsmechanismus bereit, der durch Benutzer der jeweiligen Vorrichtungen einfacher handzuhaben ist. Bei diesem Adressierungsmechanismus werden Internet-Domains, wie etwa LANs, Internet-Service-Provider (nachfolgend "ISP" genannt) und ähnliche, welche im Internet verbunden sind, durch für einen Benutzer relativ einfach les- und merkbare Namen identifiziert, die nachfolgend als "Klartextnamen" bezeichnet werden. Um den Einsatz von solchen Klartextnamen umzusetzen, werden Namen-Server, auch als DNS-Server für "Domain Name Server" bezeichnet, bereitgestellt, um die Klartextnamen in die geeigneten Internetadressen umzuwandeln. Wenn ein Bediener einer Vorrichtung, der die Übertragung eines Nachrichtenpakets an eine andere Vorrichtung wünscht, den Klartextnamen der anderen Vorrichtung eingibt, nimmt die Vorrichtung zuerst Kontakt mit einem Namen-Server auf. Im allgemeinen kann der Namen-Server ein Teil des ISP selbst sein oder er kann eine spezielle Vorrichtung sein, welche durch den ISP über das Internet zugäng-

lich ist; in jedem Fall wird der ISP den Namen-Server identifizieren, welcher für die Vorrichtung zu verwenden ist, wenn sich die Vorrichtung beim ISP einloggt, d. h. anmeldet. Falls der Namen-Server, nachdem die Vorrichtung einen Kontakt hergestellt hat, eine Zahlen-Internetadresse für den Klartext-Domainnamen besitzt oder erhalten kann, übermittelt der Namen-Server die Zahlen-Internetadresse, welche dem Klartext-Domainnamen entspricht, zu der Vorrichtung des Bedieners. Die Vorrichtung kann sodann die Zahlen-Internetadresse, welche von dem Namen-Server zurückgesendet wurde, in das Nachrichtenpaket einfügen und das Nachrichtenpaket an den ISP für die Übertragung über das Internet auf konventioneller Weise liefern. Die Internet-Schaltungsknoten verwenden die Zahlen-Internetadresse, um das Nachrichtenpaket an die gewünschte Zielvorrichtung zu übermitteln.

Andere Probleme treten insbesondere in Verbindung mit der Übertragung von Information über ein öffentliches WAN, z. B. das Internet, auf. Ein Problem besteht darin, sicherzustellen, daß die über das WAN übertragene Information, welche die Quellenvorrichtung und die Zielvorrichtung vertraulich behalten möchten, auch tatsächlich vertraulich bleibt gegenüber möglichen Lauschern, welche die Information abfangen können. Um die Vertraulichkeit zu wahren, wurden verschiedene Formen von Verschlüsselung entwickelt und werden verwendet, um die Information vor der Übertragung durch die Quellenvorrichtung zu verschlüsseln und die Information nach deren Empfang durch die Zielvorrichtung zu entschlüsseln. Falls gewünscht wird, daß beispielsweise die gesamte Information, welche zwischen einer bestimmten Quellenvorrichtung und einer bestimmten Zielvorrichtung übertragen wird, vertraulich bleiben soll, können die Vorrichtungen einen sog. "Sicherheitstunnel" zwischen den Vorrichtungen einrichten, der im wesentlichen sicherstellt, daß die gesamte Information, welche von der Quellenvorrichtung an die Zielvorrichtung übertragen wird, vor der Übertragung verschlüsselt wird (mit Ausnahme von bestimmten Protokollinformationen, wie Adresseninformation, welche den Fluß von Netzpaketen über das Netzwerk zwischen der Quellen- und Zielvorrichtung steuert), und daß die verschlüsselte Information vor der Verwendung durch die Zielvorrichtung entschlüsselt wird. Die Quellen- und Zielvorrichtungen können jeweils für sich eine Verschlüsselung bzw. Entschlüsselung durchführen, oder die Verschlüsselung und Entschlüsselung kann durch andere Vorrichtungen durchgeführt werden, bevor die Nachrichtenpakete über das Internet übertragen werden.

Ein weiteres Problem, welches insbesondere im Zusammenhang mit Unternehmen, Regierungsämtern und privaten Organisationen auftritt, deren private Netzwerke, welche LANs, WANs oder etwaige Kombinationen derselben sein können, mit öffentlichen WANs, z. B. dem Internet, verbunden sind, besteht darin, sicherzustellen, daß deren private Netzwerke sicher sind gegenüber anderen Netzwerken, zu welchen z. B. die Unternehmen keinen Zugriff haben möchten, oder einen Zugriff durch andere zu regulieren und zu kontrollieren, zu welchen z. B. die jeweiligen Organisationen einen begrenzten Zugriff haben möchten. Um dies umzusetzen, verbinden die Organisationen in der Regel ihre privaten Netzwerke mit öffentlichen WANs über eine begrenzte Anzahl von Gateways, welche manchmal als "Firewalls" bezeichnet werden, durch welche der gesamte Netzverkehr zwischen dem internen und dem öffentlichen Netzwerk läuft. In der Regel sind Netzwerkadressen von Domains und Vorrichtungen in dem privaten Netzwerk "hinter" der Firewall den Namen-Servern bekannt, welche in den privaten Netzwerken vorgesehen sind; sie sind aber nicht zugänglich für Namen-Server oder andere Vorrichtungen au-

ßerhalb der privaten Netzwerke, was die Kommunikation zwischen einer Vorrichtung außerhalb des privaten Netzwerkes und einer Vorrichtung innerhalb des privaten Netzwerkes schwierig macht.

Ein Ziel der vorliegend Erfindung ist es, hier Abhilfe zu schaffen.

Dieses Ziel erreicht die Erfindung durch die Gegenstände der Ansprüche 1, 7 und 13. Bevorzugte Ausführungsbeispiele der Erfindung sind in den jeweils abhängigen Ansprüchen beschrieben.

Danach schafft die Erfindung ein neuartiges und verbessertes System und ein Verfahren zum Vereinfachen von Kommunikation zwischen Vorrichtungen, welche mit öffentlichen Netzwerken, z. B. dem Internet, verbunden sind, und Vorrichtungen, welche mit privaten Netzwerken verbunden sind, wobei die Auflösung von Sekundäradressen, wie etwa Text- bzw. Klartextnamen im Internet, in die zugehörigen Netzwerkadressen durch Namen-Server oder ähnliche Vorrichtungen, die mit den privaten Netzwerken verbunden sind, ermöglicht wird.

Hierfür stellt die Erfindung ein System zur Verfügung mit einem virtuellen Privaten Netzwerk und einer externen Vorrichtung, welche durch ein digitales Netzwerk miteinander verbunden sind, sowie ein Kommunikationsverfahren und ein Computerprogrammprodukt zum gemeinsamen Verwenden mit einem derartiges System. Das virtuelle private Netzwerk weist eine Firewall bzw. ein Firewall-System, wenigstens eine interne Vorrichtung und einen Namen-Server auf, welche jeweils eine Netzwerkadresse besitzen. Die interne Vorrichtung besitzt ferner eine Sekundäradresse, und der Namen-Server ist derart konfiguriert, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt. In Reaktion auf eine Anfrage von der externen Vorrichtung zum Aufbau einer Verbindung zur Firewall übermittelt die Firewall der externen Vorrichtung die Netzwerkadresse des Namen-Servers. In Reaktion auf eine Anfrage von einem Bediener oder ähnlichem, welche die Sekundäradresse der internen Vorrichtung enthält und einen Zugriff an die interne Vorrichtung anfordert, erzeugt die externe Vorrichtung eine Netzwerkadressen-Anfragenachricht zur Übertragung über die Verbindung an die Firewall, welche eine Auflösung der Netzwerkadresse, die der Sekundäradresse zugeordnet ist, anfordert. Die Firewall übermittelt die Adressenauflösungsanfrage an den Namen-Server und der Namen-Server übermittelt die Netzwerkadresse, welche der Sekundäradresse zugeordnet ist, an die Firewall. Daraufhin stellt die Firewall die Netzwerkadresse in einer Netzwerkadressenantwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung bereit. Die externe Vorrichtung kann sodann die auf diese Weise bereitgestellte Netzwerkadresse in nachfolgenden an die interne Vorrichtung gerichtete Kommunikationen mit der Firewall verwenden.

Weitere Vorteile und Ausgestaltungen der Erfindung ergeben sich aus der nachfolgenden detaillierten Beschreibung eines bevorzugten Ausführungsbeispiels. In der Beschreibung wird auf die beigefügte schematische Zeichnung Bezug genommen. Darin zeigt:

Fig. 1 ein funktionelles Blockdiagramm eines erfindungsgemäßen Netzwerkes.

Fig. 1 zeigt ein funktionelles Blockdiagramm eines Netzwerkes **10**, welches gemäß der vorliegenden Erfindung aufgebaut ist. Das Netzwerk **10** gemäß **Fig. 1** umfaßt einen Internet-Service-Provider (nachfolgend "ISP") **11**, welcher die Übertragung von Nachrichtenpaketen zwischen einer oder mehreren Vorrichtungen **12(1)** bis **12(M)** (nachfolgend allgemeinen mit dem Bezugszeichen **12(m)** identifiziert), welche mit dem ISP **11** verbunden sind, und anderen Vorrich-

tungen, welche allgemein durch ein Bezugszeichen **13** gekennzeichnet sind, über das Internet **14** ermöglicht, wobei die Übertragung von Information in Nachrichtenpaketen zwischen den Vorrichtungen **12(m)** und **13** realisiert wird. Der ISP **11** verbindet das Internet **14** über eine oder mehrere logische Verbindungen oder Gateways oder ähnlichem (im vorliegenden allgemein als "Verbindungen" bezeichnet), welche allgemein durch das Bezugszeichen **41** gekennzeichnet sind. Der ISP **11** kann ein öffentlicher ISP sein, welcher in diesem Falle die Verbindung mit Vorrichtungen **12(m)** herstellt, welche durch Bediener betrieben werden können, die der allgemeinen Öffentlichkeit angehören, so daß diese Bediener Zugang zu dem Internet erlangen. Alternativ dazu kann der ISP **11** ein privater ISP sein. In diesem Falle werden die damit verbundenen Vorrichtungen **12(m)** im allgemeinen beispielsweise durch Angestellte eines bestimmten Unternehmens oder einer Regierungseinrichtung, Mitgliedern von einer privaten Organisation oder ähnlichen betrieben, um diesen Angestellten oder Mitgliedern einen Zugang in das Internet bereit zu stellen.

In an sich konventioneller Weise weist das Internet ein Netz von Schaltungsknoten auf (welche nicht separat dargestellt sind), welche die ISPs **11** und die Vorrichtungen **13** miteinander verbinden, um dazwischen die Übertragung von Nachrichtenpaketen zu ermöglichen. Die Nachrichtenpakete, welche über das Internet **14** übertragen werden, stimmen mit denjenigen überein, welche durch das sog. Internetprotokoll (IP) definiert werden, und umfassen einen Kopfabschnitt, einen Datenabschnitt und können einen Fehlererfassungs- und/oder Korrekturabschnitt aufweisen. Der Kopfabschnitt enthält Information, welche verwendet wird, um das Nachrichtenpaket über das Internet **14** zu übertragen, beispielsweise eine Zieladresse, welche die Vorrichtung identifiziert, welche das Nachrichtenpaket als Zielvorrichtung empfangen soll, und eine Quellenadresse, welche diejenige Vorrichtung identifiziert, welche das Nachrichtenpaket erzeugt hat. In jedem Nachrichtenpaket haben die Ziel- und Quellenadresse jeweils die Form einer Zahl, welche eindeutig die jeweilige Ziel- bzw. Quellenvorrichtung identifiziert. Die Schaltungsknoten im Internet **14** verwenden wenigstens die Zieladresse eines jeweiligen Nachrichtenpaketes, um das jeweilige Nachrichtenpaket an die Zielvorrichtung zu übermitteln, wenn die Zielvorrichtung an das Internet angeschlossen ist, oder an einen ISP **11** oder andere Vorrichtungen, welche an das Internet **14** angeschlossen sind, welche sodann das Nachrichtenpaket an das geeignete Ziel senden werden. Der Datenabschnitt eines jeden Nachrichtenpakets enthält die in dem Nachrichtenpaket übertragenen Daten; und der Fehlererfassungs- und/oder Korrekturabschnitt enthält Fehlererfassungs- und/oder Korrekturinformationen, welche verwendet werden können, um zu verifizieren, daß das Nachrichtenpaket in korrekter Weise von der Quelle zu der Zielvorrichtung übertragen wurde (im Fall der Fehlererfassungsinformation), und um ausgewählte Arten von Fehlern zu korrigieren, falls das Nachrichtenpaket nicht korrekt übertragen wurde (im Falle der Fehlerkorrekturinformation).

Die Vorrichtungen **12(m)**, welche mit dem ISP **11** verbunden sind, können jede beliebige Anzahl von Arten von Vorrichtungen umfassen, welche über das Internet **14** mit anderen Vorrichtungen **13** kommunizieren, umfassend z. B. Personalcomputer, Computer-Workstations und ähnliches. Jede Vorrichtung **12(m)** kommuniziert mit dem ISP **11**, um Nachrichtenpakete für die Übertragung über das Internet **14** an diesen zu übertragen, oder um Nachrichtenpakete, welche durch den ISP **11** über das Internet empfangen werden, von diesem zu empfangen. Dabei kann jedes geeignete Protokoll verwendet werden, z. B. das bekannte Point-to-Point Proto-

koll (allgemein mit "PPP" abgekürzt), falls die Vorrichtung 12(m) über eine Point-to-Point Verbindung mit dem ISP 11 verbunden ist, oder irgendein konventionelles "Multi-Drop" Protokoll, falls die Vorrichtung 12(m) mit dem ISP 11 über ein "Multi-Drop"-Netzwerk, z. B. das Ethernet, verbunden ist, oder ähnliches. Die Vorrichtungen 12(m) sind im allgemeinen entsprechend der üblichen Computerarchitektur mit gespeicherten Programmen aufgebaut, welche z. B. eine Systemeinheit, eine Bildschirmanzeigeeinheit und Bedienereingabeeinrichtungen, wie etwa eine Tastatur oder eine Maus, umfaßt. Eine Systemeinheit weist im allgemeinen eine oder mehrere Prozessor-, Speicher-, Massenspeichereinrichtungen, z. B. Festplatten- und/oder Bandspeicherelemente, oder andere Elemente (nicht separat gezeigt) auf, wie etwa Netzwerk- und/oder Telephonschnittstelleneinrichtungen, um die jeweilige Vorrichtung an den ISP 11 anzukoppeln. Die Prozessor- bzw. Verarbeitungseinrichtungen verarbeiten Programme, einschließlich Anwendungsprogramme, unter der Steuerung eines Betriebssystems, um verarbeitete Daten zu erzeugen. Die Bildschirmeinheit ermöglicht es der Vorrichtung, die verarbeiteten Daten und einen Verarbeitungsstatus der Daten dem Benutzer anzuzeigen, und die Bedienereingabeeinrichtung ermöglicht es dem Bediener, Daten einzugeben und die Verarbeitung zu steuern.

Diese Elemente der Vorrichtung 12(m) arbeiten in Verbindung mit einer geeigneten Programmierung so zusammen, um eine Vorrichtung 12(m) mit einer Anzahl von funktionellen Elementen bereit zustellen, beispielsweise eine Bedienerschnittstelle 20, eine Netzwerkschnittstelle 21, einen Nachrichtenpaketgenerator 22, einen Nachrichtenpaketempfänger und -prozessor 23, eine ISP Einloggsteuerung bzw. Anmeldungssteuerung 24, einen Internetparameterspeicher 25 und im Zusammenhang mit der vorliegenden Erfindung einen Sicherheits-Nachrichtenpaketprozessor 26. Die Bedienerschnittstelle 20 ermöglicht, daß die Vorrichtung 12(m) Eingabeinformationen von der/den Bedienereingabevorrichtung(en) der Vorrichtung 12(m) empfängt und die Ausgabeinformationen dem Bediener auf der/den Bildschirmeinrichtung(en) der Vorrichtung 12(m) angezeigt werden. Die Netzwerkschnittstelle 21 ermöglicht eine Verbindung der Vorrichtung 12(m) mit dem ISP 11 unter Verwendung des geeigneten PPP oder Netzwerkprotokolls, um Nachrichtenpakete an den ISP 11 zu übertragen und von diesem Nachrichtenpakete zu empfangen. Die Netzwerkschnittstelle 21 kann eine Verbindung mit dem ISP 11 über das öffentliche Telefonnetz vorsehen, um einen Wählverbindungsnetzwerkbetrieb (sog. Dial-Up Betrieb) der Vorrichtung 12(m) über das öffentliche Telefonnetz zu ermöglichen. Alternativ oder zusätzlich dazu kann die Netzwerkschnittstelle 21 eine Verbindung durch den ISP 11 über beispielsweise ein konventionelles LAN ermöglichen, wie etwa das Ethernet. In Reaktion auf eine durch die Bedienerschnittstelle 20 gelieferte Eingabe und/oder in Reaktion auf Anfragen aus Programmen (nicht gezeigt), welche durch die Vorrichtung 12(m) verarbeitet werden, kommuniziert die ISP Einloggsteuerung 24 über die Netzwerkschnittstelle 21, um die Initialisierung (sog. "Log-On") einer Kommunikationssitzung zwischen der Vorrichtung 12(m) und dem ISP 11 zu ermöglichen. Während dieser Kommunikationssitzung kann die Vorrichtung 12(m) Information in der Form von Nachrichtenpaketen an andere Vorrichtungen über das Internet 14 sowie an andere Vorrichtungen 12(m') (wobei m' ≠ m), welche mit der ISP 11 oder mit anderen ISPs verbunden sind, übertragen. Während eines Log-On-Betriebs empfängt die ISP Einloggsteuerung 24 die Internetprotokollparameter (IP-Parameter), welche im Zusammenhang mit einer Nachrichtenpaketerzeugung während der Kommunikationssitzung verwendet werden.

Während einer Kommunikationssitzung erzeugt der Nachrichtenpaketgenerator 22 Nachrichtenpakete zur Übertragung durch die Netzwerkschnittstelle 21 in Reaktion auf eine Eingabe, welche durch den Bediener über die Bedienerschnittstelle 20 geliefert wird und/oder in Reaktion auf Anfragen aus Programmen (nicht separat gezeigt), welche durch die Vorrichtung 12(m) verarbeitet werden. Die Netzwerkschnittstelle 21 empfängt auch Nachrichtenpakete aus dem ISP 11 und liefert diese an den Nachrichtenpaketempfänger und -prozessor 23 zur Verarbeitung und Bereitstellung an die Bedienerschnittstelle 20 und/oder anderen Programmen (nicht gezeigt), welche durch die Vorrichtung 12(m) verarbeitet werden. Falls die empfangenen Nachrichtenpakete eine Information enthalten, z. B. Web-Seiten oder ähnliches, welche dem Bediener angezeigt werden soll, kann die Information der Bedienerschnittstelle 20 geliefert werden, damit die Information auf der Bildschirmeinheit der Vorrichtung angezeigt wird. Zusätzlich oder alternativ dazu kann die Information an andere Programme (nicht gezeigt) zur Verarbeitung geliefert werden, welche durch die Vorrichtung 12(m) verarbeitet werden.

Im allgemeinen können die Elemente, wie die Bedienerschnittstelle 20, der Nachrichtenpaketgenerator 22, der Nachrichtenpaketempfänger und -prozessor 23, die ISP Einloggsteuerung 24 und der Internetparameterspeicher 25 Elemente eines konventionellen Internet-Browsers enthalten, wie die von Mosaic, Netscape Navigator und Microsoft Internet Explorer.

Wie es oben erwähnt wurde, weist die Vorrichtung 12(m) im Zusammenhang mit der vorliegenden Erfindung einen Sicherheits-Nachrichtenpaketprozessor 26 auf. Der Sicherheits-Nachrichtenpaketprozessor 26 ermöglicht den Aufbau und Verwendung eines "Sicherheitstunnels" zwischen der Vorrichtung 12(m) und anderen Vorrichtungen 12(m') (wobei m' ≠ m) oder 13, wie es welches weiter unten beschrieben wird. Im allgemeinen wird in einem solchen Sicherheitstunnel Information in wenigstens dem Datenabschnitt der zwischen der Vorrichtung 12 (m) und einer spezifischen anderen Vorrichtung 12(m') (wobei m' ≠ m) oder 13 übertragenen Nachrichtenpakete geheimgehalten, beispielsweise durch Verschlüsselung des Datenabschnittes vor der Übertragung durch die Quellenvorrichtung. Die Information in anderen Abschnitten eines derartigen Nachrichtenpakets kann ebenfalls geheimgehalten werden, mit Ausnahme der Information, welche benötigt wird, um die Übertragung des jeweiligen Nachrichtenpakets zwischen den Vorrichtungen zu ermöglichen, also z. B. wenigstens die Zielinformation, damit die Schaltungsknoten des Internets und die ISPs die Vorrichtung identifizieren können, welche das Nachrichtenpaket empfangen soll.

Zusätzlich zu dem ISP 11 kann eine Vielzahl von anderen ISPs die Verbindung zum Internet herstellen, wie es durch die Pfeile 16 angedeutet ist, um eine Kommunikation zwischen Vorrichtungen, welche an diesen anderen ISPs angeschlossen sind, mit anderen Vorrichtungen über das Internet zu ermöglichen, welche die Vorrichtungen 12(n), welche an dem ISP 11 angeschlossen sind, umfassen können.

Die Vorrichtungen 13, auf welche die Vorrichtungen 12(m) zugreifen und mit welchen diese kommunizieren, können auch von jeder beliebigen Anzahl von Arten von Vorrichtungen sein, einschließlich Personalcomputer, Computer-Workstations und ähnliches, oder auch Minicomputer und Großrechner, Großspeichersysteme, Rechenserver, lokale Netzwerke (LANs) und Fernverbindungsnetzwerke (WANs), welche derartige Vorrichtungen und zahlreiche andere Arten von Vorrichtungen enthalten, die direkt oder indirekt mit den Netzwerken verbunden werden können. Nach der vorliegenden Erfindung umfaßt wenigstens eine der Vor-

richtungen wenigstens ein privates Netzwerk, welches als virtuelles privates Netzwerk 15 gekennzeichnet ist und z. B. die Form eines LAN oder eines WAN haben kann. Das virtuelle private Netzwerk 15 kann jede der Vorrichtungen 12(m') (wobei m' ≠ m) aufweisen (wobei die Verbindung zu dem Internet 14 über einen ISP erfolgt) oder der Vorrichtungen 13 (wobei die Verbindung zu dem Internet 14 unmittelbar erfolgt). Bei dem vorliegend beschriebenen Ausführungsbeispiel wird angenommen, daß das virtuelle Netzwerk 15 eine Vorrichtung 13 aufweist. Das virtuelle private Netzwerk 15 umfaßt selbst mehrere Vorrichtungen, welche hier als eine Firewall bzw. ein Firewall-System 30, mehrere Server 31(1) bis 31(S) (im nachfolgenden allgemein mit dem Bezugszeichen 31(s) angegeben) und ein Namen-Server 32 gekennzeichnet sind, wobei allesamt durch eine Übertragungsverbindung 33 miteinander verbunden sind. Die Firewall 30 und die Server 31(s) können ähnlich sein wie jede der verschiedenen Arten von Vorrichtungen 12(m) und 13, die hier beschrieben sind, und können daher beispielsweise umfassen Personalcomputer, Computer-Workstations und ähnliches, aber auch Minicomputer und Großrechner, Großspeichersysteme, Rechenserver, lokale Netzwerke (LANs) und Fernverbindungsnetzwerke (WANs), welche derartige Vorrichtungen und zahlreiche andere Arten von Vorrichtungen umfassen, welche direkt oder indirekt mit den Netzwerken verbunden werden können.

Wie oben ausgeführt wurde, kommunizieren diese Vorrichtungen einschließlich der Vorrichtungen 12(m) und der Vorrichtungen 13 durch Übertragung von Nachrichtenpaketen über das Internet. Die Vorrichtungen 12(m) und 13 können Information in einem Peer-to-Peer bzw. gleichrangigem Modus, in einem Client-Server Modus oder nach beiden dieser Modi übertragen. Im allgemeinen überträgt eine Vorrichtung in einer Peer-to-Peer Nachrichtenpaketübertragung Information in einem oder mehreren Nachrichtenpaketen an die andere Vorrichtung. Andererseits kann eine Vorrichtung, welche in einem Client-Server Modus als Client fungiert, ein Nachrichtenpaket an eine andere Vorrichtung übertragen, welche als Server fungiert, um beispielsweise einen Dienst durch die andere Vorrichtung auszulösen. Mehrere Arten derartiger Dienste sind dem Fachmann bekannt, beispielsweise das Wiedergewinnen bzw. Auslesen von Information aus der anderen Vorrichtung, damit diese aktiviert wird, um Verarbeitungsoperationen und dergleichen durchzuführen. Falls der Server dazu dient, dem Client vor allem Informationen zu liefern, kann dieser allgemein als ein Speicherserver bezeichnet werden. Falls der Server andererseits Verarbeitungsoperationen auf Anfrage des Client ausführen soll, kann dieser allgemein als ein Rechnerserver bezeichnet werden. Andere Arten von Servern zum Ausführen von anderen Arten von Diensten und Operationen auf Anfrage von Clients sind dem Fachmann ebenfalls bekannt.

Wenn in einer Client-Server Anordnung eine Vorrichtung 12(m) einen Dienst durch beispielsweise eine Vorrichtung 13 ausgeführt haben möchte, erzeugt die Vorrichtung 12(m) eines oder mehrere Anfragenachrichtenpakete zur Übertragung an die Vorrichtung 13, welche den benötigten Dienst anfordern. Das Anfragenachrichtenpaket enthält die Internetadresse der Vorrichtung 13, welche als die Zielvorrichtung das Nachrichtenpaket empfängt und den Dienst ausführt. Die Vorrichtung 12 (m) überträgt das/die Anfragenachrichtenpaket(e) an den ISP 11. Der ISP 11 überträgt daraufhin das Nachrichtenpaket über das Internet an die Vorrichtung 13.

Falls die Vorrichtung 13 die Form eines WAN oder LAN hat, empfängt das WAN oder LAN das/die Nachrichtenpaket(e) und leitet dieses/diese zu einer dort angeschlossenen Vorrichtung weiter, welche den angeforderten Dienst aus-

führen soll.

In jedem Fall wird die Vorrichtung 13, welche den angeforderten Dienst ausführen soll, nach Empfang des/der Anfragenachrichtenpaket(e) die Anfrage bearbeiten. Falls die Vorrichtung 12(m), welche das/die Anfragenachrichtenpaket(e) erzeugt hat, oder deren Bediener die notwendigen Befugnisse hat, um den Dienst von der Vorrichtung 13 anzufordern, und falls der angeforderte Dienst die Einleitung einer Informationsübertragung aus der Vorrichtung 13 als ein Speicherserver an die Vorrichtung 12(m) als ein Client umfaßt, erzeugt die Vorrichtung 13 eines oder mehrere Antwortnachrichtenpakete, welche die angeforderten Information enthalten, und überträgt das/die Paket(e) über das Internet 14 an den ISP 11. Daraufhin überträgt der ISP 11 das/die Nachrichtenpaket(e) an die Vorrichtung 12(m). Falls andererseits der angeforderte Dienst die Einleitung eines Verarbeitungsvorganges durch die Vorrichtung 13 als ein Rechenserver beinhaltet, wird die Vorrichtung 13 den/die angeforderten Rechendienst(e) ausführen. Falls die Vorrichtung 13 verarbeitete Daten, welche während den Rechenvorgängen erzeugt wurden, an die Vorrichtung 12(m) als Client zurücksenden soll, erzeugt die Vorrichtung 13 zusätzlich eines oder mehrere Antwortnachrichtenpakete, welche die verarbeiteten Daten enthalten und überträgt das/die Paket(e) über das Internet 14 an den ISP 11. Der ISP 11 überträgt daraufhin das/die Nachrichtenpaket(e) an die Vorrichtung 12(m). Entsprechende Operationen können durch die Vorrichtungen 12(m) und 13, dem ISP 11 und dem Internet 14 in Verbindung mit anderen Arten von Diensten ausgeführt werden, welche durch die Server-Vorrichtungen 13 bereitgestellt werden können.

Wie oben angemerkt wurde, enthält jedes Nachrichtenpaket, welches durch die Vorrichtungen 12(m) und 13 zur Übertragung über das Internet 14 erzeugt wird, eine Zieladresse, welche von den Schaltungsknoten verwendet wird, um das jeweilige Nachrichtenpaket an die geeignete Zielvorrichtung zu leiten. Adressen im Internet haben die Form von "n"-Bit Zahlen (wobei "n" beim gegenwärtigen Standard 32 oder 128 sein kann). Um insbesondere einen Bediener einer Vorrichtung 12(m) von der Notwendigkeit zu befreien, sich spezifische Zahlenkolonnen bzw. Zahlen-Internetadressen zu merken und diese der Vorrichtung 12(m) einzugeben, um die Erzeugung eines Nachrichtenpakets zur Übertragung über das Internet einzuleiten, stellt das Internet einen zweiten Adressierungsmechanismus zur Verfügung, welcher einfacher durch menschliche Bediener der jeweiligen Vorrichtungen handhabbar ist. Bei diesem Adressierungsmechanismus werden Internet-Domains, wie etwa LANs, Internet-Service-Provider (ISPs) und ähnliche, welche in bzw. mit dem Internet verbunden sind, durch relativ einfach les- und merkbare Namen, sog. Klartextnamen, identifiziert. Dabei soll sich hier die Bezeichnung "Klartextname" auf jede Art von Namenstext beziehen, z. B. auch auf Abkürzungen, generische Bezeichnungen, Phantasiebegriffe, etc. Um das System der Klartext-Domainnamen umzusetzen, ist der ISP 11 mit einem Namen-Server 17 (der auch als ein DNS Server (Domain Name Server) bezeichnet werden kann) verbunden, welcher die Klartext-Domainnamen auflösen bzw. in eine gültige Internetadresse umwandeln kann, um die geeignete Internetadresse für das in dem jeweiligen Klartextnamen angegebene Ziel bereitzustellen. Im allgemeinen kann der Namen-Server ein Teil des ISP 11 oder damit direkt verbunden sein, wie es in Fig. 1 gezeigt ist, oder er kann eine bestimmte Vorrichtung sein, welche durch den ISP über das Internet zugänglich ist. Jedenfalls wenn sich die Vorrichtung 12(m) bei dem ISP 11 während einer Kommunikationssitzung einloggt, wird der ISP 11, wie oben hingewiesen wurde, verschiedene Internet-Proto-

kollparameter (IP-Parameter) zuordnen, welche die Vorrichtung 12(m) während der Kommunikationssitzung verwendet, und welche in dem Internetparameterspeicher 25 gespeichert sind. Diese IP-Parameter enthalten Informationen, wie

(a) eine Internetadresse für die Vorrichtung 12(m), welche die Vorrichtung 12(m) während der Kommuniationssitzung identifiziert; und
(b) die Identifizierung eines Namen-Servers 17, welchen die Vorrichtung 12(m) während der Kommunikationssitzung verwendet.

Wenn die Vorrichtung 12(m) Nachrichtenpakete zur Übertragung erzeugt, fügt sie ihre Internetadresse (obiger Punkt (a)) als die Quellenadresse ein. Die Vorrichtung(en) 13, welche die jeweiligen Nachrichtenpakete empfängt/empfangen, kann/können die Quellenadresse aus den Nachrichtenpaketen, welche von der Vorrichtung 12(m) empfangen werden, in Nachrichtenpaketen verwenden, welche die Vorrichtung(en) 13 zur Übertragung an die Vorrichtung 12(m) erzeugt/erzeugen, so daß das Internet in der Lage ist, die durch die jeweilige Vorrichtung 13 erzeugten Nachrichtenpakete an die Vorrichtung 12(m) zu leiten. Falls die Vorrichtung 12(m) auf den Namen-Server 17 über das Internet 14 zugreift, hat die durch den ISP 11 bereitgestellte Identifizierung des Namen-Servers 17 (siehe oben unter (b)) die Form einer Zahlen-Internetadresse, welche es der Vorrichtung 12(m) ermöglicht, für den Namen-Server 17 Nachrichten zu erzeugen, welche eine Auflösung der Klartext-Internetadressen in Zahlen-Internetadressen anfordern. Der ISP 11 kann der Vorrichtung 12(m) auch andere IP-Parameter zuordnen, wenn diese sich beim ISP 11 einloggt, beispielsweise die Identifizierung einer Verbindung zu dem Internet 14, welche für Nachrichten zu verwenden ist, die durch die Vorrichtung 12(m) übersandt werden, insbesondere falls der ISP 11 Mehrfach-Gateways aufweist. In der Regel speichert die Vorrichtung 12(m) die Internetparameter im Internetparameterspeicher 25 für die Verwendung während der Kommunikationssitzung.

Wenn ein Bediener die Vorrichtung 12(m) veranlassen möchte, daß sie ein Nachrichtenpaket an eine Vorrichtung 13 überträgt gibt der oder die Bediener(in) die Internetadresse der Vorrichtung 13 an die Vorrichtung 12(m) über die Bedienerschnittstelle 20 ein, sowie eine Information oder die Identifizierung der in der Vorrichtung 12(m) aufbewahrten Information, welche in der Nachricht überragen werden sollen. Die Bedienerschnittstelle 20 aktiviert daraufhin den Paketgenerator 22 zur Freigabe der benötigten Pakete zur Übertragung durch den ISP 11 über das Internet 14. Falls

(i) der Bediener die Zahlen-Internetadresse bereitgestellt hat, oder
(ii) der Bediener die Klartext-Internetadresse bereitgestellt hat, aber der Paketgenerator 22 bereits die Zahlen-Internetadresse besitzt, welche der durch den Bediener eingegebenen Klartext-Internetadresse entspricht,

kann der Paketgenerator 22 unmittelbar nach Aktivierung durch die Bedienerschnittstelle 20 die Pakete erzeugen und diese an die Netzwerkschnittstelle 21 zur Übertragung an den ISP 11 liefern.

Falls aber der Bediener die Klartext-Internetadresse der Vorrichtung 13, an welche die Pakete zu übertragen sind, eingegeben hat, und falls der Paketgenerator 22 die entsprechende Zahlen-Internetadresse davon nicht bereits besitzt,

ermöglicht es der Paketgenerator 22, daß die Netzwerkadresse von dem Namen-Server 17, der in dem IP-Parameterspeicher 25 identifiziert ist, erhalten wird.

Bei diesem Vorgang wird der Paketgenerator 22 anfänglich den Namen-Server 17 kontaktieren, um zu versuchen, die geeignete Zahlen-Internetadresse von dem Namen-Server 17 zu erhalten. Bei diesem Vorgang wird die Vorrichtung 12(m) geeignete Nachrichtenpakete zur Übertragung an den Namen-Server 17 unter Verwendung der Zahlen-Internetadresse des Namen-Servers 17 erzeugen, welche durch den ISP 11 bereitgestellt wird, wenn sich die Vorrichtung 12(m) zu Beginn der Kommunikationssitzung einloggt. Jedenfalls wenn der Namen-Server 17 die Zahlen-Internetadresse für den Klartextnamen besitzt oder erhalten kann, wird der Namen-Server 17 die Zahlen-Internetadresse an die Vorrichtung 12(m) übermitteln. Die Zahlen-Internetadresse wird durch den Paketgenerator 22 über die Netzwerkschnittstelle 21 und den Paketempfänger und -prozessor 23 empfangen. Nachdem der Paketgenerator 22 die Zahlen-Internetadresse empfangen hat, kann er die notwendigen Nachrichtenpakete zur Übertragung an die Vorrichtung 13 durch die Netzwerkschnittstelle 21 und den ISP 11 erzeugen.

Wie oben ausgeführt wurde, ist in **Fig. 1** eine der Vorrichtungen 13, welche an das Internet 14 angeschlossen sind, ein virtuelles privates Netzwerk 15, wobei das virtuelle private Netzwerk 15 eine Firewall bzw. ein Firewall-System 30, mehrere als Server 31(s) gekennzeichnete Vorrichtungen und einen Namen-Server 32 aufweist, die durch eine Übertragungsverbindung 33 miteinander verbunden sind. Die Server 31(s), die Firewall 30 und der Namen-Server 32 können als z. B. in einem LAN oder WAN verbundene Vorrichtungen untereinander Information in Form von Nachrichtenpaketen austauschen. Da die Firewall 30 mit dem Internet 14 verbunden ist und darüber Nachrichtenpakete empfangen kann, hat sie auch eine Internetadresse. Zusätzlich haben wenigstens die Server 31(s), welche über das Internet zugänglich sind, auch jeweilige Internetadressen. Dabei dient der Namen-Server 32 der Umwandlung von Klartext-Internetadressen für die Server 31(s) innerhalb des virtuellen privaten Netzwerkes 15 in die jeweiligen Zahlen-Internetadressen.

Im allgemeinen wird das virtuelle private Netzwerke 15 von einem Unternehmen, einem Regierungsamt, einer Organisation oder ähnlichem gehalten, welche möchten, daß die Server 31(s) Zugriff auf andere Vorrichtungen außerhalb des virtuellen privaten Netzwerkes 15 haben und an diese Information über das Internet 14 übertragen können, aber welche ebenfalls möchten, daß der Zugriff an die Server 31(s) durch Vorrichtungen 12(m) und andere externe Vorrichtungen über das Internet 14 in einer kontrollierten Weise begrenzt ist. Die Firewall 30 dient dazu, den Zugriff durch Vorrichtungen außerhalb des virtuellen privaten Netzwerkes 15 auf Server 31(s) innerhalb des virtuellen privaten Netzwerkes 15 zu kontrollieren. Bei diesem Vorgang stellt die Firewall 30 auch die Verbindung zum Internet 14 her und empfängt Nachrichtenpakete darüber zur Übertragung an einen Server 31(s). Falls das Nachrichtenpaket angibt, daß die Quelle des Nachrichtenpaketes einen Zugriff auf einen bestimmten Server 31(s) anfordert, und falls die Quelle für den Zugriff an den Server 31(s) autorisiert ist, sendet die Firewall 30 das Nachrichtenpaket über die Übertragungsverbindung 33 an den Server 31(s). Falls andererseits die Quelle nicht authorisiert ist, auf den Server 31(s) zuzugreifen, wird die Firewall 30 das Nachrichtenpaket nicht an den Server 31(s) übersenden, und kann anstelle ein Antwortnachrichtenpaket an die Quellenvorrichtung übermitteln, welches angibt, daß die Quelle nicht für den Zugriff an den Server 31(s) authorisiert ist. Die Firewall kann ähnlich aufgebaut sein wie die ande-

ren Vorrichtungen 31(s) in dem virtuellen privaten Netzwerk 15, wobei zusätzlich eine oder mehrere Verbindungen mit dem Internet vorhanden sind, welche allgemein durch das Bezugszeichen 43 gekennzeichnet sind.

Kommunikationen zwischen Vorrichtungen außerhalb des virtuellen privaten Netzwerkes 15, z. B. der Vorrichtung 12(m), und einer Vorrichtung, z. B. einem Server 31(s), innerhalb des virtuellen privaten Netzwerkes 15 kann über einen Sicherheitstunnel zwischen der Firewall 30 und der externen Vorrichtung, wie es oben beschrieben ist, erreicht werden, damit die ausgetauschten Information geheim bleiben, während diese über das Internet 14 und durch den ISP 11 übertragen werden. Ein Sicherheitstunnel zwischen der Vorrichtung 12(m) und dem virtuellen privaten Netzwerk 15 ist in Fig. 1 durch logische Verbindungen dargestellt, welche durch die Bezugszeichen 40, 42 und 44 gekennzeichnet sind; es versteht sich, daß die logischen Verbindung 42 eine der logischen Verbindungen 41 zwischen dem ISP 11 und dem Internet 14 und die logische Verbindung 44 eine der logischen Verbindungen 43 zwischen dem Internet 14 und der Firewall 30 umfaßt.

Der Aufbau eines Sicherheitstunnels kann durch eine Vorrichtung 12(m), die extern zu dem virtuellen privaten Netzwerk 15 ist, ausgelöst werden. Bei diesem Vorgang erzeugt die Vorrichtung 12(m) in Reaktion auf eine Aufforderung durch deren Bediener ein Nachrichtenpaket zur Übertragung durch den ISP 11 und das Internet 14 an die Firewall 30, welches den Aufbau eines Sicherheitstunnels zwischen der Vorrichtung 12(m) und der Firewall 30 anfordert. Das Nachrichtenpaket kann an eine bestimmte Zahlen-Internetadresse gerichtet sein, welche der Firewall 30 zugeordnet ist und welche für Sicherheitstunnelaufbauanfragen reserviert ist, und welche ferner der Vorrichtung 12(m) bekannt ist und durch den Namen-Server 17 bereitgestellt wird. Falls die Vorrichtung 12(m) authorisiert ist, auf einen Server 31(s) in dem virtuellen privaten Netzwerk 15 zuzugreifen, nehmen die Vorrichtung 12(m) als Client und die Firewall 30 einen Dialog auf, welcher den Austausch von einem oder mehreren Nachrichtenpaketen über das Internet 14 umfaßt. Während des Dialogs kann die Firewall 30 der Vorrichtung 12(m) die Identifizierung eines Entschlüsselungsalgorithmus und einen zugehörigen Entschlüsselungsschlüssel bereitstellen, welche die Vorrichtung 12(m) beim Entschlüsseln der verschlüsselten Abschnitte der Nachrichtenpakete zu verwenden hat, welche das virtuelle private Netzwerk an die Vorrichtung 12(m) überträgt. Zusätzlich dazu kann die Firewall 30 der Vorrichtung 12(m) auch die Identifizierung eines Verschlüsselungsalgorithmus und einen zugehörigen Verschlüsselungsschlüssel bereitstellen, welche die Vorrichtung 12(m) beim Verschlüsseln der Abschnitte der Nachrichtenpakete zu verwenden hat, welche die Vorrichtung 12(m) an das virtuelle private Netzwerk 15 überträgt und welche verschlüsselt werden sollen. Alternativ dazu kann die Vorrichtung 12(m) die Identifizierung des Verschlüsselungsalgorithmus und des Verschlüsselungsschlüssels, welche die Vorrichtung 12(m) verwenden wird, an die Firewall 30 während des Dialogs liefern. Die Vorrichtung 12(m) kann in ihrem IP-Parameterspeicher 25 Informationen betreffend den Sicherheitstunnel speichern, einschließlich der Information in Verbindung mit der Identifizierung der Firewall 30 und der Identifizierungen der Verschlüsselungs- und Entschlüsselungsalgorithmen und dazugehöriger Schlüssel für Nachrichtenpakete, welche durch den Sicherheitstunnel übertragen werden.

Sodann können die Vorrichtung 12(m) und die Firewall 30 Nachrichtenpakete über den Sicherheitstunnel übertragen. Beim Erzeugen von Nachrichtenpaketen zur Übertragung über den Sicherheitstunnel verwendet die Vorrichtung

12(m) den Sicherheits-Paketprozessor 26, um die Abschnitte der Nachrichtenpakete zu verschlüsseln, welche vor der Übertragung durch die Netzwerkschnittstelle 21 an den ISP 11 zur Übertragung über das Internet 14 an die Firewall 30 verschlüsselt werden sollen, und um die verschlüsselten Abschnitte der Nachrichtenpakete zu entschlüsseln, welche durch die Vorrichtung 12(m) empfangen werden und welche verschlüsselt sind. Insbesondere nachdem der Paketgenerator 22 ein Nachrichtenpaket zur Übertragung an die Firewall 30 über den Sicherheitstunnel erzeugt hat, liefert er das Nachrichtenpaket an den Sicherheits-Paketprozessor 26. Der Sicherheits-Paketprozessor 26 verschlüsselt daraufhin die Abschnitte des Nachrichtenpakets, welche verschlüsselt werden sollen, unter Verwendung des Verschlüsselungsalgorithmus und des Verschlüsselungsschlüssels. Nachdem die Firewall 30 ein Nachrichtenpaket von der Vorrichtung 12(m) über den Sicherheitstunnel empfangen hat, wird sie dieses entschlüsseln und, falls der beabsichtigte Empfänger des Nachrichtenpakets eine andere Vorrichtung, z. B. ein Server 31(s), in dem virtuellen privaten Netzwerk 15 ist, wird die Firewall 30 das Nachrichtenpaket an diese andere Vorrichtung über die Übertragungsverbindung 33 übertragen.

Wenn ein Nachrichtenpaket von einer Vorrichtung, z. B. einem Server 31(s), in dem virtuellen privaten Netzwerk 15 an die Vorrichtung 12(m) über den Sicherheitstunnel übertragen werden soll, empfängt die Firewall 30 ein solches Nachrichtenpaket über die Übertragungsverbindung 33 und verschlüsselt das Nachrichtenpaket zur Übertragung über das Internet 14 an den ISP 11. Der ISP 11 sendet daraufhin das Nachrichtenpaket an die Vorrichtung 12(m), insbesondere an deren Netzwerkschnittstelle 21. Die Netzwerkschnittstelle 21 liefert das Nachrichtenpaket an den Sicherheits-Paketprozessor 26, welcher die verschlüsselten Abschnitte des Nachrichtenpakets unter Verwendung des Entschlüsselungsalgorithmus und -schlüssels entschlüsselt.

Ein Problem tritt auf im Zusammenhang mit Zugriffen durch eine Vorrichtung, z. B. einer Vorrichtung 12(m), welche extern zum virtuellen privaten Netzwerk 15 ist, und einer Vorrichtung, z. B. einem Server 31(s), welche extern zu der Firewall ist, nämlich dann, wenn dem Namen-Server 17 keine Zahlen-Internetadressen für die Server 31(s) und andere Vorrichtungen bereitgestellt sind, die sich innerhalb des virtuellen privaten Netzwerkes 15 befinden – mit Ausnahme der Zahlen-Internetadressen, welche der Firewall 30 zugeordnet sind. Folglich wird die Vorrichtung 12(m) nach Eingabe der Klartext-Internetadresse durch den Bediener nicht in der Lage sein, die Zahlen-Internetadresse des Servers 31(s) zu erhalten, wenn er auf den Namen-Server 17 zugreift.

Wenn die Vorrichtung 12(m) und die Firewall 39 zusammenarbeiten, um einen dazwischenliegenden Sicherheitstunnel aufzubauen, liefert die Firewall 30 zur Behebung des obigen Problems an die Vorrichtung 12(m) zusätzlich zu möglichen Identifikationen der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüsseln, welche im Zusammenhang mit der Übertragung der Nachrichtenpakete über den Sicherheitstunnel zu verwenden sind, an die Vorrichtung 12(m) auch die Identifizierung eines Namen-Servers, z. B. eines Namen-Servers 32, innerhalb des virtuellen privaten Netzwerkes 15, auf welchen die Vorrichtung 12(m) zugreifen kann, um die geeigneten Zahlen-Internetadressen für die Klartext-Internetadressen zu erhalten, welche den Bediener einer Vorrichtung 12(m) eingegeben werden. Die Identifizierung des Namen-Servers 32 wird ebenfalls in dem IP-Parameterspeicher 25 gespeichert, zusammen mit der Identifizierung des Namen-Servers 17, welche durch den ISP 11 bereitgestellt wurde, sobald die Vorrichtung 12(m)

beim ISP 11 zu Beginn einer Kommunikationssitzung eingeloggt wurde. Wenn daher die Vorrichtung 12(m) ein Nachrichtenpaket an eine Vorrichtung, z. B. einen Server 31(s), in dem virtuellen privaten Netzwerk 15 unter Verwendung einer Klartext-Internetadresse übertragen möchte, welche z. B. durch einen Bediener bereitgestellt bzw. eingegeben wurde, greift die Vorrichtung 12(m) zu Beginn auf den Namen-Server 17 zu, wie es oben beschrieben wurde, um zu versuchen, die zu der Klartext-Internetadresse zugehörige Zahlen-Internetadresse zu erhalten. Da der Namen-Server 17 außerhalb des virtuellen privaten Netzwerkes 15 ist und die durch die Vorrichtung 12(m) angeforderten Information nicht besitzt, sendet er ein entsprechend lautendes Antwortnachrichtenpaket. Die Vorrichtung 12(m) wird sodann ein Anfragenachrichtenpaket zur Übertragung an den Namen-Server 32 durch die Firewall 30 und über den Sicherheitstunnel erzeugen. Falls der Namen-Server 32 eine Zahlen-Internetadresse besitzt, welche zu der Klartext-Internetadresse in dem Anfragenachrichtenpaket gehört, welches durch die Vorrichtung 12(m) geliefert wird, stellt er die Zahlen-Internetadresse in einer Weise bereit, welche im allgemeinen derjenigen ähnlich ist, welche oben im Zusammenhang mit dem Namen-Server 17 beschrieben wurde mit der Ausnahme, daß die Zahlen-Internetadresse durch den Namen-Server 32 in einem an die Firewall 30 gerichteten Nachrichtenpaket geliefert wird, und die Firewall 30 sodann das Nachrichtenpaket über den Sicherheitstunnel an die Vorrichtung 12(m) übermittelt. Es versteht sich, daß sich in dem Nachrichtenpaket, welches durch die Firewall 30 übertragen wird, die Zahlen-Internetadresse in dem Nachrichtenpaket im Datenabschnitt des Nachrichtenpakets befindet, welches über den Sicherheitstunnel übertragen wird und entsprechend verschlüsselt sein wird. Das Nachrichtenpaket wird durch die Vorrichtung 12(m) in einer ähnlichen Weise verarbeitet, wie sie oben im Zusammenhang mit anderen Nachrichtenpaketen beschrieben wurde, welche durch die Vorrichtung 12(m) über den Sicherheitstunnel empfangen werden. Das heißt, daß das Nachrichtenpaket durch den Sicherheits-Paketprozessor 26 vor dem Übermitteln an den Paketempfänger und -prozessor 23 zur Verarbeitung entschlüsselt wird. Die Zahlen-Internetadresse für den Server 31(s) kann in einem Cache in einer Zugriffskontrolliste (ACL) in dem IP-Parameterspeicher 25 gespeichert werden, zusammen mit der Zuordnungsinformation bezüglich der zugehörigen Klartext-Internetadresse, einer Angabe, daß der Server 31(s), der dieser Klartext-Internetadresse zugeordnet ist, über die Firewall 30 des virtuellen privaten Netzwerkes 15 zugänglich ist, und die Identifizierungen der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel, welche für eine Verschlüsselung und Entschlüsselung der geeigneten Abschnitte der Nachrichtenpakete zu verwenden sind, welche an den Server 31(s) übertragen und von diesem erhalten werden.

Es versteht sich, daß in Reaktion auf ein Nachrichtenpaket von der Vorrichtung 12(m), welches beim Namen-Server 32 die Bereitstellung einer Zahlen-Internetadresse für eine durch die Vorrichtung 12(m) angegebene Klartext-Internetadresse anfordert, falls der Namen-Server 32 keine Zuordnungsinformation zwischen der Klartext-Internetadresse und einer Zahlen-Internetadresse besitzt, der Namen-Server 32 ein Antwortnachrichtenpaket, das entsprechend lautet, übertragen kann. Falls die Vorrichtung 12(m) eine Identifizierung von anderen Namen-Servern besitzt, welche z. B. mit anderen virtuellen privaten Netzwerken (nicht gezeigt) verbunden sein können und zu welchen die Vorrichtung 12(m) Zugriff hat, dann kann die Vorrichtung 12(m) versuchen, auf die anderen Namen-Server in einer ähnlichen Weise, wie es oben beschrieben ist, zuzugreifen. Falls die

Vorrichtung 12(m) nicht in der Lage ist, eine Zahlen-Internetadresse, welche der Klartext-Internetadresse zugeordnet ist, von irgendeinem der Namen-Server zu erhalten, zu welchem sie Zugriff hat und welche im allgemeinen im IP-Parameterspeicher 25 der Vorrichtung 12(m) identifiziert sind, wird sie allgemein nicht in der Lage sein, auf eine Vorrichtung mit der vorgegebenen Klartext-Internetadresse zuzugreifen und wird den Bediener oder ein Programm, welche den Zugriff angefordert haben, dementsprechend unterrichten.

Mit diesem Hintergrund werden nun Operationen, welche durch die Vorrichtung 12(m) und das virtuelle private Netzwerk 15 in Verbindung mit der vorliegenden Erfindung durchgeführt werden, im Detail beschrieben. Im allgemeinen laufen die Operationen in zwei Phasen ab. In einer ersten Phase arbeiten die Vorrichtung 12(m) und das virtuelle private Netzwerk 15 zusammen, um einen Sicherheitstunnel durch das Internet 14 aufzubauen. In dieser ersten Phase liefert das virtuelle private Netzwerk 15, insbesondere die Firewall 30, die Identifizierung eines Namen-Servers 32, und es kann auch die den Verschlüsselungs- und Entschlüsselungsalgorithmus und -schlüssel betreffende Information bereitstellen, wie es oben beschrieben wurde. In der zweiten Phase, nachdem der Sicherheitstunnel eingerichtet wurde, kann die Vorrichtung 12(m) die während der ersten Phase gelieferten Information im Zusammenhang mit der Erzeugung und Übertragung von Nachrichtenpaketen an einen oder mehrere Server 31(s) in dem virtuellen privaten Netzwerk 15 und bei dem notwendigen Umwandlungsvorgang der Klartext-Internetadressen zu Zahlen-Internetadressen aus dem Namen-Server 32, welcher durch die Firewall 30 während der ersten Phase identifiziert wurde, verwenden.

Folglich erzeugt die Vorrichtung 12(m) in der ersten (Sicherheitstunnelaufbau)phase zu Beginn ein Nachrichtenpaket zur Übertragung an die Firewall 30, welches einen Aufbau eines Sicherheitstunnels anfordert. Das Nachrichtenpaket enthält eine Zahlen-Internetadresse für die Firewall, (welche durch den Bediener der Vorrichtung oder ein Programm bereitgestellt werden kann, welches durch die Vorrichtung 12(m) verarbeitet wird, oder durch den Namen-Server 17 bereitgestellt werden kann, nachdem eine Klartext-Internetadresse durch den Bediener oder ein Programm bereitgestellt wurde), und welche insbesondere dazu dient, die Firewall 30 zu veranlassen, mit der Vorrichtung 12(m) einen Sicherheitstunnel aufzubauen. Falls die Firewall 30 die Anfrage bezüglich des Sicherheitstunnelaufbaus akzeptiert und falls die Firewall 30 die Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel bereitstellt, so wie es oben angegeben wurde, erzeugt die Firewall 30 ein Antwortnachrichtenpaket zur Übertragung an die Vorrichtung 12(m), welches die Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel identifiziert. Wie oben beschrieben, wird dieses Antwortnachrichtenpaket nicht verschlüsselt. Wenn die Vorrichtung 12(m) die Antwort empfängt, werden die Identifizierungen der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel in dem IP-Parameterspeicher 25 gespeichert.

Zu einem späteren Zeitpunkt in der ersten Phase erzeugt die Firewall 30 auch ein Nachrichtenpaket zur Übertragung an die Vorrichtung 12(m), welches die Zahlen-Internetadresse des Namen-Servers 32 enthält. Bei diesem Nachrichtenpaket wird der Abschnitt des Nachrichtenpakets, welcher die Zahlen-Internetadresse des Namen-Servers 32 enthält, unter Verwendung eines Verschlüsselungsalgorithmus und Verschlüsselungsschlüssels verschlüsselt, und dies kann unter Verwendung des Entschlüsselungsalgorithmus und -schlüssels, die durch das zuvor beschriebene Antwortnachrichtenpaket geliefert wurden, wieder entschlüsselt

werden. Diese Nachricht hat im allgemeinen die folgende Struktur:

"<IIA(FW),IIA(DEV12(m)>><SEC_TUN>
<ENCR<<IIA(FW),IIA(DEV_12(m)>><(DNS_ADRS:IIA(-NS_2>>>"

wobei

    (i) "IIA(FW)" die Quellenadresse darstellt, d. h. eine Zahlen-Internetadresse der Firewall 30,
    (ii) "IIA(DEV_12(m))" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung 12 (m),
    (iii) "DNS_ADRS:IIA(NS)" angibt, daß "IIA(NS_32)" die Zahlen-Internetadresse des Namen-Servers 32 darstellt, für dessen Benutzung die Vorrichtung 12(m) authorisiert ist, und
    (iv) "ENCR<. . . .>" bedeutet, daß die Information, zwischen den Klammern "<" und ">" verschlüsselt ist.

Der Anfangsabschnitt der Nachricht "IIA(FW),IIA(DEV_12(m))>" bildet wenigstens einen Teil des Kopfabschnitts der Nachricht, und "<ENCR<<IIA(FW),IIA(DEV_12(m))><IIA(NS>>>" stellt wenigstens einen Teil des Datenabschnitts der Nachricht dar. "<SEC_TUN>" stellt einen Hinweis in dem Kopfabschnitt dar, welcher angibt, daß die Nachricht über den Sicherheitstunnel übertragen wird, wodurch auch angezeigt wird, daß der Datenabschnitt der Nachricht verschlüsselte Information enthält.

Nachdem die Vorrichtung 12(m) die Nachricht von der Firewall 30 empfängt, wie es oben beschrieben wurde, und weil das Nachrichtenpaket den <SEC_TUN> Hinweis enthält, überträgt deren Netzwerkschnittstelle 21 den verschlüsselten Abschnitt "<ENCR<<IIA(FW),IIA(DEV_12(m)><DNS_ADRS:IIA(-NS_32)>>>" an den Sicherheits-Paketprozessor 26 zur Verarbeitung. Der Sicherheits-Paketprozessor 26 entschlüsselt den verschlüsselten Abschnitt, bestimmt weiter, daß der Abschnitt "IIA(NS_32)" die Zahlen-Internetadresse des Namen-Servers darstellt, insbesondere des Namen-Servers 32, für dessen Benutzung die Vorrichtung 12(m) authorisiert ist, und speichert diese Adresse in dem IP-Parameterspeicher 25 zusammen mit einer Angabe, daß die dorthin gerichteten Nachrichtenpakete zu der Firewall 30 zu übertragen sind, und daß die Daten in den Nachrichtenpaketen unter Verwendung des Verschlüsselungsalgorithmus und -schlüssels, die davor durch die Firewall 30 übermittelt wurden, zu verschlüsseln sind. Es versteht sich, daß aufgrund der Tatsache, daß die Zahlen-Internetadresse des Namen-Servers 32 von der Firewall an die Vorrichtung 12(m) in verschlüsselter Form übertragen wird, diese vertraulich bleibt, selbst wenn das Paket durch einen Dritten abgefangen wird.

In Abhängigkeit des speziellen Protokolls, welches für den Aufbau des Sicherheitstunnels verwendet wird, können die Firewall 30 und die Vorrichtung 12(m) auch Nachrichtenpakete austauschen, welche andere Information enthalten als die oben beschriebenen.

Wie oben erwähnt wurde, kann die Vorrichtung 12(m) in der zweiten Phase nach der Einrichtung des Sicherheitstunnels die Information, welche während der ersten Phase bereitgestellt wurde, im Zusammenhang mit dem Erzeugen und Übertragen von Nachrichtenpaketen zu einem oder mehreren der Server 31(s) in dem virtuellen privaten Netzwerk 15 nutzen. Falls bei diesen Operationen der Bediener einer Vorrichtung 12(m) oder ein Programm, welches durch eine Vorrichtung 12(m) verarbeitet wird, möchte, daß die Vorrichtung 12(m) ein Nachrichtenpaket an einen Server

31(s) in dem virtuellen privaten Netzwerk 15 überträgt, und falls der Bediener durch die Bedienerschnittstelle 20 oder das Programm eine Klartext-Internetadresse bereitstellt, wird zunächst die Vorrichtung 12(m), insbesondere der Paketgenerator 22, bestimmen, ob der IP-Parameterspeicher 25 dort in einem Cache eine Zahlen-Internetadresse gespeichert hat, welche zu der Klartext-Internetadresse gehört. Falls dies nicht der Fall ist, erzeugt der Paketgenerator 22 ein Anfragenachrichtenpaket zur Übertragung an den Namen-Server 17, um von diesem die zu der Klartext-Internetadresse gehörige Zahlen-Internetadresse anzufordern. Falls der Namen-Server 17 eine zu der Klartext-Internetadresse gehörige Zahlen-Internetadresse besitzt, wird dieser die Zahlen-Internetadrese an die Vorrichtung 12(m) liefern. Es versteht sich, daß dies nur erfolgen kann, wenn die Klartext-Internetadresse im Anfragenachrichtenpaket sowohl einer Vorrichtung 13 außerhalb des virtuellen privaten Netzwerkes 15 als auch einem Server 32(s) in dem virtuellen privaten Netzwerk 15 zugeordnet wurde. Danach kann die Vorrichtung 12(m) die Zahlen-Internetadresse verwenden, um Nachrichtenpakete zur Übertragung über das Internet zu erzeugen, wie es oben beschrieben wurde.

Falls andererseits angenommen wird, daß der Namen-Server 17 keine der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse besitzt, wird der Namen-Server 17 ein entsprechend lautendes Antwortnachrichtenpaket an die Vorrichtung 12(m) übermitteln. Sodann erzeugt der Paketgenerator 22 der Vorrichtung 12(m) ein Anfragenachrichtenpaket zur Übertragung an den nächsten Namen-Server, der in ihrem IP-Parameterspeicher 25 identifiziert ist, um von diesem Namen-Server die der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse anzufordern. Falls dieser nächste Namen-Server der Namen-Server 32 ist, liefert der Paketgenerator 22 das Nachrichtenpaket an den Sicherheits-Paketprozessor 26 zur weiteren Verarbeitung. Der Sicherheits-Paketprozessor 26 erzeugt daraufhin ein Anfragenachrichtenpaket zur Übertragung über den Sicherheitstunnel an die Firewall 30. Diese Nachricht hat im allgemeinen folgende Struktur:

"<IIA(DEV_12(m)),IIA(FW)><SEC_TUN>
<ENCR<<IIA(DEV_12(m)),IIA(NS_32))><IIA_REQ>>>-"

wobei

    (i) "IIA(DEV_12(m))" die Quellenadresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung 12(m),
    (ii) "IIA(FW)" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Firewall 30,
    (iii) "IIA(NS_32)" die Adresse des Namen-Servers 32 darstellt,
    (iv) "<<IIA(DEV_12(m)),IIA(NS_32))><IIA_REQ>->>" das Anfragenachrichtenpaket darstellt, welches durch den Paketgenerator 22 erzeugt wird, wobei "<IIA(DEV_12(m)),IIA(NS_32)>" den Kopfabschnitt des Anfragenachrichtenpakets und "<IIA_REQ>" den Datenabschnitt des Anfragenachrichtenpakets darstellt,
    (v) "ENCR<. . . .>" angibt, daß die Information zwischen den Klammern "<" und ">" verschlüsselt ist, und
    (vi) "<SEC_TUN>" einen Hinweis in dem Kopfabschnitt des Nachrichtenpakets darstellt, welches durch den Sicherheitspaketgenerator 26 erzeugt wird und angibt, daß die Nachricht über den Sicherheitstunnel übertragen wird, wobei hierdurch angegeben wird, daß der Datenabschnitt der Nachricht verschlüsselte Information enthält.

Wenn die Firewall **30** das durch den Sicherheitspaketgenerator **26** erzeugte Anfragenachrichtenpaket empfängt, wird diese den verschlüsselten Abschnitt des Nachrichtenpakets entschlüsseln, um "<<IIA(DEV_12(m)),IIA(NS_32))><IIA_REQ>>" zu erhalten. Dies stellt das Anfragenachrichtenpaket dar, welches durch den Paketgenerator **22** erzeugt wird. Nachdem das Anfragenachrichtenpaket erhalten wurde, überträgt die Firewall **30** dieses über die Übertragungsverbindung **33** an den Namen-Server **32**. In Abhängigkeit von dem Protokoll zur Übertragung von Nachrichtenpaketen über die Übertragungsverbindung **33** kann es bei diesem Prozeß für die Firewall **30** notwendig sein, das Anfragenachrichtenpaket zu modifizieren, damit es dem Protokoll der Übertragungsverbindung **33** entspricht.

Nachdem der Namen-Server **32** das Anfragenachrichtenpaket erhalten hat, wird dieser das Anfragenachrichtenpaket verarbeiten, um zu bestimmen, ob er eine der Klartext-Internetadresse, welche in dem Anfragenachrichtenpaket gesendet wird, zugeordnete Zahlen-Internetadresse besitzt. Falls der Namen-Server feststellt, daß er eine solche Zahlen-Internetadresse aufweist, wird dieser ein Antwortnachrichtenpaket zur Übertragung an die Firewall erzeugen, welches die Zahlen-Internetadresse enthält. Im allgemeinen hat das Antwortnachrichtenpaket die folgende Struktur:

"<<IIA(NS_32),IIA(DEV_12(m))><IIA_RESP>>"

wobei

    (i) "IIA(NS_32)" die Quellenadresse darstellt, d. h. die Zahlen-Internetadresse des Namen-Servers **32**,
    (ii) "IIA(DEV_12(m))" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung **12(m)**, und
    (iii) "IIA_RESP" die Zahlen-Internetadresse darstellt, welche der Klartext-Internetadresse zugeordnet ist.

Nachdem die Firewall **30** das Antwortnachrichtenpaket empfangen hat, und weil die Kommunikation mit der Vorrichtung **12(m)** über den dazwischenliegenden Sicherheitstunnel stattfindet, verschlüsselt die Firewall **30** das von dem Namen-Server **32** empfangene Antwortnachrichtenpaket und erzeugt ein Nachrichtenpaket zur Übertragung an die Vorrichtung **12(m)**, welches das verschlüsselte Antwortnachrichtenpaket enthält. Im allgemeinen hat das durch die Firewall **30** erzeugte Nachrichtenpaket die folgende Struktur:

"<IIA(FW),IIA(DEV12(m))><SEC_TUN)>
<ENCR<<IIA(NS_32),IIA(DEV_12(m))><IIA_RESP>>->"

wobei

    (i) "IIA(FW)" die Quellenadresse darstellt, d. h. die Zahlen-Internetadresse der Firewall **30**,
    (ii) "IIA(DEV_12(m))" die Zieladresse darstellt, d. h. die Zahlen-Internetadresse der Vorrichtung **12(m)**,
    (iii) "SEC_TUN" einen Hinweis in dem Kopfabschnitt des Nachrichtenpakets darstellt, welches durch den Sicherheitspaketgenerator **26** erzeugt wird, und angibt, daß die Nachricht über den Sicherheitstunnel übertragen wird, und wobei auch angegeben wird, daß der Datenabschnitt der Nachricht verschlüsselte Information enthält,
    (iv) "ENCR<...>" angibt, daß die Information zwischen den Klammern "<" und ">" (was dem von dem Namen-Server **32** empfangenen Antwortnachrichten-

paket entspricht) verschlüsselt ist.

Zusätzlich kann es je nach dem Protokoll zur Übertragung von Nachrichtenpaketen über die Übertragungsverbindung **33** für die Firewall **30** notwendig sein, das Nachrichtenpaket zu bearbeiten und/oder zu modifizieren, damit dieses dem Protokoll des Internets **14** entspricht.

Wenn die Vorrichtung **12(m)** das Nachrichtenpaket von der Firewall **30** empfängt, wird das Nachrichtenpaket an den Sicherheits-Paketprozessor **26** geliefert. Der Sicherheitspaketprozessor **26** entschlüsselt daraufhin den verschlüsselten Abschnitt des Nachrichtenpakets, um die der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse zu erhalten und lädt diese Information in den IP-Parameterspeicher **25**. Danach kann die Vorrichtung diese Zahlen-Internetadresse beim Erzeugen von Nachrichtenpaketen zur Übertragung an den Server **31(s)** verwenden, welcher zu der Klartext-Internetadresse gehört.

Es versteht sich, daß, falls der Namen-Server **32** keine Zahlen-Internetadresse besitzt, welche der durch die Vorrichtung **12(m)** in dem Anfragenachrichtenpaket gelieferte Klartext-Internetadresse zugeordnet ist, dies der Namen-Server **32** in dem durch ihn erzeugten Antwortnachrichtenpaket entsprechend anzeigen. Die Firewall **30** erzeugt dann in Reaktion auf das durch den Namen-Server **32** gelieferte Antwortnachrichtenpaket auch ein Nachrichtenpaket zur Übertragung an die Vorrichtung **12(m)**, welches einen verschlüsselten Abschnitt enthält, der das Antwortnachrichtenpaket umfaßt, das durch den Namen-Server **32** erzeugt wurde. Nachdem die Vorrichtung **12(m)** das Nachrichtenpaket empfangen hat, wird der verschlüsselte Abschnitt durch den Sicherheitspaketprozessor **26** entschlüsselt, welcher daraufhin den Paketgenerator **22** darüber informiert, daß der Namen-Server **32** keine der Klartext-Internetadresse zugeordnete Zahlen-Internetadresse besitzt. Falls der IP-Parameterspeicher **25** die Identifizierung eines anderen Namen-Servers enthält, erzeugt sodann der Paketgenerator **22** der Vorrichtung **12(m)** ein Anfragenachrichtenpaket zur Übertragung an den nächsten Namen-Server, der in deren IP-Parameterspeicher **25** identifiziert ist, um von diesem Namen-Server die Zahlen-Internetadresse anzufordern, welche der Klartext-Internetadresse zugeordnet ist. Falls andererseits der IP-Parameterspeicher **25** keine Identifizierung eines anderen Namen-Servers enthält, kann der Paketgenerator **22** die Bedienerschnittstelle **20** oder ein Programm darüber informieren, daß er nicht in der Lage ist, ein Nachrichtenpaket zur Übertragung an eine Vorrichtung zu erzeugen, welche der Klartext-Internetadresse zugeordnet ist, welche durch die Bedienerschnittstelle **20** oder ein Programm eingegeben bzw. bereitgestellt wurde.

Die Erfindung liefert eine Anzahl von Vorteilen. Insbesondere schafft die Erfindung ein System zum Vereinfachen der Kommunikation zwischen Vorrichtungen, welche mit einem öffentlichen Netzwerk verbunden sind, z. B. mit dem Internet **14**, und Vorrichtungen, welche mit privaten Netzwerken verbunden sind, z. B. mit dem virtuellen privaten Netzwerk **15**, indem die Umwandlung von Klartextadressen in Netzwerkadressen durch einen Namen-Server, der bevorzugt über einen Sicherheitstunnel mit den privaten Netzwerken verbunden ist, ermöglicht wird.

Es versteht sich, daß eine Vielzahl von Modifikationen an der im Zusammenhang mit **Fig. 1** beschriebenen Anordnung durchgeführt werden können. Obwohl das Netzwerk **10** so beschrieben wurde, daß die Identifizierung der Verschlüsselungs- und Entschlüsselungsalgorithmen und -schlüssel durch die Vorrichtung **12(m)** und die Firewall **30** während des Dialogs, währenddessen der Sicherheitstunnel eingerichtet wird, ausgetauscht wird, versteht es sich, daß bei-

spielsweise Information durch die Vorrichtung 12(m) und die Firewall 30 getrennt von dem Aufbau eines solchen Sicherheitstunnels bereitgestellt werden können.

Obwohl die Erfindung im Zusammenhang mit dem Internet beschrieben wurde, versteht es sich ferner, daß die Erfindung in Verbindung mit jedem, insbesondere globalen, Netzwerk verwendet werden kann. Obwohl die Erfindung im Zusammenhang mit einem Netzwerk beschrieben wurde, welches ein System von Klartext-Netzwerkadressen bereitstellt, versteht es sich ferner, daß die Erfindung nicht darauf beschränkt ist sondern in Verbindung mit jedem Netzwerk verwendet werden kann, welches irgendeine Form einer – den systemeigenen Netzwerkadressen übergeordnete Sekundär-Netzwerkadresseneinrichtung oder vergleichbare nicht-formeller Netzwerkadresseneinrichtung vorsieht.

Es versteht sich ferner, daß ein erfindungsgemäßes System als ganzes oder in Teilen aus speziell hierfür geeigneter Hardware oder einem allgemein geeigneten Computersystem oder jeder Kombination davon aufgebaut werden kann, wobei jeder Abschnitt davon durch ein geeignetes Programm gesteuert werden kann. Jedes Programm kann als ganzes oder in Teilen einen Teil des Systems umfassen oder auf dem System in einer konventionellen Weise gespeichert sein, oder es kann als ganzes oder in Teilen in das System über ein Netzwerk oder andere Mechanismen zur Übertragung von Information in einer konventionellen Weise bereitgestellt werden. Zusätzlich versteht es sich, daß das System betrieben und/oder auf andere Art und Weise mittels Information gesteuert werden kann, welche durch einen Bediener mittels Bedienereingabeelementen (nicht gezeigt) bereitgestellt wird, welche direkt an das System angeschlossen sein können oder welche die Information über ein Netzwerk oder andere Mechanismen zur Übertragung von Information in einer konventionellen Weise übertragen können.

Die vorstehende Beschreibung hat sich auf ein spezifisches Ausführungsbeispiel der Erfindung bezogen. Es versteht sich jedoch, daß verschiedene Variationen und Modifikationen der Erfindung gemacht werden können, bei welchen einige oder alle der Vorteile der Erfindung erreicht werden. Diese und andere Variationen und Modifikationen fallen in den Schutzbereich der vorliegenden Erfindung, der durch die nachfolgenden Ansprüche bestimmt ist.

Patentansprüche

1. System umfassend ein virtuelles privates Netzwerk (15) und eine externe Vorrichtung (12 (m)), welche über ein digitales Netzwerk (14) kommunizieren, wobei:

das virtuelle private Netzwerk (15) eine Firewall (30), wenigstens eine interne Vorrichtung (31(s)) und einen Namen-Server (32) aufweist, welche jeweils eine Netzwerkadresse besitzen, wobei die interne Vorrichtung (31(s)) auch eine Sekundäradresse besitzt und der Namen-Server (32) derart konfiguriert ist, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt,

die Firewall (30) derart konfiguriert ist, daß sie der externen Vorrichtung (12(m)) in Reaktion auf deren Anfrage zum Aufbau einer Verbindung zur Firewall (30) die Netzwerkadresse des Namen-Servers (32) liefert, und

die externe Vorrichtung (12(m)) derart konfiguriert ist, daß sie in Reaktion auf eine Anfrage zum Zugriff auf die interne Vorrichtung (31(s)), welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält, eine Netzwerkadressen-Anfragenachricht zur Übertragung über die Verbindung an die Firewall (30) erzeugt, wel-

che eine Auflösung der der Sekundäradresse zugeordneten Netzwerkadresse anfordert, wobei die Firewall (30) derart konfiguriert ist, daß sie die Adressenauflösungsanfrage an den Namen-Server (32) übermittelt, der Namen-Server (32) derart konfiguriert ist, daß er die der Sekundäradresse zugeordnete Netzwerkadresse bereitstellt, und die Firewall (30) daraufhin die Netzwerkadresse in einer Netzwerkadressen-Antwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung (12(m)) bereitstellt.

2. System nach Anspruch 1, bei welchem die externe Vorrichtung (12(m)) derart konfiguriert ist, daß sie die in der Netzwerkadressen-Antwortnachricht bereitgestellte Netzwerkadresse beim Erzeugen von wenigstens einer Nachricht zur Übertragung an die interne Vorrichtung (31(s)) verwendet.

3. System nach Anspruch 1 oder 2, bei welchem die externe Vorrichtung (12(m)) derart konfiguriert ist, daß sie mit dem Netzwerk (14) durch einen Netzwerk-Service-Provider (11) verbunden wird.

4. System nach Anspruch 3, bei welchem die externe Vorrichtung (12(m)) derart konfiguriert ist, daß sie eine Kommunikationssitzung mit dem Netzwerk-Service-Provider (11) aufbaut, wobei der Netzwerk-Service-Provider (11) der externen Vorrichtung (12(m)) die Identifizierung eines weiteren Namen-Servers übermittelt, wobei der weitere Namen-Server derart konfiguriert ist, daß er eine Zuordnung zwischen einer Sekundäradresse und einer Netzwerkadresse für wenigstens eine Vorrichtung bereitstellt.

5. System nach einem der vorstehenden Ansprüche, bei welchem die externe Vorrichtung (12(m)) derart konfiguriert ist, daß sie eine Liste von Namen-Servern erhält, welche der externen Vorrichtung (12(m)) identifiziert wurden, und die externe Vorrichtung (12(m)) die Namen-Server in der Liste nacheinander in Reaktion auf eine Anfrage zum Zugriff auf eine andere Vorrichtung abfragt, wobei die Anfrage eine Sekundäradresse der anderen Vorrichtung enthält, solange bis die externe Vorrichtung (12(m)) eine Netzwerkadresse empfängt, wobei die externe Vorrichtung (12(m)) in jedem Abfragevorgang eine Netzwerkadressen-Anfragesnachricht zur Übertragung über das Netzwerk (14) erzeugt, welche durch einen der Namen-Server in der Liste zu beantworten ist, und von diesem eine Netzwerkadressen-Antwortnachricht empfängt.

6. System nach einem der vorstehenden Ansprüche, bei welchem die Verbindung zwischen der externen Vorrichtung (12(m)) und der Firewall (30) ein Sicherheitstunnel ist, in welchem wenigstens ein der zwischen der externen Vorrichtung (12(m)) und der Firewall (30) übertragenen Nachrichten verschlüsselt ist.

7. Verfahren zum Betreiben eines Systems umfassend ein virtuelles privates Netzwerk (15) und eine externe Vorrichtung (12(m)), welche durch ein digitales Netzwerk (14) miteinander verbunden sind, wobei das virtuelle private Netzwerk (15) eine Firewall (30), wenigstens eine interne Vorrichtung (31(s)) und einen Namen-Server (32) aufweist, welche jeweils eine Netzwerkadresse besitzen, wobei die interne Vorrichtung (31(s)) auch eine Sekundäradresse besitzt, und der Namen-Server (32) derart konfiguriert ist, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt, wobei:

A. in Reaktion auf eine Anfrage der externen Vorrichtung (12(m)) zum Aufbau einer Verbindung zur Firewall (30) die Firewall (30) der externen Vorrichtung (12(m)) die Netzwerkadresse des

Namen-Servers (32) übermittelt; und

B. (i) in Reaktion auf eine Anfrage zum Zugriff auf die interne Vorrichtung (31(s)), welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält, die externe Vorrichtung (12(m)) eine Netzwerkadressen-Anfragenachricht zur Übertragung über die Verbindung an die Firewall (30) erzeugt, welche eine Auflösung der Netzwerkadresse, welche der Sekundäradresse zugeordnet ist, anfordert,

(ii) die Firewall (30) die Adressenauflösungsanfrage an den Namen-Server (32) übermittelt, (iii) der Namen-Server (32) die der Sekundäradresse zugeordnete Netzwerkadresse bereitstellt, und

(iv) die Firewall (30) die Netzwerkadresse in einer Netzwerkadressen-Antwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung (12(m)) bereitstellt.

8. Verfahren nach Anspruch 7, bei welchem die externe Vorrichtung (12((m) ferner die in der Netzwerkadressen-Antwortnachricht bereitgestellte Netzwerkadresse beim Erzeugen von wenigstens einer Nachricht zur Übertragung an die interne Vorrichtung (31(s)) verwendet.

9. Verfahren nach Anspruch 7 oder 8, bei welchem die externe Vorrichtung (12(m)) mit dem Netzwerk (14) durch einen Netzwerk-Service-Provider (11) verbunden werden kann.

10. Verfahren nach Anspruch 9, bei welchem die externe Vorrichtung (12(m)) eine Kommunikationssitzung mit dem Netzwerk-Service-Provider (11) aufbaut, wobei der Netzwerk-Service-Provider (11) der externen Vorrichtung (12(m)) die Identifizierung eines weiteren Namen-Servers übermittelt, wobei der weitere Namen-Server eine Zuordnung zwischen einer Sekundäradresse und einer Netzwerkadresse für wenigstens eine Vorrichtung bereitstellt.

11. Verfahren nach einem der Ansprüche 7 bis 10, bei welchem die externe Vorrichtung (12(m)) eine Liste von Namen-Servern erhält, welche der externen Vorrichtung (12(m)) identifiziert wurden, und die externe Vorrichtung (12(m)) die Namen-Server in der Liste nacheinander in Reaktion auf eine Anfrage zum Zugriff auf eine andere Vorrichtung abfragt, wobei die Anfrage eine Sekundäradresse der anderen Vorrichtung enthält, solange bis die externe Vorrichtung (12(m)) eine Netzwerkadresse empfängt, wobei die externe Vorrichtung (12(m)) in jedem Abfragevorgang eine Netzwerkadressen-Anfragenachricht zur Übertragung über das Netzwerk (14) erzeugt, welche durch einen der Namen-Server in der Liste zu beantworten ist, und von diesem eine Netzwerkadressen-Antwortnachricht empfängt.

12. Verfahren nach einem der Ansprüche 7 bis 11, bei welchem die Verbindung zwischen der externen Vorrichtung (12(m)) und der Firewall (30) ein Sicherheitstunnel ist, in welchem wenigstens ein Abschnitt der zwischen der externen Vorrichtung (12(m)) und der Firewall (30) übertragenen Nachrichten verschlüsselt ist.

13. Computerprogramm-Produkt zur gemeinsamen Verwendung mit einem virtuellen privaten Netzwerk (15) und einer externen Vorrichtung (12(m)), welche durch ein digitales Netzwerk (14) miteinander verbunden sind, wobei das virtuelle private Netzwerk eine Firewall (30), wenigstens eine interne Vorrichtung (31(s)) und einen Namen-Server (32) aufweist, welche jeweils eine Netzwerkadresse besitzen, wobei die interne Vorrichtung (31(s)) auch eine Sekundäradresse

besitzt, und der Namen-Server (32) derart konfiguriert ist, daß er eine Zuordnung zwischen der Sekundäradresse und der Netzwerkadresse bereitstellt, wobei das Computerprogrammprodukt ein maschinenlesbares Medium mit folgenden Codes aufweist:

A. ein Namen-Server-Identifizierungscodemodul, welches veranlaßt, daß die Firewall (30) der externen Vorrichtung (12(m)) in Reaktion auf deren Anfrage zum Aufbau einer Verbindung zur Firewall (30) die Netzwerkadresse des Namen-Servers (32) übermittelt,

B. ein Codemodul zur Erzeugung einer Netzwerkadressen-Anfragenachricht, welches veranlaßt, daß die externe Vorrichtung (12(m)) in Reaktion auf eine Anfrage zum Zugriff auf die interne Vorrichtung (31(s)), welche die Sekundäradresse der internen Vorrichtung (31(s)) enthält, eine Netzwerkadressen-Anfragenachricht zur Übertragung über die Verbindung an die Firewall (30) erzeugt, welche die Auflösung der der Sekundäradresse zugeordneten Netzwerkadresse anfordert,

C. ein Modul zur Übermittlung einer Adressenauflösungsanfrage, welches veranlaßt, daß die Firewall (30) die Adressenauflösungsanfrage an den Namen-Server (32) übermittelt,

D. ein Namen-Server-Steuerungsmodul, welches veranlaßt, daß der Namen-Server (32) die der Sekundäradresse zugeordnete Netzwerkadresse bereitstellt, und

E. ein Modul zur Übermittlung einer Netzwerkadressen-Antwortnachricht, welches veranlaßt, daß die Firewall (30) die Netzwerkadresse in einer Netzwerkadressen-Antwortnachricht zur Übertragung über die Verbindung an die externe Vorrichtung (12(m)) bereitstellt.

14. Computerprogramm-Produkt nach Anspruch 13, welches ferner ein Netzwerkadressenverwendungsmodul aufweist, welches veranlaßt, daß die externe Vorrichtung (12(m)) die in der Netzwerkadressen-Antwortnachricht übermittelte Netzwerkadresse beim Erzeugen von wenigstens einer Nachricht zur Übertragung an die interne Vorrichtung (31(s)) verwendet.

15. Computerprogramm-Produkt nach Anspruch 13 oder 14, welches ferner ein Netzwerk-Service-Provider-Steuerungsmodul aufweist, welches veranlaßt, daß die externe Vorrichtung (12(m)) mit dem Netzwerk (14) durch einen Netzwerk-Service-Provider (11) verbunden wird.

16. Computerprogramm-Produkt nach Anspruch 15, bei welchem das Netzwerk-Service-Provider-Steuerungsmodul ein Kommunikationssitzungsaufbaumodul umfaßt, welches veranlaßt, daß die externe Vorrichtung (12(m)) mit dem Netzwerk-Service-Provider (11) eine Kommunikationssitzung aufbaut und von diesem eine Identifizierung von einem weiteren Namen-Server empfängt.

17. Computerprogramm-Produkt nach einem der Ansprüche 13 bis 16, welches ferner ein Namen-Server-Abfragesteuerungsmodul aufweist, welches veranlaßt, daß die externe Vorrichtung (12(m)) eine Liste von Namen-Servern erhält, welche der externen Vorrichtung (12(m)) identifiziert wurden, und die Namen-Server in der Liste nacheinander in Reaktion auf eine Anfrage zum Zugriff auf eine andere Vorrichtung abfragt, wobei die Anfrage eine Sekundäradresse der anderen Vorrichtung enthält, solange bis die externe Vorrichtung (12(m)) eine Netzwerkadresse empfängt, und wobei die externe Vorrichtung (12(m)) in jedem Abfragevor-

gang eine Netzwerkadressen-Anfragesnachricht zur Übertragung über das Netzwerk (14) erzeugt, welche durch einen der Namen-Server in der Liste zu beantworten ist, und von diesem eine Netzwerkadressen-Antwortnachricht empfängt.
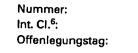
18. Computerprogramm-Produkt nach einem der Ansprüche 13 bis 17, bei welchem die Verbindung zwischen der externen Vorrichtung (12(m)) und der Firewall (30) ein Sicherheitstunnel ist, in welchem wenigstens ein Abschnitt der zwischen der externen Vorrichtung (12(m)) und der Firewall (30) übertragenen Nachrichten verschlüsselt ist.

Hierzu 1 Seite(n) Zeichnungen

NETZWERK 10

SERVER 31(1)

SERVER 31(S)

VPN NAMEN-SERVER 32

33

FIRE-WALL 30

VIRTUELLES PRIVATES NETZWERK 15

AN/VON ZUGRIFFS-VORRICHTUNGEN

AN/VON ZUGRIFFS-VORRICHTUNGEN

43

44

INTERNET 14

AN/VON ANDERE(N) ISP'S

AN/VON ANDERE(N) ISP'S

16

16

42

41

13

F I G. 1

NAMEN-SERVER 17

INTERNET SERVICE PROVIDER 11

40

VORRICHTUNG 12(1)

VORRICHTUNG 12(M)

VORRICHTUNG 12(m)

20

24

25

21

22

23

26

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1	("6725268").PN.	US-PGPUB; USPAT	OR	OFF	2006/06/17 18:45
L2	0	("10714849").PN.	US-PGPUB; USPAT	OR	OFF	2006/06/17 18:45
L3	1	("6502135").PN.	US-PGPUB; USPAT	OR	OFF	2006/06/17 18:59
L4	10	secure near4 domain near5 name near5 service	US-PGPUB; USPAT	OR	ON	2006/06/17 19:02
L5	698	secure same domain same name	US-PGPUB; USPAT	OR	ON	2006/06/17 19:03
L6	93	l5 and portal	US-PGPUB; USPAT	OR	ON	2006/06/17 19:04
L7	63	l6 and (authenticat$ or cryptographic)	US-PGPUB; USPAT	OR	ON	2006/06/17 19:05
L8	63	l7 and (portal or router)	US-PGPUB; USPAT	OR	ON	2006/06/17 19:05
L9	8	l8 and secure near4 network near4 address$	US-PGPUB; USPAT	OR	ON	2006/06/17 19:08
L10	23	l8 and (domain near4 name).ti,ab, clm.	US-PGPUB; USPAT	OR	ON	2006/06/17 19:12
L11	2196	DNS.ti,ab,clm.	US-PGPUB; USPAT	OR	ON	2006/06/17 19:13
L12	14	l11 and portal.ti,ab,clm.	US-PGPUB; USPAT	OR	ON	2006/06/17 19:15
L13	84	l11 and secure same network same address	US-PGPUB; USPAT	OR	ON	2006/06/17 19:16
L14	70	l13 and (portal or router)	US-PGPUB; USPAT	OR	ON	2006/06/17 19:17
L15	69	l14 and internet	US-PGPUB; USPAT	OR	ON	2006/06/17 19:17
L16	51	l15 and (authenticat$ or cryptographic$)	US-PGPUB; USPAT	OR	ON	2006/06/17 19:18

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	185	scom.ti,ab,clm. or sorg.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:11
L2	142	l1 and @ad<="19990607"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:16
L3	0	l2 and sgov.ti,ab,clm. and snet.ti, ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:12
L4	0	l2 and secure.ti,ab,clm. and network.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:13
L5	1	l2 and secure and network	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:13
L6	14	l2 and (secure domain name service)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:18
L7	990019	(secure domain name service).ti,ab, clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:18
L8	89178	l7 and server.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:19

EAST Search History

L9	57641	I8 and (secure network connection).ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:19
L10	7778	I9 and authenticat$.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:20
L11	1809	I10 and (edge router)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:20
L12	1809	I11 and @ad<="199900607"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:21
L13	380	I12 and cryptographic	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:21
L14	380	I13 and server.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:22
L15	380	I14 and (virtual private network)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:27
L16	0	I15 and (".scom" or ".sorg" or ".snet")	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:23

L17	39	l15 and @ad<="19990607"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/09/02 11:27

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	10714849
	Filing Date	2003-11-18
	First Named Inventor	Victor Larson
	Art Unit	2143
	Examiner Name	TBD
	Attorney Docket Number	000479.00111

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	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 Code[1]	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 Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	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.		☐

EFS Web 2.0

<table>
<tr><td></td><td>4</td><td>9827783</td><td>WO</td><td>A</td><td>1998-06-25</td><td>Northern Telecom Limited; Antonio, G; Hui, Margare</td><td></td><td>☐</td></tr>
<tr><td></td><td>5</td><td>2 317 792</td><td>GB</td><td>A</td><td>1998-04-01</td><td>Secure Computing Corporation</td><td></td><td>☐</td></tr>
<tr><td></td><td>6</td><td></td><td></td><td></td><td></td><td></td><td></td><td>☐</td></tr>
</table>

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.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	10714849
	Filing Date	2003-11-18
	First Named Inventor	Victor Larson
	Art Unit	2143
	Examiner Name	TBD
	Attorney Docket Number	000479.00111

[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	10714849	
	Filing Date	2003-11-18	
	First Named Inventor	Victor Larson	
	Art Unit	2143	
	Examiner Name	TBD	
	Attorney Docket Number	000479.00111	

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 to make the appropriate selection(s):

[X] 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

This collection of information is required by 37 CFR 1.97 and 1.98. 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 take 1 hour 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.**

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.

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. Data supplied from the esp@cenet database - Worldwide

----
Courtesy of http://v3.espacenet.com

PCT

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, 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.

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.

- 3 -

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

- 4 -

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

- 5 -

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

- 6 -

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

- 7 -

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.

- 11 -
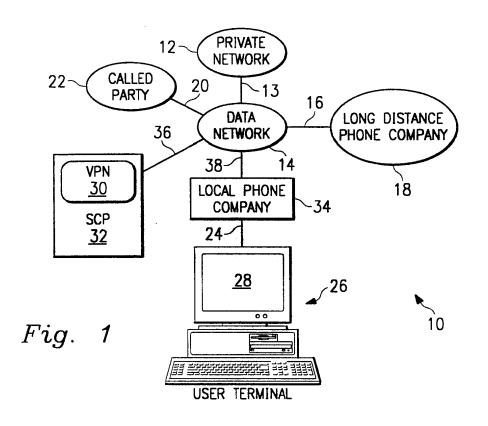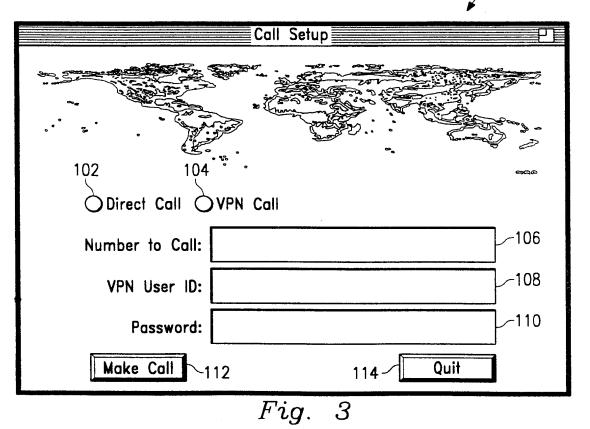
18.   The apparatus of claim 17, further comprising means for encrypting the user's password, and sending the user identification and the encrypted password to the virtual private network service provider.

19.   The apparatus of claim 18, further comprising means for sending a set-up message to the data network.
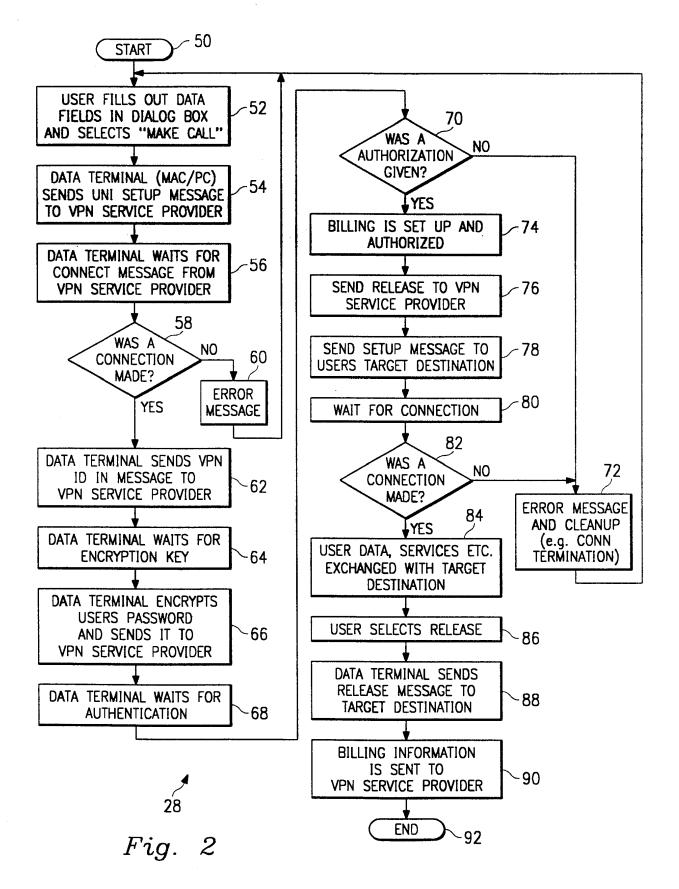
20.   The apparatus of claim 19, further comprising means for decrypting the encrypted password.

Fig. 1

Fig. 3

Fig. 2

Fig. 4A

MACINTOSH   HOTEL ATM INTERFACE   PUBLIC ATM NETWORK   MOE'S VPN SERVICE   TRAVEL ATM INTERFACE   TRAVEL SERVICE

DATA

RELEASE   RELEASE   RELEASE

RELEASE CMP

BILLING RECORD

BILLING ACK

Fig. 4B

300   WAIT FOR EVENT ← SETUP MESSAGE (SIGNALLING PORT)

302   ASSIGN CCB TO CALL BASED ON CALL REFERENCE NUMBER

304   COMPILE CONNECT MESSAGE

Fig. 5A

306   SEND CONNECT MESSAGE TO CALLING ADDRESS (HOTEL)

30

308   CONDENSE

310   WAIT FOR EVENT ← CONNECT ACK MESSAGE (SIGNALLING PORT)

Fig. 5B

312   ACCESS CCB BASED ON CALL REFERENCE NUMBER

30

314   CONDENSE

Fig. 5C

Fig. 5D

348 — WAIT FOR EVENT ← RELEASE MESSAGE (SIGNALLING PORT)

350 — ACCESS CCB BASED ON CALL REFERENCE NUMBER

Fig. 5E

352 — COMPILE RELEASE COMPLETE MESSAGE

~30

354 — SEND RELEASE COMPLETE MESSAGE

356 — CONDENSE

358 — WAIT FOR EVENT ← 3rd PARTY BILLING SETUP MESSAGE (SIGNALLING PORT)

30

360 — VERIFY TOKEN

VALID   362   NOT VALID

368   364

SEND MODIFIED 3rd PARTY BILLING SETUP MESSAGE TO NETWORK

SEND RELEASE MESSAGE

370 — CONDENSE

CONDENSE — 366

Fig. 5F

INTERNATIONAL SEARCH REPORT
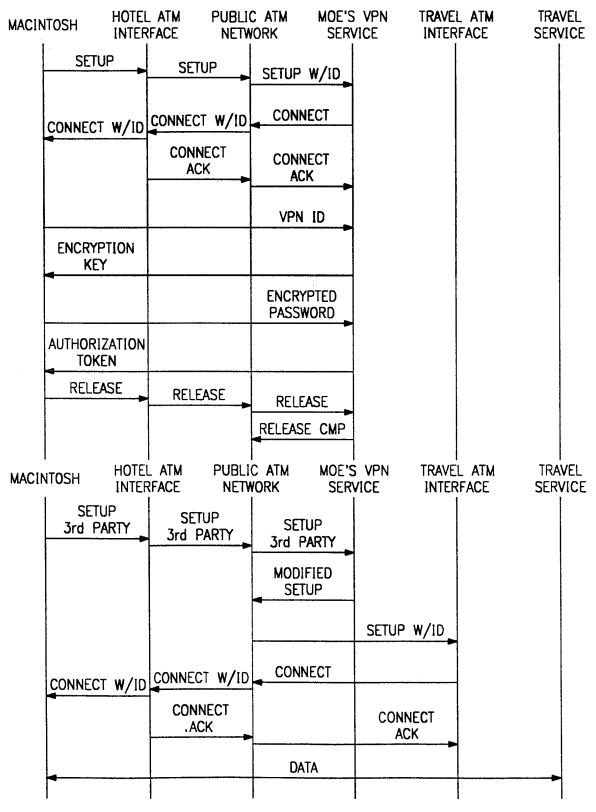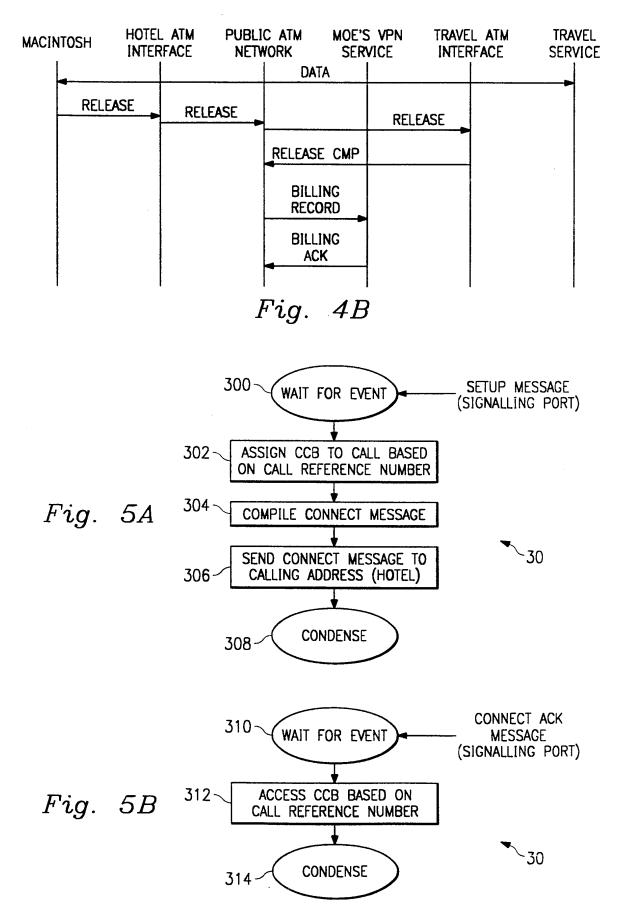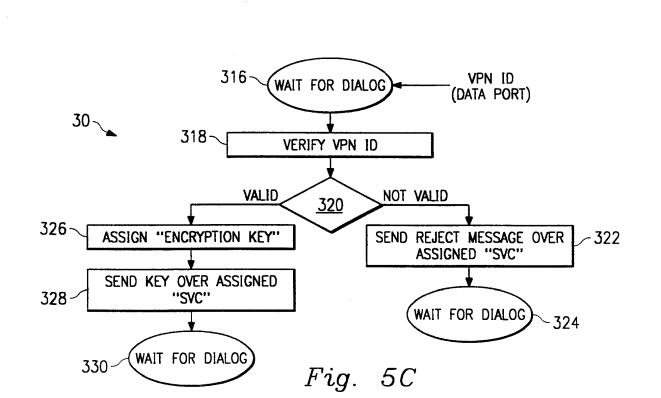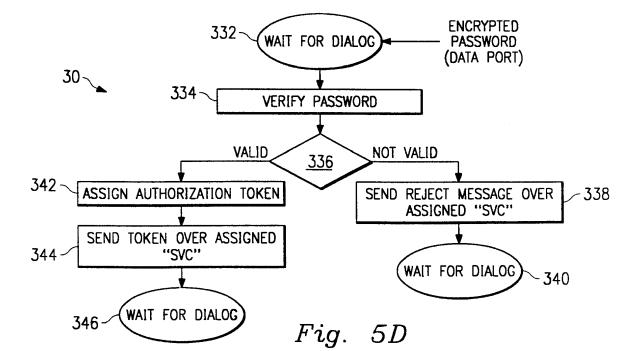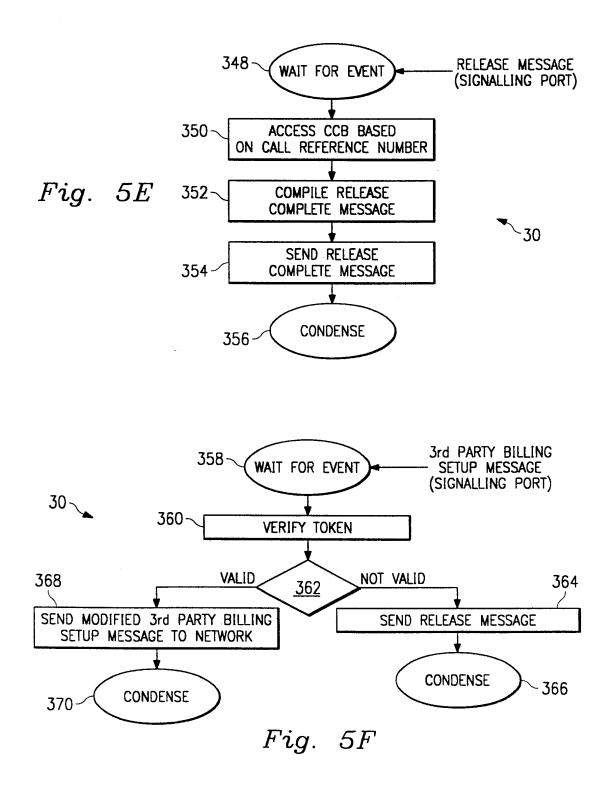
A. CLASSIFICATION OF SUBJECT MATTER
IPC 6   H04Q11/04   H04L12/22

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 docu-
    ments, 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
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)

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

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

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.

Data supplied from the esp@cenet database - Worldwide

Courtesy of http://v3.espacenet.com

(12) UK Patent Application (19) GB (11) 2 334 181 (13) A

(43) Date of A Publication 11.08.1999

(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

B_CH

DOWNLOAD CHANNEL

SIGNALLING CONTROL

Figure: 1

Figure 2

Switch_ON — P_CH — Check if a radio scheme is compliant with the environment

Check if a radio scheme is compliant with the environment → Resource Unavailable

Check if a radio scheme is compliant with the environment → Resource Available — Logical Mapping — Local B_CH — Change Transmission Mode Application ?

Change Transmission Mode Application ? — no → End of Session

Change Transmission Mode Application ? — yes

Communications over the B_CH

Information Exchange — Radio Scheme/ Protocol loaded in terminal memory ?

Radio Scheme/ Protocol loaded in terminal memory ? — no → Selection Script — OTA Download — Set Up

Radio Scheme/ Protocol loaded in terminal memory ? — yes → Set Up

Set Up — Installation Complete — Network Update

3/3

Figure 3

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

yes

Selection
Script

OTA
Download

Set Up

Installation
Complete

Network
Update

Communications over the B_CH

Petitioner RPX Corporation - Ex. 1002, p. 266

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.

Application No: GB 9802545.5	**Examiner:** Glyn Hughes
Claims searched: 1 to 5	**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.

Data supplied from the esp@cenet database - Worldwide

Courtesy of http://v3.espacenet.com

(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

Description

FIELD OF THE INVENTION

5   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

10   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 communica-
15   tions, 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 con-
20   tained 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.
25   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 transac-
30   tions, 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.
35   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.
40   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

45
       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
50   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
55   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 net-

2

work 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 por-

3

tion 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 Soft-

ware 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. Stored Configuration Capabilities

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

6

terminal configuration parameters to be reset (**Step 614**) and the alternate data network to be automatically disconnected (**Step 615**).

D. <u>Off-Line</u> <u>Form</u> <u>Capability</u>

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,

7

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

8

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

9

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.
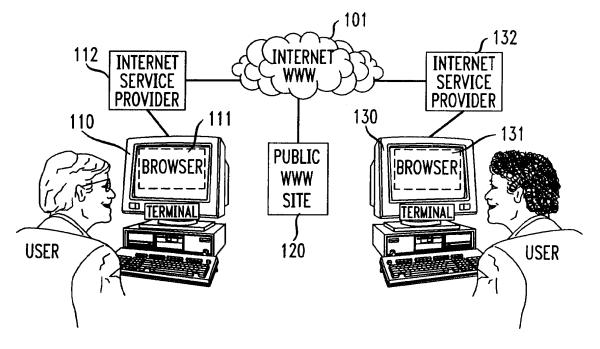
24. The method for managing connections of Claim 23 wherein said recognizing step occurs at a point when said terminal device is connected to a given source/processor.

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.
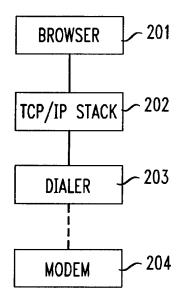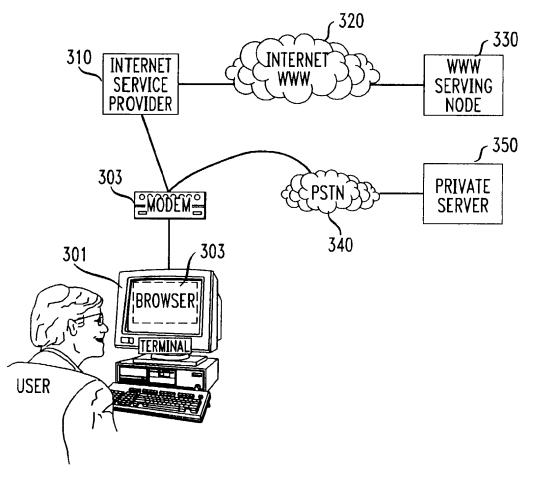
27. The method for managing connections of Claim 26 wherein said selected information items are processed by said user at said terminal device prior to uploading to said other information source/processor.

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

FIG. 2

FIG. 3

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

14

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

Ⓐ

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

18

FIG. 7B FROM FIG.7A
[A]

┌───┐
│ BRIDGING SOFTWARE RECEIVES RECONFIGURATION   │
│ INSTRUCTIONS FOR SELECTED OFF-LINE FORM APPLICATION │ ─ 708
│ FROM ASSOCIATED RE-CONFIGURATION FILE   │
└───┘

┌───┐
│ BRIDGING SOFTWARE EDITS USER-TERMINAL COMMUNICATIONS │
│ SOFTWARE PARAMETERS, RECONFIGURING TERMINAL TO DIAL │ ─ 709
│ DATA NETWORK FOR SELECTED OFF-LINE FORM APPLICATION │
└───┘

┌───┐
│ BRIDGING SOFTWARE AUTOMATICALLY STARTS TERMINAL │
│ BROWSER AND DIALS DATA NETWORK FOR SELECTED   │ ─ 710
│ OFF-LINE FORM APPLICATION   │
└───┘

┌───┐
│ 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   │ ─ 712
│ DATA TO THAT NETWORK   │
└───┘

┌───┐
│ 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 │ ─ 715
│ DATA NETWORK TO BRIDGING SOFTWARE   │
└───┘

┌───┐
│ BRIDGING SOFTWARE EDITS USER-TERMINAL COMMUNICATIONS │
│ SOFTWARE PARAMETER, RECONFIGURING TERMINAL TO DIAL   │ ─ 716
│ ORIGINAL (OR ANOTHER) DATA NETWORK   │
└───┘

┌───┐
│ BRIDGING SOFTWARE AUTOMATICALLY DISCONNECTS CONNECTION │ ─ 717
│ BETWEEN TERMINAL AND OFF-LINE FORM APPLICATION DATA NETWORK │
└───┘

(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.

Data supplied from the esp@cenet database - Worldwide

Courtesy of http://v3.espacenet.com

(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

Petitioner RPX Corporation - Ex. 1002, p. 298

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.
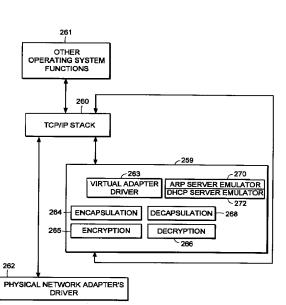
Thus there is disclosed a new pseudo network adapter providing a virtual private network having dynamically determined end points to support users in a mobile computing environment. The new pseudo network adapter provides a system for capturing a fully formed frame prior to transmission. The new pseudo 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 Edi-

tion, 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.

7

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 discon-

nects 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 request 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 applica-

tion. 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.

FIG. 1

FIG. 2

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

20

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

FIG. 9

FIG. 10

FIG. 11

FIG. 12

FIG. 13

FIG. 14

FIG. 15

OPERATING SYSTEM

310

282

TCP/IP STACK | TCP/IP TRANSMIT FUNCTION | TCP/IP RECEIVE FUNCTION

286 TRANSMITTED ETHERNET PACKETS   284 RECEIVED ETHERNET PACKETS   312   314   280

288 VIRTUAL NETWORK ADAPTER (EMULATES AN ETHERNET DEVICE)   PSEUDO NETWORK ADAPTER   302

ARP AND DHCP PACKETS

290 TRANSMIT PATH   RECEIVE PATH   296   304

292 ENCRYPTION   298 DECRYPTION   ARP SERVER EMULATOR

294   300   DHCP SERVER EMULATOR

ENCAPSULATION   DECAPSULATION

306

PHYSICAL NETWORK ADAPTER (SENDS AND RECEIVES PACKETS ON PHYSICAL NETWORK) — 308

FIG. 16

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.

THIS EVENT OCCURS WHEN THE
TCP/IP STACK SENDS A PACKET TO
THE TUNNEL'S VIRTUAL LAN

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

330

DHCP PACKET ? —YES→ GENERATE DHCP
RESPONSE

332

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

29

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

FIG. 19

TUNNEL APPLICATION — 429

5 — 430

TCP LAYER WITHIN TCP/IP STACK — 422

IP LAYER WITHIN TCP/IP STACK — 420

2 — 418

424

3

ROUTING TABLE — 416

428 — 4

426 (PSEUDO ADAPTER)

PHYSICAL NETWORK DRIVER — 414

1 — 410

412

PHYSICAL NETWORK

FIG. 20

FIG. 21

FIG. 22

TUNNEL APPLICATION | PSEUDO NETWORK ADAPTER

SEND RELAY —500

SEND REQUEST —502

RECEIVE RESPONSE —504

PASS CLIENT ADAPTER —506
TO PSEUDO ADAPTER

508

CLIENT DATA

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 —510
STATUS TO STACK

RECEIVE DHCP —512
REQUEST FROM STACK

SEND DHCP RESPONSE —517
TO STACK

RECEIVE ARP REQUEST
FOR MAC ADDRESS
FOR TUNNEL SERVER —516
PSEUDO ADAPTER
IP ADDRESS

SEND ARP RESPONSE —518
BACK TO STACK

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.
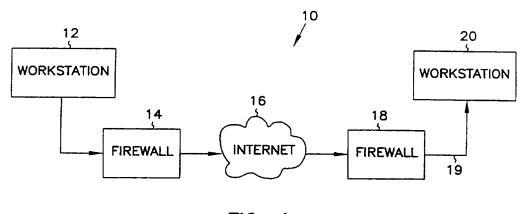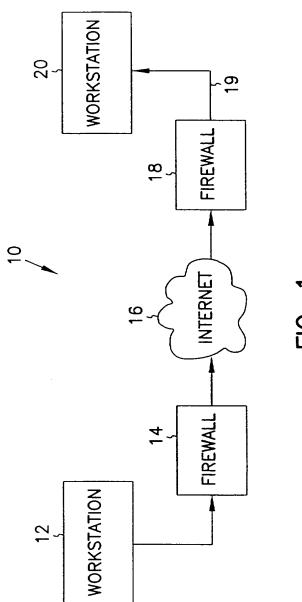
FIG. 1

At least one drawing originally filed was informal and the print reproduced here is taken from a later filed formal copy.
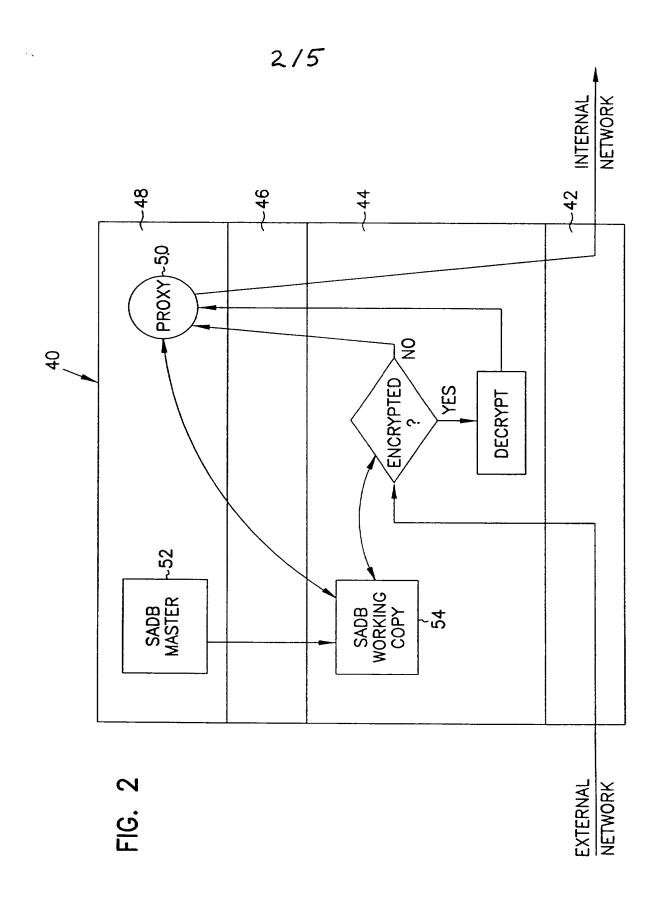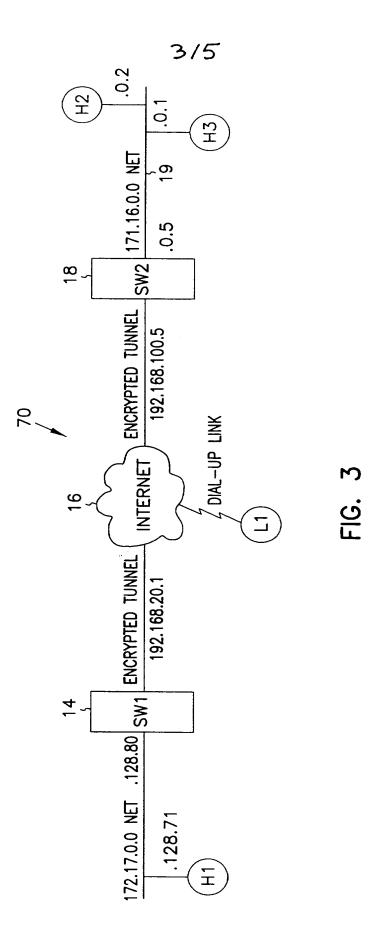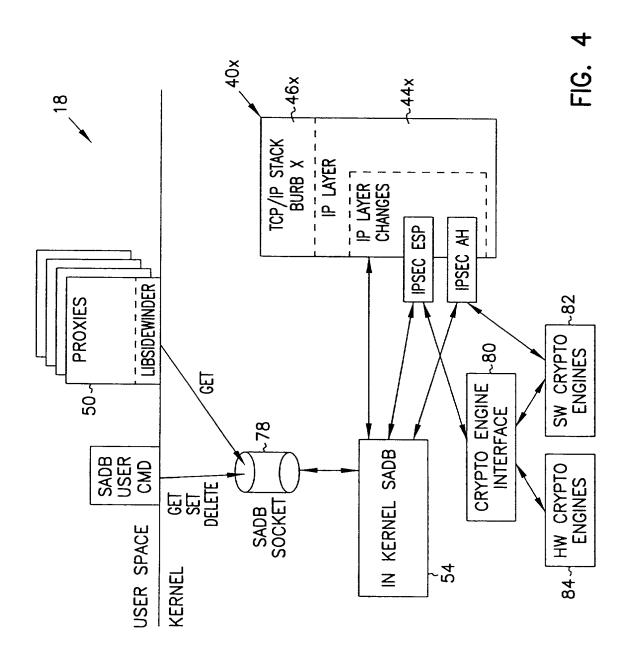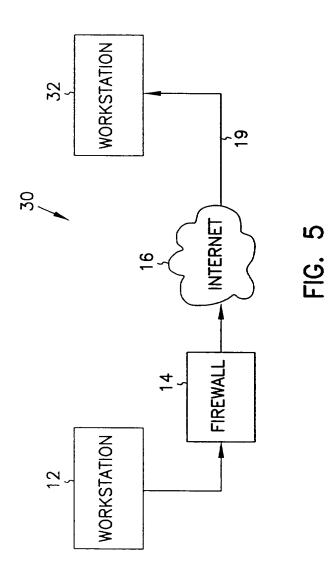
FIG. 1

2/5

FIG. 2

INTERNAL
NETWORK

48

46

44

42

40

PROXY ~50

SADB
MASTER ~52

SADB
WORKING
COPY

54

ENCRYPTED
?

NO

YES

DECRYPT

EXTERNAL
NETWORK

FIG. 3

-4 /5

FIG. 4

5/5

FIG. 5

VIRTUAL PRIVATE NETWORK ON APPLICATION GATEWAY

5   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.

10   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

15   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

20   "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-*

25   *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

30   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.

5 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

10 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-*

15 *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

20 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)

25 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

30 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

5   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

10   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

15   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

20   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

25   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

30   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

5   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

10   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,

15   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

20   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

25   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

30   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

5   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

10   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
15   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.

20   - 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.

25   - 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.

30   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.

5   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.

10   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
15   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
20   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
25   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
30   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.

5   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.

10   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

15   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

20   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

25   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

30   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

5   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

10   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

15   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

20   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

25   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

30   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

5   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

10   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

15   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.

20   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

25   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

5   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

10   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

15   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

20   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.

25   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

30   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

5 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

10 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).

15 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

20 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

25 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

30 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

5   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.

10   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

15   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

20   (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

25   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.

30   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

5   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

10   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
15   if [ $# -ne 1 ]
     then
           echo "usage: $0 <on>|<off>" >&2
           exit 1
     fi
20   ONOFF=$1

     addsa ()
     {
     IPADDRESS=$2
25   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
30   PORT=0
     PROTOCOL=0
     UID=0
     DESALG=1                       # I = DES-CBC
     IVLEN=4                        # bytes
35   DESKEY=0b0b0b0b0b0b0b0b
     DESKEYLEN=8                    # bytes
     AHALG=1                        # 1 = MD5
     AHKEY=3031323334353637303132333435363 7
     AHKEYLEN=16                    # bytes
40   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  0b0b0b0b0b0b0b0b0b
             8  0b0b0b0b0b0b0b0b0b
        MD5-RFC1828(1)  4
             16  3031323334353637303132333435363 7
             16  3031323334353637303132333435363 7
        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  0b0b0b0b0b0b0b0b0b
             8  0b0b0b0b0b0b0b0b0b
        MD5-RFC1828(1) 4
             16 3031323334353637303132333435363 7
             16 3031323334353637303132333435363 7
        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=
```

5

```
# 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
```

10

```
#  Hypothetical SW2 Config File
#
#  Fields are laid out in the following manner:
#   srcaddrornet= localSPI= peeraddr= peerSPI=
        realsrcaddr= localaddr= key=
```

15

```
#   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
```

20

25

With this setup, all traffic is encrypted using one key, no matter who is

talking to whom. For example, traffic from H2 to H1 as well as traffic from H3

30   to H1 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

35   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 H1, and H3 excludes H2 in the same way.

40

```
#   Hypothetical SW1 Config File
#
#   Fields are laid out in the following manner:
#   srcaddrornet= localSPI= peeraddr= peerSPI=
5 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
10 peerSPI=777 \
       realsrcaddr=192.168.100.5
localaddrs=178.17.128.71 \
       key=0x0a0a0a0a0a0a0a0a


15 # 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
20 localaddrs=178.17.128.71 \
       key=0x0b0b0b0b0b0b0b0b


#   Hypothetical SW2 Config File
#
25 #   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
30 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
35
# 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 \
40       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

45 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

5   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

10   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

15   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

20   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
25   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
30   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.

5   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

10   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.

15   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

20   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

25   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.

30   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

5   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

10   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.

15   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

20   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

25   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

30   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.

Petitioner RPX Corporation - Ex. 1002, p. 361

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
5 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
10 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
15 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
20 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
25 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
30 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
5   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.

10

5.   A firewall, comprising:

a first communications interface;

a second communications interface;

a network protocol stack connected to the first and the second
15   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

20   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.

25 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
30 (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;

5   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

10   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

15   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.

20   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

25   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
5 application level proxy;

determining an authentication protocol appropriate for the connection request;

executing the authentication protocol to authenticate the connection request; and

10   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
15 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
20 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
25 connection request was encrypted or not encrypted.

The
**Patent
Office**

2̲8

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:	1304381
Application Number:	10714849
International Application Number:	
Confirmation Number:	3154
Title of Invention:	Agile network protocol for secure communications using secure domain names
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	22907
Filer:	Steve S. Chang
Filer Authorized By:	
Attorney Docket Number:	000479.00111
Receipt Date:	09-NOV-2006
Filing Date:	18-NOV-2003
Time Stamp:	17:32:59
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	00111IDS.pdf	137984	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):	6416940

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.

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.
10/714,849	11/18/2003	Victor Larson	000479.00111	3154

22907   7590   12/07/2006

BANNER & WITCOFF
1001 G STREET N W
SUITE 1100
WASHINGTON, DC 20001

EXAMINER
LIM, KRISNA

ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 12/07/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

PTO-90C (Rev. 10/03)

-- *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 _1_ 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 _____ .

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) _1-27_ 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) _____ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☒ Claim(s) _1-27_ 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: _____ .

DETAILED ACTION

Election/Restrictions

Restriction to one of the following inventions is required under 35 U.S.C. 121:

1.   Claims 1-27 are still pending for examination.

2.   Restriction to one of the following inventions is required under 35 U.S.C. § 121:

I.   Claims 1-12 and 26-27 drawn to a system for providing a secure domain name service over a computer network, comprising: a server and a domain name database, classified in Class 709, subclass 226..

II.   Claims 13-25 drawn a method for registering a secure domain name, comprising steps of: a) receiving a request, b) verifying ownership information, and c) registering the secure domain name, classified in Class 709, subclass 223.

3.   Inventions I and II are related as combination and subcombination. Inventions in this relationship are distinct if it can be shown that (1) the combination as claimed does not require the particulars of the subcombination as claimed for patentability, and (2) that the subcombination has utility by itself or in other combinations (MPEP § 806.05(c)). In the instant case, the combination as claimed of Invention I, claims 1-12 and 26-27, does not require the particular steps of the subcombination: a) receiving a request, b) verifying ownership information, and c) registering the secure domain name, and the subcombination Group 2, claims 13-25, does not require: a server and a domain name database.

4   Moreover, the searches for these two inventions would not be co-extensive because these groups would require different searches on PTO's classification class and subclass as following:

1) The Group I search (claims 1-12 and 26-27) would require use of search class 709, subclass 226 (which would not required for the group II).

2) The Group II search (claims 13-25) would require use of search class 709, subclass 223 (which would not required for the group I).

5.   Applicant is reminded that the required for response to this requirement is **30 days, not one month.**

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

December 4, 2006

KRISNA LIM
PRIMARY EXAMINER

Index of Claims

Application/Control No. 10/714,849	**Applicant(s)/Patent under Reexamination** LARSON ET AL.
Examiner Krisna Lim	**Art Unit** 2153

√	Rejected	
=	Allowed	
—	(Through numeral) Cancelled	
+	Restricted	
N	Non-Elected	
I	Interference	
A	Appeal	
O	Objected	

Final	Original	12/4/06	Final	Original		Final	Original	
	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	

UNITED STATES PATENT AND TRADEMARK OFFICE

CONFIRMATION NO. 3154

Bib Data Sheet

SERIAL NUMBER 10/714,849	FILING OR 371(c) DATE 11/18/2003 **RULE**	CLASS 709	GROUP ART UNIT 2153	ATTORNEY DOCKET NO. 000479.00111

APPLICANTS

Victor Larson, Fairfax, VA;
Robert Durham Short III, Leesburg, VA;
Edmund Colby Munger, Crownsville, MD;
Michael Williamson, South Riding, VA;

** CONTINUING DATA *************************

This application is a CON of 09/558,210 04/26/2000 ABN which is a CIP 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 **
02/12/2004

Foreign Priority claimed ☐ yes ☑ no	STATE OR COUNTRY VA	SHEETS DRAWING 40	TOTAL CLAIMS 23	INDEPENDENT CLAIMS 5
35 USC 119 (a-d) conditions met ☐ yes ☑ no ☐ Met after Allowance				
Verified and Acknowledged Examiner's Signature Initials				

ADDRESS
22907

TITLE

Agile network protocol for secure communications using secure domain names

FILING FEE RECEIVED 1294	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

7/22/06

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	:	
	:	
Victor Larson et al.	:	Confirmation No.: 3154
	:	
Application No.: 10/714,849	:	Group Art Unit: 2153
	:	
Filed: November 18, 2003	:	Examiner: Lim, Krisna
	:	
For: An Agile Network Protocol for Secure	:	Atty Docket: 007170.00025
Communications Using Secure	:	
Domain Name		

RESPONSE TO RESTRICTION REQUIREMENT

U.S. Patent and Trademark Office
Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

Sir:

This paper is responsive to the Office Action mailed December 7, 2006, in connection with the above-identified patent application and is filed prior to the expiration of the one (1) month period for response set therein.

In response, Applicants elect without traverse, the invention of Group I, i.e., claims 1-12 and 26-27. Applicants reserve the right to file divisional applications directed to the subject matter of the non-elected claims prior to the termination of proceedings in this patent application.

No fee is believed to be associated with the filing of this paper. However, should the U.S. Patent and Trademark Office determine a fee is required, authorization is given to charge our Deposit Account No. 19-0733.

Respectfully submitted,

_____/John M. Fleming/_____
John M. Fleming
Reg. No. 56,536

January 8, 2007

Banner & Witcoff, Ltd.
Eleventh Floor
1001 G Street, N.W.
Washington, D.C. 20001-4597

Electronic Acknowledgement Receipt

EFS ID:	1422133
Application Number:	10714849
International Application Number:	
Confirmation Number:	3154
Title of Invention:	Agile network protocol for secure communications using secure domain names
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	22907
Filer:	John McClellan Fleming/Laura Sunderland
Filer Authorized By:	John McClellan Fleming
Attorney Docket Number:	007170.00025
Receipt Date:	08-JAN-2007
Filing Date:	18-NOV-2003
Time Stamp:	13:07:43
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	Response to Election / Restriction Filed	717000025_response_to_restriction_requirement.pdf	19236	no	1
Warnings:					

Information:		
Total Files Size (in bytes):		19236

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.

PATENT APPLICATION FEE DETERMINATION RECORD
Effective October 1, 2003

Application or Docket Number: 10/714,849

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS	23	
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	23 minus 20=	* 3
INDEPENDENT CLAIMS	5 minus 3 =	* 2
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	385.00	OR	BASIC FEE	770.00
X$ 9=		OR	X$18=	54
X43=		OR	X86=	172
+145=		OR	+290=	
TOTAL		OR	TOTAL	996

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	* 27	Minus	** 23	= 4
Independent	* 7	Minus	*** 5	= 2
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	126
X43=		OR	X86=	172
+145=		OR	+290=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	298

13, 15 17 21
1/8/07

AMENDMENT B

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	* 27	Minus	** 27	= —
Independent	* 7	Minus	*** 7	= —
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 9=		OR	X$18=	
X43=		OR	X86=	
+145=		OR	+290=	
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=	
X43=		OR	X86=	
+145=		OR	+290=	
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.

Best Available Copy

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.
10/714,849	11/18/2003	Victor Larson	007170.00025	3154

22907 7590 03/21/2007	EXAMINER	
BANNER & WITCOFF, LTD.	LIM, KRISNA	
1100 13th STREET, N.W.		
SUITE 1200	ART UNIT	PAPER NUMBER
WASHINGTON, DC 20005-4051	2153	

SHORTENED STATUTORY PERIOD OF RESPONSE	MAIL DATE	DELIVERY MODE
3 MONTHS	03/21/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

PTOL-90A (Rev. 10/06)

	Application No.	Applicant(s)
Office Action Summary	10/714,849	LARSON 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 <u>3</u> 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 <u>08 January 2007</u>.

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) *1-27* is/are pending in the application.

   4a) Of the above claim(s) *13-25* is/are withdrawn from consideration.

5) ☐ Claim(s) _____ is/are allowed.

6) ☒ Claim(s) *1-8,26 and 27* is/are rejected.

7) ☒ Claim(s) *9-12* 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 1-27 are pending in the application. Claims 1-12 and 26-27 are elected without traverse for examination. And, non-elected claims 13-25 are withdrawn from consideration.

2.   Claims 1-12 and 25-27 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 unclear where a query is coming from or who queries to the server and what is being queried. Moreover, it is unclear how the system is providing a secure domain name service.

        In claim 26, it is unclear what kind of an apparatus is it. It is unclear how this apparatus is configured. And, it is unclear for what a computer network address is queried.

        In claim 27, it is similar to claim 26.

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 1-12 and 26-27 are rejected under 35 U.S.C. § 103(a) as being unpatentable over IP Security Chapter 13 of XP-002167283,[Hereinafter XP]. This reference was cited in 1449 Form that was submitted by the applicant.

5.   XP disclosed the invention substantially as claimed. Taking claims 1-3, and 26-27 as exemplary claims, the reference disclosed a system for providing a secure domain name service (IP Security, IPSec, page 399) over a computer network, comprising:

a) a server connected to a computer network, the server authenticating a query
for a secure computer network address having a top-level domain (security at IP level),
page 399) reserved for secure network connection;

b) a domain name database (routers) connected to the computer network
through the server the domain name database storing secure computer network
addresses for the computer network.

6.   While XP further disclosed IP and IPsec (IP security, page 400), XP did not
explicitly call its security at IP level as a non-standard top-level domain such as .scom,
.sorg, .snet, .sedu, .smil and .sint, etc. It would have been obvious to one of an ordinary
skilled in the art to recognize that the domain names such as .com, .org, .net, .edu, .mil,
.int, .gov, etc. are the well-known top-level domains for companies, organizations,
government, etc.(see any computer dictionaries) and the additional security of HTTPS
to the standard HTTP. Thus, adding additional security to these standard domain
names of .com, .org, . net, .edu, .mil, .int, etc. to become .scom, .sorg, .snet, .sedu,
.smil, and .sint, etc. would be obviously similar to the teaching of IP and IPsec of XP.
Moreover, having these non-secured top-level domains secured would have been a
desirable feature in the art because the security of the Internet is a vital element for the
national security, business security, individual's privacy security, etc.

7.   As to claim 4, XP disclosed the computer network includes the Internet (e.g., see
Fig. 13.1, page 401).

8.   As to claim 5, XP disclosed the server comprises an edge router (e.g., see
routing application, a figure 13.10, page 402).

9.   As to claim 6, XP disclosed the server authenticates the query using a
cryptographic technique (e.g., see authentication algorithm, a figure 13.2, Encryption
Algorithm and Authentication Algorithm on page 403, Cryptographic keys on page 404).

10.   As to claim 7, XP disclosed the server is connectable to a virtual private network link through the computer network (e.g., see page 421).

9.   As to claim 8, XP disclosed the secure communication link is one of a plurality of secure communication links in a hierarchy of secure communication links (e.g., see IP Security of Fig. 13.1 and IP security architecture on page 402 and 403).

10.   Claims 9-12 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

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 7, 2007

KRISNA LIM
PRIMARY EXAMINER

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
Application Number	10714849	
Filing Date	2003-11-18	
First Named Inventor	Victor Larson	
Art Unit	~~2148~~ 2153	
Examiner Name	~~TBD~~ K. Lim	
Attorney Docket Number	000479.00111	

U.S.PATENTS

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
VL	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 Code[1]	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 Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
VL	1	0 838 930	EP	A	1998-04-29	Compaq Computer Corp.		☐
VL	2	0 814 589	EP	A	1997-12-29	AT&T Corp.		☐
VL	3	2 334 181	GB	A	1999-08-11	NEC Technologies; Globalmart Ltd.		☐

EFS Web 2.0

		Application Number	10714849
INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		Filing Date	2003-11-18
		First Named Inventor	Victor Larson
		Art Unit	2143 *2153*
		Examiner Name	~~TBD~~ *K. Lim*
		Attorney Docket Number	000479.00111

Vic	4	9827783	WO	A	1998-06-25	Northern Telecom Limited; Antonio, G; Hui, Margare		☐
Vic	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
Vic	1	Laurie Wells (LANCASTERBIBELMAIL MSN COM); "Subject: Security Icon" USENET NEWSGROUP, 19 October 1998, XP002200606	☐
Vic	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	3/7/09

*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.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)	Application Number	10714849
	Filing Date	2003-11-18
	First Named Inventor	Victor Larson
	Art Unit	2143 2153
	Examiner Name	TBD K. Lim
	Attorney Docket Number	000479.00111

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.

Substitute for form 1449A/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

		Complete if Known	
	Application Number	10/714,849	
	Filing Date	November 18, 2003	
	First Named Inventor	Victor LARSON	
	Art Unit	~~2143~~ 2153	
	Examiner Name	~~TBA~~ K. Lim	
Sheet	1	of	2	Attorney Docket Number	000479.00111

U.S. PATENT DOCUMENTS

Examiner Initials *	Cite No.¹	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
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			
		US-			

FOREIGN PATENT DOCUMENTS

Examiner Initials*	Cite No.¹	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 Passages or Relevant Figures Appear	T⁶
/KL/		EP 0 838 930 A2	04/29/1998	Kenneth F. ALDEN, et al.		
/KL/		DE 199 24 575 A1	12/02/1999	Joseph E. PROVINO		

Examiner Signature	*[signature]*	Date Considered	3/6/07

*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 See Kinds Codes of USPTO Patent Documents at www.uspto.gov or MPEP 901.04. ³ Enter Office that issued the document, by the two-letter code (WIPO Standard ST.3). ⁴ For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. ⁵ Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST. 16 if possible. ⁶ Applicant is to place a check mark here if English language Translation is attached.

This collection of information is required by 37 CFR 1.97 and 1.98. 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 take 2 hours 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.

Substitute for form 1449B/PTO

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Use as many sheets as necessary)

				Complete if Known	
				Application Number	10/714,849
				Filing Date	November 18, 2003
				First Named Inventor	Victor LARSON
				Art Unit	2143
				Examiner Name	TBD
Sheet	2	of	2	Attorney Docket Number	000479.00111

NON PATENT LITERATURE DOCUMENTS

Examiner Initials *	Cite No.[1]	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.	T[2]
VC		Donald E. EASTLAKE, III, "Domain Name System Security Extensions", Internet Draft, April 1998.	
VC		P. SRISURESH, et al., "DNS Extensions to Network Address Translators', Internet Draft, July 1998.	
VC		D.B. CHAPMAN, et al., "Building Internet Firewalls, chapters 8 and 10 (parts)", pp. 278-296 and pp. 351-375.	

Examiner Signature		Date Considered	3/6/07

PTO-1449 (Modified)	ATTY. DOCKET NO. 000479.00111	SERIAL NUMBER TBD 10/714,849
U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	APPLICANT Victor Larson et al.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT	FILING DATE Herewith	GROUP ART UNIT TBD 2153

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
/JL	6,119,171	9/2000	Alkhatib			
/JL	5,588,060	12/24/96	Aziz			
/JL	5,689,566	11/18/9	Nguyen			
/JL	5,842,040	11/24/98	Hughes et al.			
/JL	4,933,846	06/12/90	Humphrey et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
/JL	199 24 575	12/2/99	DE				
/JL	0 838 930	4/29/98	EPO				
/JL	2 317 792	4/1/98	GB				
/JL	0 814 589	12/29/97	EPO				
/JL	WO 98/27783	6/25/98	PCT				

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/JL	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
	FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security: Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER KRISHA Lim	DATE CONSIDERED 3/6/07

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)	ATTY. DOCKET NO. 000479.00111	SERIAL NUMBER TBD 10/714,849
U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	APPLICANT Victor Larson et al.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
	6,353,614	3/5/02	Borella et al.			
	6,332,158	12/18/01	Risley et al.			
	6,330,562	12/11/01	Boden et al.			
	6,286,047	9/4/01	Ramanathan et al.			
	6,243,749	6/5/01	Sitaraman et al.			
	6,226,751	5/1/01	Arrow et al.			
	6,178,505	1/23/01	Schneider et al.			
	6,119,171	9/12/00	Alkhatib			
	6,079,020	6/20/00	Liu			
	6,052,788	4/18/00	Wesinger, Jr. et al.			
	6,016,318	1/18/00	Tomoike			
	6,006,259	12/21/99	Adelman et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

EXAMINER *Krisna Lim*	DATE CONSIDERED 3/6/07

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. 000479.00111	SERIAL NUMBER TBD
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
KL	5,905,859	5/18/99	Holloway et al.			
	5,898,830	4/27/99	Wesinger, Jr. et al.			
	5,892,903	4/6/99	Klaus			
	5,878,231	3/2/99	Baehr et al.			
	5,805,801	9/8/98	Holloway et al.			
	5,796,942	8/18/98	Esbensen			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO
KL	WO 00/70458	11/23/00	PCT			

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

KL	Linux FreeS/WAN Index File, printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/ on February 21, 2002, 3 Pages
	J. Gilmore, "Swan: Securing the Internet against Wiretapping", printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/rationale.html on February 21, 2002, 4 pages
	Glossary for the Linux FreeS/WAN project, printed from http://liberty.freeswan.org/freeswan_trees/freeswan-1.3/doc/glossary.html on February 21, 2002, 25 pages
	Alan O. Frier et al., "The SSL Protocol Version 3.0", November 18, 1996, printed from http://www.netscape.com/eng/ssl3/draft302.txt on February 4, 2002, 56 pages

EXAMINER KRISHNA Linn	DATE CONSIDERED 3/6/07

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. 000479.00111	SERIAL NUMBER ~~TBD~~ 10/714,849
	APPLICANT Victor Larson et al.	
	FILING DATE Herewith	GROUP ART UNIT ~~TBD~~ 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				
	WO 01 50688	7/12/01	PCT				
	WO 98 59470	12/30/98	PCT				
	WO 99 48303	9/23/99	PCT				
	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
	Search Report (dated 8/23/02), International Application No. PCT/US01/13260
	Shree Murthy et al., "Congestion-Oriented Shortest Multipath Routing", Proceedings of IEEE INFOCOM, 1996, pages 1028-1036
	Jim Jones et al., "Distributed Denial of Service Attacks: Defenses", Global Integrity Corporation, 2000, pages 1-14
	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
/KL	Rich Winkel, "CAQ: Networking With Spooks: The NET & The Control Of Information", Internet Newsgroup, June 21, 1997, 4 pages

EXAMINER KRISNA LIM	DATE CONSIDERED 3/6/07

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)	ATTY. DOCKET NO. 000479.00111	SERIAL NUMBER TBD
U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	APPLICANT Victor Larson et al.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
KL	6,016,512	1/20000	Christian Huitema	⌐—		

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO
KL	WO 98 55930	12/10/98	PCT	⌐—		

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

KL	Search Report (dated 10/7/02), International Application No. PCT/US01/13261
	F. Halsall, "Data Communications, Computer Networks And Open Systems", Chapter 4, Protocol Basics, 1996, pages 198-203
	Reiter, Michael K. and Rubin, Aviel D. (AT&T Labs – Research), "Crowds: Anonymity for Web Transmissions", pages 1-23
	Dolev, Shlomi and Ostrovsky, Rafil, "Efficient Anonymous Multicast and Reception" (Extended Abstract), 16 pages
	Rubin, Aviel D., Greer, Daniel, and Ranum, Marcus J. (Wiley Computer Publishing), "Web Security Sourcebook", pages 82-94
	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 *3/6/07*

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.

Index of Claims

Application/Control No.	**Applicant(s)/Patent under Reexamination**
10/714,849	LARSON ET AL.
Examiner	**Art Unit**
Krisna Lim	2153

R	Rejected	−	(Through numeral) Cancelled	
=	Allowed	÷	Restricted	
N	Non-Elected			
I	Interference			
A	Appeal			
O	Objected			

Final	Original	3/7/07							Final	Original									Final	Original							
	1									51										101							
	2	R								52										102							
	3	R								53										103							
	4	R								54										104							
	5	R								55										105							
	6	R								56										106							
	7	R								57										107							
	8	R								58										108							
	9	O								59										109							
	10	O								60										110							
	11	O								61										111							
	12	O								62										112							
	13	N								63										113							
	14	N								64										114							
	15	N								65										115							
	16	N								66										116							
	17	N								67										117							
	18	N								68										118							
	19	N								69										119							
	20	N								70										120							
	21	N								71										121							
	22	N								72										122							
	23	N								73										123							
	24	N								74										124							
	25	N								75										125							
	26	R								76										126							
	27	R								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							

Search Notes	Application/Control No.	Applicant(s)/Patent under Reexamination
	10/714,849	LARSON ET AL.
	Examiner	Art Unit
	Krisna Lim	2153

SEARCHED

Class	Subclass	Date	Examiner
709	226, 221	3/7/2007	KL
713	201	3/7/2007	KL

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
EAST	9/2/2006	KL
Inventors	3/7/2007	

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner

U.S. Patent and Trademark Office

Part of Paper No. 20060902

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)

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:	77580-027/5496	*Original to Follow by Mail:*	No
		Number of Pages, Including Cover:	

Re: In re Application of: Victor Larson, et al.

Serial No.: 10/714,849

Filing Date: November 18, 2003

Title: An Agile Network Protocol For Secure Communications Using Secure Domain Names

Docket No.: 77580-027 (VRNK-1CP3CN)

Message:

Please enter the attached Change of Correspondence Address And Revocation Of Power Of Attorney And

Appointment of New Attorney.

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of: Victor Larson, et al.
Serial No.:   10/714,849
Filing Date: November 18, 2003
Title: An Agile Network Protocol For Secure Communications
     Using Secure Domain Names
Group Art Unit: 2153
Confirmation No.:   3154
Docket No.:   77580-027 (VRNK-1CP3CN)

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 11 2007
Date

Cynthia Joseph

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of:	Victor Larson, et al.
Serial No.:	10/714,849
Filing Date:	November 18, 2003
Title:	An Agile Network Protocol For Secure Communications Using Secure Domain Names
Docket No.:	77580-027 (VRNK-1CP3CN)

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 1543706-1.077580.0027

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: 6/11/07

By _____
Name: Randall Larson
Title: President/CEO

BST99 1543706-1.077580.0027

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.
10/714,849	11/18/2003	Victor Larson	007170.00025	3154

22907   7590   06/21/2007
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
06/21/2007	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)

APPLICATION NO./ CONTROL NO.	FILING DATE	FIRST NAMED INVENTOR / PATENT IN REEXAMINATION	ATTORNEY DOCKET NO.
10714849	11/18/03	LARSON ET AL.	007170.00025

EXAMINER
Krisna Lim

ART UNIT	PAPER
2153	20070618

BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

DATE MAILED:

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner for Patents

Per the telephone request by the attorney of the record, Examiner provided the reforence.

KRISNA LIM
PRIMARY EXAMINER

PTO-90C (Rev.04-03)

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
10/714,849	11/18/2003	Victor Larson	007170.00025

CONFIRMATION NO. 3154

22907
BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

OC000000024495906

Date Mailed: 06/22/2007

NOTICE REGARDING POWER OF ATTORNEY

This is in response to the Power of Attorney filed 06/12/2007 . The Power of Attorney in this application is not accepted for the reason(s) listed below:

- The Power of Attorney is from an assignee and the Certificate required by 37 CFR 3.73(b) has not been received.

Office of Initial Patent Examination (571) 272-4000, or 1-800-PTO-9199
OFFICE COPY

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)

FACSIMILE

Date: July 11, 2007		**Time Sent:**	

To:	**Company:**	**Facsimile No:**	**Telephone No:**
Commissioner for Patents	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:	77580-042/5496	*Original to Follow by Mail:*	No
		Number of Pages, Including Cover:	21

Re: In re Application of: Victor Larsen, et al.

Serial No.: 10/714,849

Filing Date: November 18, 2003

Title: An Agile Network Protocol For Secure Communications

Docket No.: 77580-042 (VRNK-1CP3CN)

Message:

Please enter the attached Amendment "B" in response to the Office Action of March 21, 2007

RECEIVED
CENTRAL FAX CENTER
JUL 11 2007

☑004/018

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Victor Larsen et al.
Serial No.:   10/714,849
Filing Date: November 18, 2003
Group Art Unit: 2153
Examiner: Krisna Lim
Confirmation No.:   3154
Title: An Agile Network Protocol for Secure Communications
Docket No: 77580-042 (VRNK-1CP3CN)

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: Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted (571-273-8300) to the USPTO, on the date indicated below.

Date: 11 July 2007

_____
(Cynthia Joseph)

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

AMENDMENT "B"

This paper is a response to the Official action dated 21 March 2007.

The Applicants request reconsideration and further examination in view of the following:

Amendments to the Claims, as reflected in the listing of claims beginning on page 2 of this paper; and

Remarks, beginning on page 14 of this paper.

07/12/2007 TL0111   00000039 501133   10714849
02 FC:1202   500.00 DA

BST99 1546854-1.077580.0042

PAGE 4/18 * RCVD AT 7/11/2007 5:19:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:2738300 * CSID:1 617 535 3800 * DURATION (mm-ss):04-52

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

<u>Listing of Claims:</u>

1. (Canceled).

2. (Currently Amended) The system of claim ~~1~~29, wherein the top-level domain name is a non-standard top-level domain name.

3. (Original) The system of claim 2, wherein the non-standard top-level domain name is one of .scom, .sorg, .snet, .sgov, .sedu, .smil and .sint.

4. (Currently Amended) The system of claim ~~1~~28, wherein the ~~computer~~ <u>communication</u> network includes the Internet.

5. (Currently Amended) The system of claim ~~1~~28, wherein the <u>domain name service system</u>~~server~~ comprises an edge router.

6. (Currently Amended) The system of claim ~~1~~29, wherein the <u>domain name service system is configured to authenticate</u> ~~server authenticates~~ the query using a cryptographic technique.

7. (Currently Amended) The system of claim ~~1~~28, wherein the <u>domain name service system</u>~~server~~ is connectable to a virtual private network ~~link~~ through the ~~computer~~<u>communication</u> network.

8. (Currently Amended) The system of claim 7, wherein the <u>virtual private network</u>~~secure communication link~~ is one of a plurality of secure communication links in a hierarchy of secure communication links.

BST99 1546854-1.077580.0042

9. (Currently Amended) The system of claim 7, wherein the virtual private network is based on inserting into each data packet <u>communicated over a secure communication link</u> one or more data values that vary according to a pseudo-random sequence.

10. (Currently Amended) The system of claim 7, wherein the virtual private network is based on a ~~computer~~-network address hopping regime that is used to pseudorandomly change ~~computer~~-network addresses in packets transmitted between a first ~~computer~~-<u>device</u> and a second ~~computer~~<u>device</u>.

11. (Currently Amended) The system of claim 7, wherein the virtual private network is based on comparing a value in each data packet transmitted between a first ~~computer~~-<u>device</u> and a second ~~computer~~-<u>device</u> to a moving window of valid values.

12. (Currently Amended) The system of claim 7, wherein the virtual private network is based on a comparison of a discriminator field in a header of each data packet to a table of valid discriminator fields maintained for a first ~~computer~~<u>device</u>.

13. (Withdrawn) A method for registering a secure domain name, comprising steps of:
  receiving a request for registering a secure domain name;
  verifying ownership information for an equivalent non-secure domain name corresponding to the secure domain name;
  registering the secure domain name in a secure domain name service when the ownership information for the equivalent non-secure domain name is consistent with ownership information for the secure domain name.

14. (Withdrawn) The method according to claim 13, wherein the step of verifying ownership information includes steps of:
  determining whether the equivalent non-secure domain name corresponding to the secure domain name has been registered in a non-secure domain name service; and

querying whether the equivalent non-secure domain name should be registered in the nonsecure domain name service when the equivalent non-secure domain name has not been registered in the non-secure domain name service.

15. (Withdrawn) A computer-readable storage medium, comprising:

a storage area; and computer-readable instructions for a method for registering a secure domain name, the method comprising steps of:

receiving a request for registering a secure domain name;

verifying ownership information for an equivalent non-secure domain name corresponding to the secure domain name; and

registering the secure domain name in a secure domain name service when the ownership information for the equivalent non-secure domain name is consistent with ownership information for the secure domain name.

16. (Withdrawn) The computer-readable medium according to claim 15, wherein the step of verifying ownership information includes steps of:

determining whether the equivalent non-secure domain name corresponding to the secure domain name has been registered in a non-secure domain name service; and

querying whether the equivalent non-secure domain name should be registered in the non-secure domain name service when the equivalent non-secure domain name has not been registered in the non-secure domain name service.

17. (Withdrawn) A method for registering a domain name, comprising steps of:

(i) receiving a request for registering a first domain name;

(ii) verifying ownership information for a second domain name corresponding to the first domain name; and

(iii) registering the first domain name when the ownership information for the second domain name is consistent with ownership information for the first domain name.

BST99 1546854-1.077580.0042

18. (Withdrawn) The method of claim 17, wherein the first domain name comprises a nonstandard top-level domain and the second domain name comprises a standard top-level domain.

19. (Withdrawn) The method of claim 17, further comprising the step of storing information corresponding to the registration performed in step (iii) in a database separate from a database storing information for standard domain name registrations.

20. (Withdrawn) The method according to claim 17, wherein the step of verifying ownership information includes steps of:

(a) determining whether the second domain name has been registered in a domain name service; and

(b) querying whether the second domain name should be registered in the domain name service when the second domain name has not been registered in the domain name service.

21. (Withdrawn) A computer-readable medium, comprising computer-readable instructions for a method for registering a domain name, the method comprising steps of:

(i) receiving a request for registering a first domain name;

(ii) verifying ownership information for a second domain name corresponding to the first domain name; and

(iii) registering the first domain name when the ownership information for the second domain name is consistent with ownership information for the first domain name.

22. (Withdrawn) The computer readable medium of claim 21, wherein the first domain name comprises a non-standard top-level domain and the second domain name comprises a standard top level domain.

23. (Withdrawn) The computer-readable medium of claim 21, wherein the step of verifying ownership information includes steps of:

(a) determining whether the second domain name has been registered in a domain name service; and

(b) querying whether the second domain name should be registered in the domain name service when the second domain name has not been registered in the domain name service.

24. (Withdrawn) The method of claim 13, wherein the secure domain name has a top-level domain reserved for secure network connections.

25. (Withdrawn) The computer-readable storage medium of claim 15, wherein the secure domain name has a top-level domain reserved for secure network connections.

26. (Canceled).

27. (Canceled).

28. (New) A system for providing a domain name service for establishing a secure communication link, the system comprising:

a domain name service system configured to be connected to a communication network, to store a plurality of domain names and corresponding network addresses, to receive a query for a network address, and to comprise an indication that the domain name service system supports establishing a secure communication link.

29. (New) The system of claim 28, wherein at least one of the plurality of domain names comprises a top-level domain name.

30. (New) The system of claim 28, wherein the domain name service system is configured to respond to the query for the network address.

31. (New) The system of claim 28, wherein the domain name service system is configured to provide, in response to the query, the network address corresponding to a

domain name from the plurality of domain names and the corresponding network
addresses.

32. (New) The system of claim 28, wherein the domain name service system is
configured to receive the query initiated from a first location, the query requesting the
network address associated with a domain name, wherein the domain name service
system is configured to provide the network address associated with a second location,
and wherein the domain name service system is configured to support establishing a
secure communication link between the first location and the second location.

33. (New) The system of claim 28, wherein the domain name service system is
connected to a communication network, stores a plurality of domain names and
corresponding network addresses, and comprises an indication that the domain name
service system supports establishing a secure communication link.

34. (New) The system of claim 28, wherein at least one of the plurality of domain
names is reserved for secure communication links.

35. (New) The system of claim 28, wherein the domain name service system
comprises a server.

36. (New) The system of claim 35, wherein the domain name service system
further comprises a domain name database, and wherein the domain name database stores
the plurality of domain names and the corresponding network addresses.

37. (New) The system of claim 28, wherein the domain name service system
comprises a server, wherein the server comprises a domain name database, and wherein
the domain name database stores the plurality of domain names and the corresponding
network addresses.

38. (New) The system of claim 28, wherein the domain name service system is configured to store the corresponding network addresses for use in establishing secure communication links.

39. (New) The system of claim 28, wherein the domain name service system is configured to authenticate the query for the network address.

40. (New) The system of claim 28, wherein at least one of the plurality of domain names comprises an indication that the domain name service system supports establishing a secure communication link.

41. (New) The system of claim 28, wherein at least one of the plurality of domain names comprises a secure name.

42. (New) The system of claim 28, wherein at least one of the plurality of domain names enables establishment of a secure communication link.

43. (New) The system of claim 28, wherein the domain name service system is configured to enable establishment of a secure communication link between a first location and a second location transparently to a user at the first location.

44. (New) The system of claim 28, wherein the secure communication link uses encryption.

45. (New) The system of claim 28, wherein the secure communication link is capable of supporting a plurality of services.

46. (New) The system of claim 45, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof.

BST99 1546854-1.077580.0042

47. (New) The system of claim 46, 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.

48. (New) The system of claim 45, wherein the plurality of services comprises audio, video, or a combination thereof.

49. (New) The system of claim 28, wherein the domain name service system is configured to enable establishment of a secure communication link between a first location and a second location.

50. (New) The system of claim 49, wherein the query is initiated from the first location, wherein the second location comprises a computer, and wherein the network address is an address associated with the computer.

51. (New) The system of claim 28, wherein the domain name service system comprises a domain name database connected to a communication network and storing a plurality of domain names and corresponding network addresses for communication,

wherein the domain name database is configured so as to provide a network address corresponding to a domain name in response to a query in order to establish a secure communication link.

52. (New) A machine-readable medium comprising instructions executable in a domain name service system, the instructions comprising code for:

connecting the domain name service system to a communication network;

storing a plurality of domain names and corresponding network addresses;

receiving a query for a network address; and

supporting an indication that the domain name service system supports establishing a secure communication link.

53. (New) A method of providing a domain name service for establishing a secure communication link, the method comprising:

connecting a domain name service system to a communication network, the domain name service system comprising an indication that the domain name service system supports establishing a secure communication link;

storing a plurality of domain names and corresponding network addresses; and

receiving a query for a network address for communication.

REMARKS

Claims 2-25 and 28-53 remain in the application. Claims 1, 26 and 27 have been canceled. Claims 2 and 4-12 have been amended and claims 28-53 have been added in order to more clearly define applicant's invention. Claims 13-25, drawn to a non-elected invention, are withdrawn from consideration. Applicants note with appreciation that claims 9-12 are considered allowable if rewritten in independent form,—._These claims however, have not been rewritten in independent form in the belief that all of the claims are now considered allowable.

Claims 1-12 and 25-27 have been rejected under 35 U.S.C. §112, second paragraph. Claims 1-12 and 26-27 have also been rejected under 35 U.S.C. §103 (a). These rejections are respectfully traversed and reconsideration is requested in view of the foregoing amendments and following remarks.

Claims 1-12 and 25 (sic) -27 have been rejected under 35 U.S.C. §112, second paragraph, because the Examiner believes that these claims are indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. It is noted that claim 25 is directed to a non-elected invention, and is considered withdrawn. Applicants believe the that it was the intent of the Examiner to reject claims 1-12, 26 and 27, and not claim 25, under 35 U.S.C. § 112, second paragraph. Accordingly, Applicant's comments are directed to the more limited grouping.

The Examiner states that with regard to claim 1, "it is unclear where the query is coming from or who queries to the server and what is being queried." The Examiner also states that "it is unclear how the system is providing a secure domain name service." Claim 1 has been canceled in favor of new claim 28. It is submitted that a query can originate from any source and provided to the domain name service recited in claim 28, with the domain name service configured "to receive a query for a network address." Further, the clarifying amendments presented by new claim 28 should address the second issue raised by the Examiner.

BST99 1546854-1.077580.0042

Regarding the rejection of claims 26 and 27, the Examiner states that "it is unclear what kind of apparatus [the claimed subject matter] it is. It is unclear how this apparatus is configured. And, it is unclear for what a computer network address is queried. (sic)" Again, it is submitted that the clarifying amendments presented by new claim 28 should render the rejection moot. Claim 28 specifically describes the configuration of the domain name service system for establishing a secure communication link.

Claims 1-12, 26 and 27 have also been rejected under 35 U.S.C. § 103(a) as being unpatentable over IP Security Chapter 13 of XP-002167283 (also referred to as XP). Claims 1, 26 and 27 have been canceled. It is submitted that the reference neither anticipates nor makes obvious the claimed invention as defined by new claim 28.

New claim 28 recites a system for providing a domain name service for establishing a secure communication link. The system comprises a domain name service system configured to be connected:

to a communication network,

to store a plurality of domain names and corresponding network addresses,

to receive a query for a network address, and

to comprise an indication that the domain name service system supports establishing a secure communication link.

It is submitted that the reference XP neither anticipates nor makes obvious the claimed subject matter of claim 28, and the claims dependent thereon. The Examiner cites pages 399 and 400 of XP. These pages describe prior art approaches to IP security known at the time the reference was published (1998). XP mentions:

> ...security mechanisms in a number of application areas, including electronic mail (S/MIME, PGP), client/server Kerberos), Web access (Secure Sockets Layer), and others. However, users have some security concerns that cut across protocol layers. For example, an enterprise can run a secure, private TCP/IP network by disallowing links to untrusted sites, encrypting packets that leave the premises, and authenticating packets that enter the premises. By implementing security at the IP level, an organization can ensure secure networking not only for applications that have security mechanisms but for the many security-ignorant applications.
>
> IP-level security encompasses three functional areas: authentication, confidentiality, and key management. The authentication mechanism

assures that a received packet was, in fact, transmitted by the party identified as the source in the packet header. In addition, this mechanism assures that the packet has not been altered in transit. The confidentiality facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties. The key management facility is concerned with the secure exchange of keys...

Thus, the reference describes the three function areas of any security system for protecting transmitted data across any network. The reference goes on to cite a 1994 report of the Internet Architecture Board that states that the Internet needs more and better security, and it identifies key areas for security mechanisms. Among these are the need to secure the network infrastructure form unauthorized monitoring and control of network traffic and the need to secure end-user-to-end-user traffic using authentication and encryption mechanisms. The report cites reported security incidents, the most serious types of attacks including IP spoofing, and various forms of eavesdropping and packet sniffing. Thus, the reference generally describes the need for secure transmission of data.

In describing IP Security (IPSec) the reference goes on to state:

IPSec provides the capability to secure communications across a LAN, across private and public wide area networks (WANs), and across the Internet. Examples of its use include the following:

Secure branch office connectivity over the Internet: A company can build a secure virtual private network over the Internet or over a public WAN This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead.

Secure remote access over the Internet: An end user whose system is equipped with IP security protocols can make a local call to an Internet service provider (ISP) and gain secure access to a company network. This reduces the cost of toll charges for traveling employees and telecommuters...

The reference states at the bottom of page 400, "the principal feature of IPSec that enables it to support these varied applications is that it can encrypt and/or authenticate all traffic at the IP level."

The reference describes the overall architecture in Figure 13.2 (page 403). The blocks shown in the Figure are defined on pages 403 and 404. It is clear that the

architecture described and shown in the reference does not describe nor even suggest a domain name service system configured to be connected to a communication network, to store a plurality of domain names and corresponding network addresses, to receive a query for a network address, and to comprise an indication that the domain name service system supports establishing a secure communication link. (cf., claim 28 of the above-identified application).

Dependent claims 2-12 and 29-51 currently under consideration in the application are dependent from independent Claim 28 discussed above and therefore are believed to be allowable over the applied reference 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.

Finally, claims 52 and 53 have been added. Claim 52 recites "a machine-readable medium comprising instructions executable in a domain name service system. The instructions comprise code for connecting the domain name service system to a communication network; storing a plurality of domain names and corresponding network addresses; receiving a query for a network address; and supporting an indication that the domain name service system supports establishing a secure communication link." Claim 53 recites "a method of providing a domain name service for establishing a secure communication link. The method comprises connecting a domain name service system to a communication network, the domain name service system comprising an indication that the domain name service system supports establishing a secure communication link; storing a plurality of domain names and corresponding network addresses; and receiving a query for a network address for communication."

Accordingly, all of the pending claims currently under consideration, claims 2-25 and 28-53, are believed to be patentable over the cited reference. An early and favorable action thereon is therefore earnestly solicited.

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: 11 July 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 1546854-1.077580.0042

PAGE 18/18 * RCVD AT 7/11/2007 5:19:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:2738300 * CSID:1 617 535 3800 * DURATION (mm-ss):04-52

COMBINED AMENDMENT & PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) (Large Entity)	Docket No. 77580-042 (VRNK-1CP3CN)
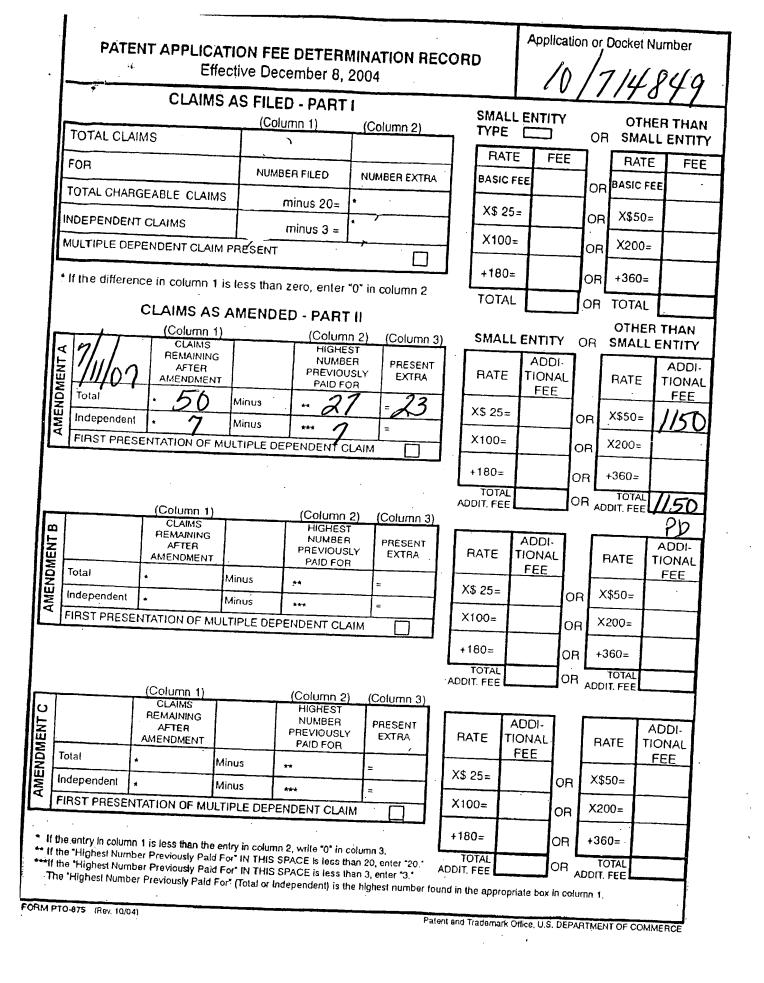
In Re Application Of: Victor Larsen, et al.

Application No. 10/714,849	Filing Date November 18, 2003	Examiner Krisna Lim	Customer No. 23630	Group Art Unit 2153	Confirmation No. 3154

Invention: AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS

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 ___March 21, 2007___ in the above-identified application.
                                            Date

The requested extension is as follows (check time period desired):

☒ One month   ☐ Two months   ☐ Three months   ☐ Four months   ☐ Five months

from: ___June 21, 2007___   until: ___July 21, 2007___
         Date   *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	37 -	27 =	10	x $50.00	$500.00
INDEP. CLAIMS	3 -	7 =	0	x $200.00	$0.00
			FEE FOR AMENDMENT		$500.00
			FEE FOR EXTENSION OF TIME		$120.00
			TOTAL FEE FOR AMENDMENT AND EXTENSION OF TIME		$620.00

```
07/12/2007 TL0111   00000039 501133   10714849
01 FC:1251       120.00 DA
```

P2BLARGE/REV06

COMBINED AMENDMENT & PETITION FOR EXTENSION OF TIME UNDER 37 CFR 1.136(a) (Large Entity)	Docket No. 77580-042 (VRNK-1CP3CN)

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 $620.00

☒ 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

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:

Dated: 7/11/07

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

July 11, 2007 or by facsimile
    (Date)

_____
Signature of Person Mailing Correspondence

Cynthia Joseph

Typed or Printed Name of Person Mailing Correspondence

P28LARGE/REV06

PAGE 3/18 * RCVD AT 7/11/2007 5:19:28 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-1/14 * DNIS:2738300 * CSID:1 617 535 3800 * DURATION (mm-ss):04-52

PATENT APPLICATION FEE DETERMINATION RECORD
Effective December 8, 2004

Application or Docket Number

10/714849

CLAIMS AS FILED - PART I

	(Column 1)	(Column 2)
TOTAL CLAIMS		
FOR	NUMBER FILED	NUMBER EXTRA
TOTAL CHARGEABLE CLAIMS	minus 20=	*
INDEPENDENT CLAIMS	minus 3 =	*
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	OR	RATE	FEE
BASIC FEE		OR	BASIC FEE	
X$ 25=		OR	X$50=	
X100=		OR	X200=	
+180=		OR	+360=	
TOTAL		OR	TOTAL	

CLAIMS AS AMENDED - PART II

AMENDMENT A   7/11/07

	(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA
Total	50	Minus	** 27	= 23
Independent	7	Minus	*** 7	=
FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM				☐

SMALL ENTITY		OR	OTHER THAN SMALL ENTITY	
RATE	ADDITIONAL FEE		RATE	ADDITIONAL FEE
X$ 25=		OR	X$50=	1150
X100=		OR	X200=	
+180=		OR	+360=	
TOTAL ADDIT. FEE		OR	TOTAL ADDIT. FEE	1150

PD

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$ 25=		OR	X$50=	
	X100=		OR	X200=	
	+180=		OR	+360=	
	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$ 25=		OR	X$50=	
	X100=		OR	X200=	
	+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: 07/18/2007

ASMITH   SALE #00000001   Mailroom Dt: 07/11/2007   501133   10714849
                    01   FC : 1202   650.00 DA

SPW

STATEMENT UNDER 37 CFR 1.97(e) ACCOMPANYING INFORMATION DISCLOSURE STATEMENT	Docket No. 77580-042 (VRNK-1CP3CN)

In Re Application Of:　Victor Larson, et al.

Application No. 10/714,849	Filing Date November 18, 2003	Examiner Krisna Lim	Customer No. 23630	Group Art Unit 2153	Confirmation No. 3154

Invention:　AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS

JUL 16 2007

COMMISSIONER FOR PATENTS:

This is a statement under the provisions of 37 CFR 1.97(e) in the above-identified application.

Check applicable statement herebelow:

Statement Under 37 CFR 1.97(e)(1)

☒ Each item of information contained in the accompanying 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.

Statement Under 37 CFR 1.97(e)(2)

☐ No item of information contained in the accompanying Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the undersigned person, after making reasonable inquiry, no item of information contained in the accompanying 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.

_____
Signature

Dated: 7/12/07

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

Certificate of Mailing by First Class Mail

I 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 "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on

7-12- 07
_____
(Date)

_____
Signature of Person Mailing Correspondence

Cynthia Joseph
_____
Typed or Printed Name of Person Mailing Correspondence

cc:

P10C/REV04

INFORMATION DISCLOSURE CITATION *(Use several sheets if necessary)*		Docket Number (Optional) 77580-042 (VRNK-1CP3CN)		Application Number 10/714,849
		Applicant(s) Victor Larson, et al.		
JUL 16 2007		Filing Date November 18, 2003	Group Art Unit 2153	

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL	REF	DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
		6,557,037	04/29/2003	Provino	709	227	05/29/1998

U.S. PATENT APPLICATION PUBLICATIONS

*EXAMINER INITIAL	REF	DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	REF	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUBCLASS	Translation YES	NO

OTHER DOCUMENTS   *(Including Author, Title, Date, Pertinent Pages, Etc.)*

		Eastlake, D. E., "Domain Name System Security Extensions", Internet Draft, April 1998 (1998-04), XP002199931, Sections 1, 2.3 and 2.4.

EXAMINER	DATE CONSIDERED

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP Section 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Form PTO-A820
(also form PTO-1449)

P09A/REV05

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

SHEET 1 OF 1

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	1099823	(secure domain name service).ti,ab, clm. or DSN.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/17 07:53
L2	561599	l1 and (secure communication link).ti, ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/17 07:54
L3	36321	l2 and (authenticat$4 or cryptograph$3)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/17 07:54
L4	5	l3 and DSN.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/17 07:55
L5	386	l3 and DNS.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/17 07:56
L6	303	l5 and server.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/17 07:56
L7	6292	709/226, "221".ccls. or 713/201.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/17 07:57
L8	17	l7 and l6	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/17 07:57

L9	4	l8 and @ad<="19990607"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/09/17 08:00

McDermott
Will & Emery

RECEIVED
CENTRAL FAX CENTER

OCT 0 4 2007

FACSIMILE

Date: October 4, 2007   **Time Sent:**

To:	Company:	Facsimile No:	Telephone No:
Commissioner for Patents	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:	77580-042/5496	*Original to Follow by Mail:*	No
		Number of Pages, Including Cover:	2

Re: In re Application of: Victor Larson, et al.

Serial No.: 10/714,849

Filing Date: November 18, 2003

Title: An Agile Network Protocol For Secure Communications Using Secure Domain Names

Docket No.: 77580-042 (VRNK-1CP3CN)

Message:

Please enter the attached Status Inquiry.

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Victor Larson, et al.
Serial No: 10/714,849
Filing Date: November 18, 2003
Title: An Agile Network Protocol For Secure Communications
                    Using Secure Domain Names
Group Art Unit: 2153
Confirmation No.:   3154
Docket No: 77580-042 (VRNK-1CP3CN)

Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

STATUS INQUIRY

Applicants make a request as to the status of the above-identified application and for information as to when they might expect to receive an Office Action.

Respectfully submitted,

Toby H. Kusmer

Toby H. Kusmer, P.C.
Registration Number 26,418
McDermott Will & Emery LLP
28 State Street
Boston, Massachusetts 02109-1775
Telephone: (617) 535-4065
Facsimile: (617) 535-3800
e-mail: tkusmer@mwe.com

CERTIFICATE OF 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 and is addressed to: Commissioner For Patents, P. O. Box 1450, Alexander, VA 22313-1450 on the date indicated below.

Date: _October 4 2007_   _Cynthia Joseph_
                                           Cynthia Joseph

BST99 1553801-1.077580.0042

UNITED STATES PATENT AND TRADEMARK OFFICE

NOTICE OF ALLOWANCE AND FEE(S) DUE

22907	7590	10/29/2007

BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

EXAMINER
LIM, KRISNA

ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 10/29/2007

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/714,849	11/18/2003	Victor Larson	007170.00025	3154

TITLE OF INVENTION: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

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	01/29/2008

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: <u>Mail</u> | **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450
or <u>Fax</u> (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)

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.

| 22907 | 7590 | 10/29/2007 |

BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

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.
10/714,849	11/18/2003	Victor Larson	007170.00025	3154

TITLE OF INVENTION: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

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	01/29/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-226000

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. _____

This collection of information is required by 37 CFR 1.311. 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 take 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, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

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.

PTOL-85 (Rev. 08/07) Approved for use through 08/31/2010.   OMB 0651-0033   U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Petitioner RPX Corporation - Ex. 1002, p. 431

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.
10/714,849	11/18/2003	Victor Larson	007170.00025	3154

22907   7590   10/29/2007

BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

EXAMINER
LIM, KRISNA

ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 10/29/2007

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 663 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 663 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.

	Application No.	Applicant(s)
Notice of Allowability	10/714,849	LARSON 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 7/11/07*.

2. ☒ The allowed claim(s) is/are *2-12 and 28-53*.

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 claims 13-25.

Pursuant to 37 C.F.R 1.109 and M.P.E.P 1302.14, the following is an Examiner's Statement of Reasons for Allowance:

The prior arts of record do not teach or a domain name service system configured to be connected to a communication network, to store a plurality of domain names and corresponding network addresses, to receive a query for a network address, and to comprise an indication that the domain name service system supports establishing a secure communication link.

The examiner considers the applicants' claims 2-12 and 28-53 to be allowable based on the claim interpretation and the aforesaid prior arts of record.

Any comments considered necessary by applicant must be submitted no later than the payment of the Issue Fee and, to avoid processing delays, should preferably **accompany** the Issue Fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

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 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

September 16, 2007

KRISNA LIM
PRIMARY EXAMINER

		Docket Number (Optional)	Application Number
INFORMATION DISCLOSURE CITATION *(Use several sheets if necessary)*		77580-042 (VRNK-1CP3CN)	10/714,849
		Applicant(s) Victor Larson, et al.	
		Filing Date November 18, 2003	Group Art Unit 2153

JUL 16 2007

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL	REF	DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
KL		6,557,037	04/29/2003	Provino	709	227	05/29/1998

U.S. PATENT APPLICATION PUBLICATIONS

*EXAMINER INITIAL	REF	DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	REF	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUBCLASS	Translation YES	NO

OTHER DOCUMENTS *(Including Author, Title, Date, Pertinent Pages, Etc.)*

KL		Eastlake, D. E., "Domain Name System Security Extensions", Internet Draft, April 1998 (1998-04), XP002199931, Sections 1, 2.3 and 2.4.

EXAMINER *KRISNA LIM*	DATE CONSIDERED *9/16/07*

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP Section 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Issue Classification

‖‖‖‖‖‖‖‖‖‖‖‖‖‖‖ (barcode)

Application/Control No.	Applicant(s)/Patent under Reexamination
10/714,849	LARSON ET AL.

Examiner	Art Unit
Krisna Lim	2153

ISSUE CLASSIFICATION

ORIGINAL		INTERNATIONAL CLASSIFICATION								
CLASS	SUBCLASS	CLAIMED					NON-CLAIMED			
709	226	G	06	F	15	/173				/

CROSS REFERENCES									
CLASS	SUBCLASS (ONE SUBCLASS PER BLOCK)					/			/
709	221					/			/
713	201					/			/
						/			/
						/			/
						/			/

(Assistant Examiner) (Date)

KRISNA LIM
PRIMARY EXAMINER 9/7/1?

(Primary Examiner) (Date)

(Legal Instruments Examiner) (Date)

Total Claims Allowed: 37

O.G. Print Claim(s)	O.G. Print Fig.
1	1

☐ 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	15	31		61		91		121		151		181
3	2	16	32		62		92		122		152		182
4	3	17	33		63		93		123		153		183
6	4	18	34		64		94		124		154		184
7	5	19	35		65		95		125		155		185
5	6	20	36		66		96		126		156		186
8	7	21	37		67		97		127		157		187
13	8	22	38		68		98		128		158		188
9	9	23	39		69		99		129		159		189
10	10	24	40		70		100		130		160		190
11	11	25	41		71		101		131		161		191
12	12	26	42		72		102		132		162		192
	13	27	43		73		103		133		163		193
	14	28	44		74		104		134		164		194
	15	29	45		75		105		135		165		195
	16	30	46		76		106		136		166		196
	17	31	47		77		107		137		167		197
	18	32	48		78		108		138		168		198
	19	33	49		79		109		139		169		199
	20	34	50		80		110		140		170		200
	21	35	51		81		111		141		171		201
	22	36	52		82		112		142		172		202
	23	37	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
	27		57		87		117		147		177		207
1	28		58		88		118		148		178		208
2	29		59		89		119		149		179		209
14	30		60		90		120		150		180		210

Index of Claims

Application/Control No.	Applicant(s)/Patent under Reexamination	
10/714,849	LARSON ET AL.	
Examiner	Art Unit	
Krisna Lim	2153	

Symbol	Meaning	Symbol	Meaning	Symbol	Meaning	Symbol	Meaning
√	Rejected	–	(Through numeral) Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

Final	Original	9/16/07		Final	Original	9/16/07		Final	Original
	1			35	51	=			101
3	2	=		36	52	=			102
4	3	=		37	53	=			103
6	4	=			54				104
7	5	=			55				105
5	6	=			56				106
8	7	=			57				107
13	8	=			58				108
9	9	=			59				109
10	10	=			60				110
11	11	=			61				111
12	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
1	28	=			78				128
2	29	=			79				129
14	30	=			80				130
15	31	=			81				131
16	32	=			82				132
17	33	=			83				133
18	34	=			84				134
19	35	=			85				135
20	36	=			86				136
21	37	=			87				137
22	38	=			88				138
23	39	=			89				139
24	40	=			90				140
25	41	=			91				141
26	42	=			92				142
27	43	=			93				143
28	44	=			94				144
29	45	=			95				145
30	46	=			96				146
31	47	=			97				147
32	48	=			98				148
33	49	=			99				149
34	50	=			100				150

Search Notes

Application/Control No. 10/714,849	**Applicant(s)/Patent under Reexamination** LARSON ET AL.
Examiner Krisna Lim	**Art Unit** 2153

SEARCHED

Class	Subclass	Date	Examiner
709	226, 221	9/16/2007	KL
713	201	9/16/2007	KL

INTERFERENCE SEARCHED

Class	Subclass	Date	Examiner
709	**226, 221**	9/16/2007	KL
713	201	9/16/2007	KL

SEARCH NOTES (INCLUDING SEARCH STRATEGY)

	DATE	EXMR
EAST	9/16/2007	KL
Google	9/16/2007	KL

Docket No.: 077580-004♦   **PATENT**

11-09-07

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant :   Victor Larson et al.

Appl. No. :   10/714,849

Filed   :   November 18, 2003

Title   :   AN AGILE NETWORK PROTOCOL
             FOR SECURE COMMUNICATIONS
             USING SECURE DOMAIN NAMES

Grp./A.U. :   2153

Examiner: :   LIM, Krisna

Customer No.: 23,630

Confirmation No.: 3154

CERTIFICATE OF MAILING (37 CFR. § 1.10)

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as "Express Mail Post Office to Addressee" under 37 CFR 1.10 in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on **November 8, 2007.**

Express Mail Mailing Label: EV 942455095 US

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.

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

11/13/2007 THGUYEN2 00000002 501133   10714849

01 FC:1806   180.00 DA

BST99 1556883-1.077580.0042

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 8 , 2007

**Please recognize our Customer No. 23630
as our correspondence address.**

BST99 1556883-1.077580.0042

10/714,849

		INFORMATION DISCLOSURE CITATION IN AN APPLICATION	ATTY. DOCKET NO. 077580-0042		SERIAL NO. 10/714,849
			APPLICANT Larson et al.		
		(PTO-1449)	FILING DATE Nov. 18, 2003		GROUP 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
	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.

3

INFORMATION DISCLOSURE CITATION IN AN APPLICATION	ATTY. DOCKET NO. **077580-0042**	SERIAL NO. **10/714,849**
	APPLICANT **Larson et al.**	
(PTO-1449)	FILING DATE **Nov. 18, 2003**	GROUP **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
	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.

4

10/714,849

INFORMATION DISCLOSURE CITATION IN AN APPLICATION		ATTY. DOCKET NO. **077580-0042**	SERIAL NO. **10/714,849**
		APPLICANT **Larson et al.**	
(PTO-1449)		FILING DATE **Nov. 18, 2003**	GROUP **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
	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 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

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.

5

BST99 1556883-1.077580.0042

10/714,849

INFORMATION DISCLOSURE CITATION IN AN APPLICATION	ATTY. DOCKET NO. 077580-0042	SERIAL NO. 10/714,849
	APPLICANT Larson et al.	
(PTO-1449)	FILING DATE Nov. 18, 2003	GROUP 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

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
	B1	EP 836306A1	4/15/1998	Sasyan et al.			
	B2	WO 00/17775	3/30/2000	Miller et al.			

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	RFC 2401 (dated Nov. 1998) Security Architecture for the Internet Protocol (RTP)	
	C2	RFC 2543-SIP (dated March 1999): Session Initiation Protocol (SIP or SIPS)	
	C3	Search Report, IPER (dataed Nov. 13, 2002), International Applicatoin No. PCT/US01/04340.	
	C4	Search Report, IPER (dated Feb. 06, 2002), International Application No. PCT/US01/13261.	
	C5	Search Report, IPER (dated Jan. 14, 2003), International Application No. PCT/US01/13260.	
	C6	Shankar, A.U. "A verified sliding window protocol with variable flow control". Proceedings of ACM SIGCOMM conferece on Communications architectures & protocols. pp. 84-91, ACM Press, NY,NY 1986.	
	C7	W. Stallings, "Cryptography And Network Security", 2nd, Edition, Chapter 13, IP Security, Jun. 8, 1998, pp. 399–440.	
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.

6

(19)

Europäisches Patentamt

European Patent Office

Office européen des brevets

(11) **EP 0 836 306 A1**

(12) **EUROPEAN PATENT APPLICATION**

(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)

(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 ($I_{M(j)}$,$I_{T(j)}$) 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

Description

<u>Field of the Invention</u>

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.

<u>Background of the Invention</u>

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 ordi-

narily 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.
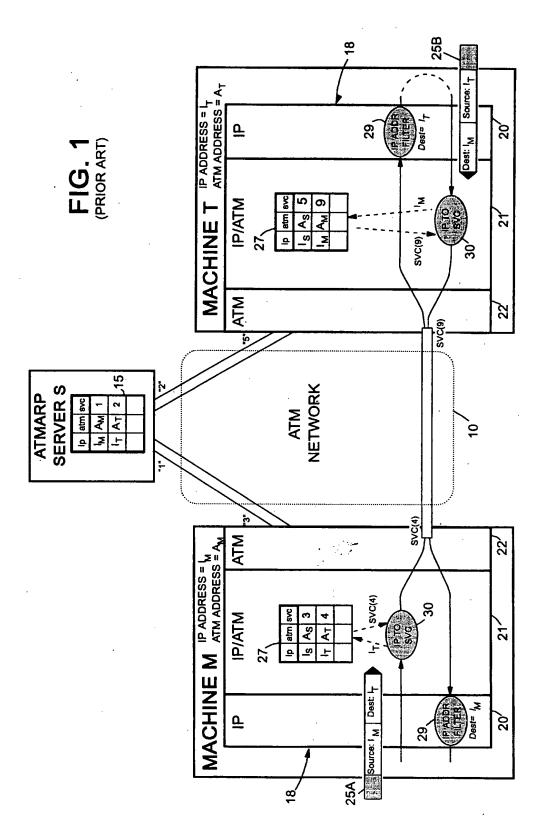
11. A system according to any one of claims 1 to 9, wherein said high-level addresses are MAC addresses and said network is a emulated LAN over an ATM network.

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.
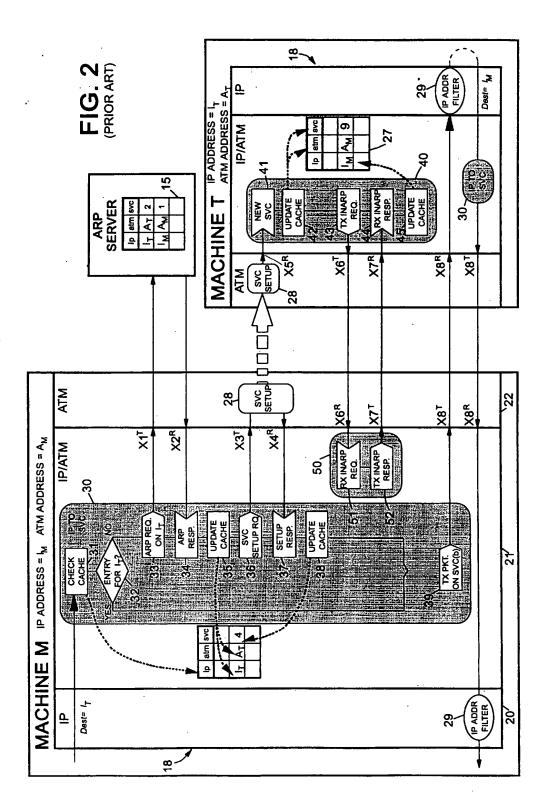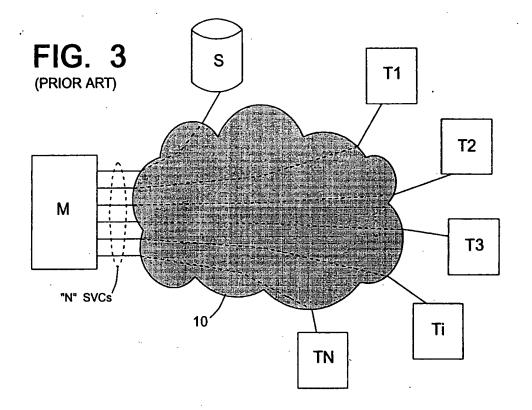
FIG. 1
(PRIOR ART)

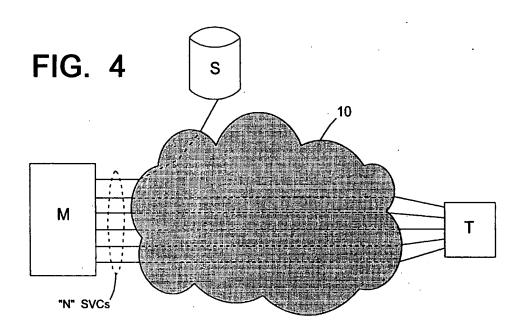FIG. 2
(PRIOR ART)

FIG. 3
(PRIOR ART)

FIG. 4

FIG. 5

EP 0 836 306 A1

FIG. 6

FIG. 7
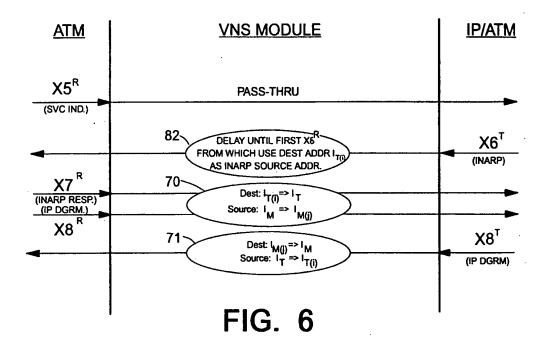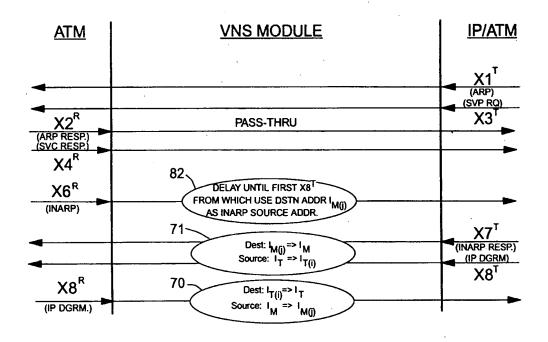
14
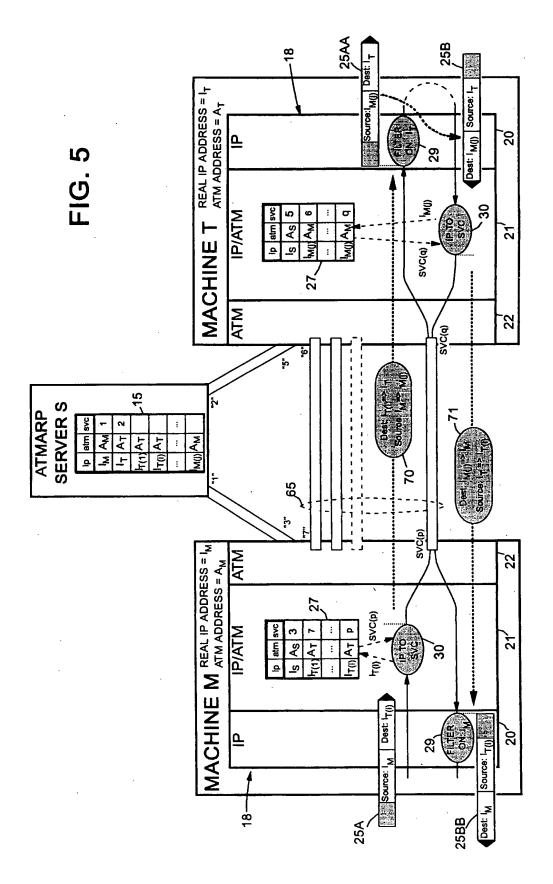
Petitioner RPX Corporation - Ex. 1002, p. 459

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

EPO FORM 1503 03.82 (P04C01)

B2

PCT

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(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.

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

1

1   An analysis of the problem, however, indicated that estimating the relativities ·

2   was not essential to facilitate trading, or, in a broader sense, that trading was the only

3   way to address the problem of insuring concentrated risk. The key difficulty was

4   determining how to create greater efficiency in the reinsurance market, whether by

5   introducing new instruments (like swaps), bringing new capital to the market,

6   connecting more buyers to more traders, or reducing the cost of placing reinsurance.

7   It was determined that the above concept could be implemented in an electronic

8   trading system that could play an important role in promoting these factors, and

9   could, in fact, transform the reinsurance market, which is not very automated. A

10   system that allowed trading was developed and implemented. A more detailed

11   description of this system, as enhanced in accordance with various inventive

12   principles herein (referred to as "first-generation" complex instrument trading

13   technology), are provided below.   More generally, as electronic commerce (and

14   business-to-business commerce, in particular) has grown, various companies have

15   developed software tools and services to facilitate transactions on the Internet and

16   over private networks. E-Bay, for example, hosts a well-known web site that

17   operates a transaction model (a so-called "concurrent auction") that permits buyers

18   to submit bids on items offered by individuals. Lotus Notes provides a network-

19   oriented system that allows users within a company to collaborate on projects.

20   Oracle Corporation hosts various transaction engines for clients that pay to host such

21   services on a web site. DIGEX Corporation similarly hosts web-based application

22   programs including various transaction engines. Other companies sell so-called

23   "shrink wrap" software that allows individuals to set up web sites that provide

24   catalog ordering facilities and the like.

25   Some Internet service providers, such as America Online, host "chat rooms"

26   that permit members to hold private discussions with other members who enter

27   various rooms associated with predetermined topics. A company known as

28   blueonline.com hosts a web site that facilitates collaboration on construction projects.

29   Various virtual private networks have been created to facilitate communication

30   among computer users across the Internet and other networks, but these networks

31   provided very limited functionality (e.g., e-mail services); are not user-defined (they

32   must be created and installed by system administrators); and they cannot be easily

33   destroyed when they are no longer needed.

2

SUBSTITUTE SHEET (RULE 26)

1   The aforementioned products and services are generally not well suited to
2   facilitating complex electronic transactions. As one example, most conventional
3   services are predefined (not user-defined) and are centrally administered. Thus, for
4   example, a group of companies desiring to collaborate on a project must fit their
5   collaboration into one of the environment models provided by an existing service
6   provider (or, alternatively, build a custom system at great expense).

7   Suppose, for example, that a group of high school students needs to
8   collaborate on a research paper that requires soliciting volunteers for a survey on drug
9   use, conducting the survey, brainstorming on the survey results, posing follow-up
10   questions to survey participants anonymously, publishing a report summarizing the
11   results, and advertising the report for sale to newspapers and radio stations. This
12   project requires elements of communication among persons inside a defined group
13   (those writing the paper) and outside the group (e.g., survey participants); conducting
14   research (conducting the survey, compiling the results, comparing the results with
15   other surveys published by news sources; and brainstorming on the meaning of the
16   results); and conducting a commercial transaction (e.g., publishing the survey in
17   electronic form and making it available at a price to those who might be interested
18   in the results). No existing software product or service is available to meet the
19   specific needs of this research team. Creating a user-defined environment including
20   tools and communication facilities to perform such a task would be prohibitively
21   expensive. Even if such a tailor-made environment could be created, it would be
22   difficult to disassemble the environment (computers, networks, and software) after
23   the project was completed.

24   In short, there is a need to provide a user-defined collaborative environment
25   that is tailored to the needs of particular groups that conduct communication,
26   research, electronic transactions, and deal-making.

27   **SUMMARY OF THE INVENTION**

28   A first embodiment of the invention, referred to as a complex instrument
29   trading engine (CITE), facilitates negotiation between two or more parties. In this
30   embodiment, a set of negotiation tools and techniques such as anonymous email,
31   secure communication, document retention, and bid and proposal listing services are
32   provided in order to facilitate the negotiation and execution of complex instruments
33   such as contracts between corporations, governments, and individuals.

3

SUBSTITUTE SHEET (RULE 26)

1   A second embodiment of the invention, referred to as a dynamic collaborative

2   environment (DCE), allows members of a group to define a dynamic virtual private

3   network (DVPN) environment including user-selected tools that facilitate

4   communication, research, analysis, and electronic transactions both within the group

5   and outside the group. The environment can be destroyed easily when it is no longer

6   needed. Multiple environments can co-exist on the same physical network of

7   computers.

8   Although the two embodiments are described separately for ease of

9   comprehension, it should be understood that the two embodiments share many

10 features and, in fact, the second embodiment could include some or all of the features

11 of the first embodiment in a generalized collaborative system. Consequently,

12 references to a specific embodiment in the following description should not be

13 deemed to limit the scope of features or tools included in each embodiment.

14 Moreover, references to specific applications, such as the reinsurance industry,

15 should not be deemed to limit the application of the invention to any particular field.

16 **BRIEF DESCRIPTION OF THE DRAWINGS**

17   FIG. 1A shows a four-step model of deal making including meeting, analysis,

18 negotiation, and closing the deal.

19   FIG. 1B shows contract formation among a group of parties to a contract.

20   FIG. 2 shows a listing display system showing all offers for contracts and

21 responses thereto.

22   FIG. 3 shows details of a listing that has been selected by a user.

23   FIG. 4 shows one possible implementation of a reply card definition screen.

24   FIG. 5 shows one possible implementation of a document management

25 screen.

26   FIG. 6 shows one possible implementation of a screen indicating persons

27 having access to a shared folder.

28   FIG. 7 shows a list of consummated deals in the system.

29   FIG. 8A shows detailed information regarding a completed trade.

30   FIG. 8B shows a deal summary including structured and unstructured

31 information concerning the deal.

32   FIG. 9 shows a "flip widget" in a first state.

33   FIG. 10 shows a "flip widget" in a second state.

<div align="center">4</div>

1   FIG. 9A shows a more detailed example of a "flip widget" in a first state.

2   FIG. 10A shows a more detailed example of a "flip widget" in a second state.

3   FIG. 11 shows method steps that can be carried out to define, create, and

4   destroy an environment according to a second embodiment of the invention.

5   FIG. 12 shows one possible system architecture in which various principles

6   of the invention can be implemented.

7   FIGS. 13A through 13C show one possible user interface for creating a group

8   and identifying group members.

9   FIG. 14A shows one possible user interface for selecting group members from

10 one or more lists.

11   FIG. 14B shows one possible user interface for selecting group members by

12 composing invitations.

13   FIG. 14C shows one possible user interface for selecting group members by

14 composing an advertisement.

15   FIG. 15 shows a banner advertisement 1501 displayed on a web site, wherein

16 the banner advertisement solicits participation in a group.

17   FIG. 16 shows one possible user interface for selecting communication tools

18 to be made available to group members.

19   FIG. 17 shows one possible user interface for selecting research tools to be

20 made available to group members.

21   FIG. 18 shows one possible user interface for selecting transaction engines

22 to be made available to group members.

23   FIG. 19 shows one possible user interface for selecting participation engines

24 to be made available to group members.

25   FIG. 20A shows an authentication screen for group members to gain access

26 to a newly created environment.

27   FIG. 20B shows a web page generated for a specific user-defined

28 environment, including tools available to group members having access to the

29 environment.

30   FIG. 21 shows one possible method of generating environments in accordance

31 with various aspects of the present invention.

32   FIG. 22 shows one possible data storage arrangement for storing and

33 manipulating brain writing cards.

5

1   __DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS__

2   A. __COMPLEX INSTRUMENT TRADING ENGINE EMBODIMENT__

3   A first embodiment of the present invention provides a second-generation

4   version of a complex instrument trading system. The second-generation system

5   includes specialized tools that were not included in the first version of the prior art

6   CATEX insurance trading system described above.   These tools represent a

7   substantial improvement over the first generation and incorporate new concepts of

8   communications in a trading environment, and other capabilities that did not exist in

9   the first generation technology. In addition, it is believed that many of these tools are

10   also applicable to software systems other than the Complex Instrument Trading

11   Engine or Negotiating System (CITE) described herein.   Thus, the inventive

12   principles are not limited to trading systems for complex instruments, nor even to

13   trading systems in general.

14   Primarily, the tools described herein ameliorate certain difficulties associated

15   with trading of complex instruments. Complex instruments are instruments where

16   there is more than one dimension for negotiation. As compared to such instruments

17   as securities, complex instrument transactions take longer to research and

18   consummate and require more extensive documentation. For example, stock trading

19   employs a simple instrument (a share) and negotiation focuses on one dimension

20   (price) while insurance contracts have many dimensions (term, price, coverage,

21   definitions of perils, etc.). The stock market is relatively simple to automate -- as

22   soon as bid and asked prices match, the deal is concluded in an instant according to

23   the rules of the exchange. Automation of complex trading is much more difficult,

24   since the parties must negotiate and reach agreement on multiple dimensions and

25   document that agreement using an instrument specific to the precise agreement.

26   Automation of complex instrument trading is more difficult in every way than trading

27   simple instruments.

28   The trading model behind the Complex Instrument Trading Engine or

29   Negotiating System is built around a simple, four-step model of deal making.

30   Referring to FIG. 1A, the steps are as follows:

31   1. __Meeting__: Potential buyers connect with potential sellers with reciprocal

32   interests. This connection does not mean that a deal will necessarily be concluded but

33   simply that the two parties have some basis for continuing discussion. In simple

1 instrument trading, it is typically only necessary to advertise quantity and price

2 offered or sought. Offers for complex instruments must include substantially more

3 detail and (frequently) extensive attachments or exhibits.

4   2. <u>Research/Analysis</u>: Each company considers its own position and/or offer

5 and the counter party's position. Using information and analytic tools from various

6 sources, including internal resources and resources provided by or through the trading

7 system, each party does research and refines its position. The multiple dimensions

8 of complex instruments increases the analytical complexity and limits the value of

9 a simple market price. As indicated by the arrows in FIG. 1, this step is usually

10 performed iteratively with the negotiation.

11   3. <u>Negotiation:</u> Parties to the negotiation speak directly and exchange

12 whatever information is necessary to advance the deal. As indicated by the arrows

13 in FIG. 1A, this step is usually performed iteratively with the research step.

14   4. <u>Close</u>: the companies negotiate and sign an instrument that documents the

15 deal. This can be a complete and detailed contract, or it may be a simple

16 memorandum. In simple instrument trading, the actual trade agreement is often

17 standardized by the exchange. In complex instrument trading, the agreement must

18 be more specific to the deal, though it is possible to use such tools and fill-in-the

19 blank forms.

20   Within a system using these complex instrument tools, trading parties can

21 place offers to buy, sell, or trade in a public area, and examine such offers ("listings")

22 posted by others. Using advanced communications tools the parties can conduct

23 initial discussions to determine if a placement is possible. Using tools described

24 herein, the initial contact can be done anonymously.

25   If a deal seems possible, the system preferably provides access to the

26 extensive information necessary to assess the possible deal. This can include static

27 information (e.g. reports or data) maintained within the system, links to information

28 providers outside the system, online analytical tools, and links to providers of

29 analytical services.

30   For complex instruments, the process of negotiating a deal is contemplated

31 to be an iterative one, with successive stages of analysis and discussion. The need

32 for extensive communication is one of the critical distinctions between trading of

33 simple instruments (e.g. retail sale) and complex instruments. Complex instrument

7

1 trading requires dialog and more -- exchange of documents (often voluminous),

2 consultation with counsel and intermediaries, conferencing, and working together on

3 the final agreement. For electronic commerce to have an impact in complex

4 instrument trading, it must support and facilitate this communication, and not force

5 traders to fall back on methods and technology outside the electronic trading

6 environment.

7 The final step is closing the deal. The companies can negotiate a contract

8 online. Tools provide sample, fill-in the blank contracts and memoranda of

9 understanding as a starting point. Negotiators can begin with these, or they can use

10 one of their own. Collaborative software makes it possible to display text

11 simultaneously on each negotiator's screen and to work on the language together.

12 When the contract is final, the system allows for secure, online signature, though

13 companies not comfortable with electronic signature for very large deals may print

14 a hard copy and sign it conventionally.

15 By creating electronic exchanges for complex instrument trading, the CITE

16 tools can have a fundamental and positive impact on many areas of commerce:

17 1. An electronic exchange makes it possible to put an offer in front of more

18 people more quickly than could be informed through direct contact, even allowing

19 for active intermediaries or brokers.

20 2. Traders can advertise and conclude deals without the need for an

21 intermediary when they have adequate support or internal resources.

22 3. Through better communications, wider exposure for offers, and the first

23 steps towards standard contract language, electronic trading of complex instruments

24 can substantially reduces transaction costs.

25 4. With lower transaction costs, it is possible to conclude deals that were not

26 possible with higher overhead.

27 5. Through the immediate posting of the results of trades, pricing is moved

28 towards a market basis, reducing research and analysis costs enormously. This

29 speeds placement.

30 6. Smaller exposure means lower risk, and market pricing is an adequate

31 surrogate for analytically derived pricing in some circumstances. Together these

32 factors make it possible for traders to participate in markets or market segments in

33 which they would not normally do business.

8

SUBSTITUTE SHEET (RULE 26)

1   7. By making it possible for all companies, large and small, to talk directly

2   to each other, electronic trading of complex instruments can lead to the

3   democratization of the marketplace increasing competition.

4   Overall, electronic trading of complex instruments has the potential to

5   improve the efficiency of markets enormously, and to establish markets in areas of

6   commerce that are currently done through intermediaries or on a one-on-one basis.

7   The trading tools described herein are designed to facilitate electronic trading of

8   complex instruments. The first-generation complex instrument trading tools broke

9   new ground in the extension of electronic commerce into new and more complicated

10   markets. The table below summarizes the areas of new and improved technology,

11   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

9

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

1

2   Referring to FIG. 1B, one aspect of the system within the framework of the

10

SUBSTITUTE SHEET (RULE 26)

1   negotiation/analysis loop shown in FIG. 1, is the ability to define one or more ·

2   contracts, for example, in the parlance of the reinsurance trade, "slip sheets." Various

3   members of a group of authorities modify the contract causing it gradually to take a

4   final form that is either rejected as untenable or accepted as a finalized deal. The

5   system exposes various aspects of the contract and attendant documents to the

6   appropriate participants in the transaction, also providing each with a level of

7   authority to add, delete, or modify documents as well as the evolving contract or

8   contracts (assuming there may be various contract templates being discussed). These

9   filters (filter 1 through filter 4, for example), as shown in FIG. 1B, determine the

10   authority of the party (Party 1-Party 4) to modify or see the data object, whether it is

11   a document or a slip sheet. The system combines this system of filters with signature

12   technology for closing the deal; that is, implementing signatures so that an

13   enforceable contract is generated.

14   A deal is like any other data object and once it is defined and entered, it

15   cannot be modified. Elements of the deal can be "signed" such as documents

16   attached to a contract (for example, Contract 1 has documents D1 and D2 attached

17   to (combined with) it. Together these elements, the contract and the attachments,

18   define the deal. Also, the entire deal 245 can be signed using a signature device

19   ("widget") S8. Other documents may relate to a deal but not be attached. These can

20   be viewed using a document manager described further below.

21   Listing System

22   Referring to FIG. 2, a listing screen displays all offers for contracts, for

23   example offer 314, as well as responses to them, for example, response 313. The

24   parameters of the offers and responses to them are shown in columns, the heading of

25   each of which may be selected to sort the listings by that heading, for example

26   heading 315 if clicked would sort by the unique index number for the listing. Notice

27   that the responses (for example, response 313) are shown indented to indicate a series

28   of elements of a dialogue-thread. As indicated, the responses have a "daughter"

29   relationship to the parent listings. That is, listing 314 is a parent and reply 313 is a

30   daughter. The daughters remain in their hierarchical position beneath the parent

31   despite sorting by the column headings. This makes the tabular sort scheme

32   compatible with a threaded display, which is useful to show dialogues.

33   Referring now also to FIG. 3, when a user invokes a display of the details of

11

1   a listing by clicking on an index hyperlink 312 to show the details of the listing, a

2   user interface element displays the lister's defined parameters of the listing. As

3   shown, various parameters are displayed, many of which are hyperlinked. For

4   example, attachments 304 may be selected to display the corresponding attachments.

5   A detailed description 301 may be provided as well as specific instructions for

6   responding 302. A reply button 303 permits the user to reply. Activating the reply

7   button 303 will either invoke a standard public reply screen which creates a new

8   listing similar to the parent listing or a special reply defined by a reply card which is

9   further described below.

10   A reply to a listing can take the form of a public reply that invokes a screen

11   substantially the same as FIG. 3 but with blank spots for entry of reply information.

12   A more useful kind of response element is a reply card that can be defined by the

13   lister. This is because in negotiations on complex transactions such as reinsurance

14   contracts and, for example, pollution emission allowances, the parties with whom a

15   lister would be willing to trade are limited in terms of certain criteria. These criteria

16   will vary from one type of transaction to another.

17   In an active trading system, the number of listings can quickly grow to a large

18   number and quickly exceed the number which can conveniently be displayed in a

19   single table. Several capabilities are built into the system to address this problem.

20   First, by default, listings are presented in order from newest to oldest. Second, the

21   sort capabilities previously described allow users to modify the standard order.

22   Third, the total market may be divided into subcategories. In the area of insurance

23   catastrophe risk, these could include categories for different lines of insurance (e.g.

24   marine, aviation, commercial buildings). Fourth, users may enter search criteria to

25   identify a subset of listings of particular interest.

26   Searching listings: A user may enter a keyword such as "hurricane" to

27   identify all listings that contain that word in the title, description, and (optionally)

28   attachments. To improve the reliability of the search, users are provided access to

29   a standard lexicon when composing a listing. In the first embodiment, this capability

30   is invoked by pressing the right mouse button while the cursor is any field of the

31   listing. A list of common terms is displayed. The user can select the term of

32   interest, which is then placed into the text of the listing at the insertion point marked

33   by the cursor. For example, a listing for insurance risk would typically include a

<center>12</center>

<center>**SUBSTITUTE SHEET (RULE 26)**</center>

1   field for geographic scope (i.e. the location of the properties to be insured). When

2   in this field, the lexicon displayed would include terms such as "California" and

3   "Coastal Florida". Choosing a term from the lexicon insures uniformity of

4   terminology across listings and between the search engine and the listings.

5   "California" will be used rather than a mix of "Ca", "CA", "Calif", etc. The search

6   is further improved by symantic indexing. Essentially, this means that synonymous

7   terms are grouped, so that searches for one will find the other. A person who

8   searches for "California" will get listings for "Los Angeles" that do not include the

9   word "California".

10   The search engine can include an agent capability. This agent capability

11   offers the user the option of saving a search, after the user reviews the results and

12   deems them acceptable. This search is retained in a library of searches along with the

13   email address of the owner of the agent. The search is retained in the library until

14   is it either deleted by the user when it is no longer needed or automatically deleted

15   in a cleanup of searches older than a certain date. Whenever a new listing is placed

16   on the system, all of the saved searches are executed. If the new listing meets any

17   of the search criteria, a message is sent to the owner of that criterion via email or

18   instant messaging.

19   A model was developed to allow a lister to define a set of criteria and request

20   a set of information from any respondents in the form of an anonymous reply "card."

21   The card defines a set of requested information which may be packaged as a

22   document object and placed in the document manager system and connected with

23   each listing. A user would download the reply card and fill the card out and send it

24   back to the posting party.

25   A document object, called a reply card, is made available to a respondent

26   through the document manager. The respondent is permitted to retain his anonymity

27   as is the lister. Each may communicate with the other through an Amail system

28   described in more detail below. The respondent supplies the requested information

29   and sends the data to the lister. A system in the listing manager allows a lister to

30   define a reply card having any particular fields and instructions required of a

31   respondent. Some of the information required may be obtained automatically from

32   a set of default data stored on the respondent's computer.

33   Referring to FIG. 4, a reply card definition screen is invoked to define the

13

1   parameters of a new listing. The new listing is defined using a user-interface element.

2   looking much like FIG. 3. While the details are not critical, the definition of reply

3   card involves, in essence, the definition of a user-interface control such as a dialog

4   with radio buttons, text boxes, etc.   These are definable for server-side

5   implementation through HTML and are well known so the details are not discussed

6   here. The lister defines a set of controls that allow the entry by a replying party of

7   the information that the lister requires. The reply card is stored as any other

8   information object and may be organized and accessed through the document

9   manager described below. FIG. 4 shows a simple example of a format of a reply

10 card.

11   A reply card is created by a user when posting a new listing. The lister

12 specifies the information that must be included in a response, and the type of

13 information object to display for the data element (e.g. a text box, check box, radio

14 button). The system then creates an HTML page to collect the requested information.

15 When a respondent clicks "Reply Card" on the listing screen, the page is displayed.

16 All of the responses are automatically entered into a database created automatically

17 when the reply card is composed. As each respondent fills out a reply card, a new

18 record is added to the database of the system and the lister is permitted to view it

19 through an appropriate filter as discussed above.

20 Signature System

21   As business is increasingly done in an electronic environment, electronic

22 signature and approval is becoming more critical. The typical electronic signature

23 model has focused on two aspects:

24   1. Electronic validation of the user -- specifically determining that the person

25   viewing a document on line is the authorized signatory; and

26   2. Validating the document being signed by a means that either prevents

27   modification of a document or will reveal whether changes have been made.

28   Methods for validation of identity range from simple personal identification

29 numbers or passwords, to electronic signature pads, and more advanced methods of

30 biogenic validation such as fingerprint or retinal patterns. Methods for document

31 validation range from simple archiving of one or more copies in a read-only model

32 or inaccessible location to methods based on mathematical algorithms that create a

33 characteristic number or alphanumeric string for a document. These strings are

14

1   termed "electronic signatures." Changes to the document change the electronic .

2   signatures. Because the signatures are much shorter than the documents, very many

3   documents have precisely the same signature, but the algorithms to calculate the

4   signature are very difficult to invert, so that it is effectively impossible to deduce a

5   meaningful change to a document that will preserve a specific signature.

6   These two aspects of electronic signature are highly developed, but there has

7   been little analysis or development of the general process by which documents can

8   be signed.

9   The invention allows for secure and reliable routing of documents, for which

10   signatures are required, to a specified list of signatories. Unlike prior art systems,

11   such as ordering or accounts payable systems which have highly structured signature

12   procedures tailored to a specific process, the present invention provides a flexible

13   method and system that allows a signature-type of authority/requirement to be

14   attached any kind of information object. The method is sufficiently abstract, flexible,

15   and general that it can be applied in many contexts aside from the CITE embodiment

16   described in the present specification.

17   One signature method/device employs the following steps:

18   1.   Registration of signatories – This process provides a register of identifiers

19   indicating entities with signatory authority and correlates these identifiers with the

20   information objects for which the signatory authority is applicable. The same register

21   may also be used to identify other types of authority in the system in which the

22   signature device is implemented.   For example, document read authority,

23   modification authority, exclusive access to documents, etc. may also be provided in

24   the same register. Signature registration may be provided automatically in certain

25   systems where registration of, for example, read/write authority is provided since any

26   entity with signatory authority would in almost all instances, also be provided with

27   some other kind of authority, most notably, read authority. Thus, where the signatory

28   system is embedded in certain kinds of systems, it may be that no particular

29   additional method or device is required to implement signatory registration since an

30   existing register may already exist or be required for other purposes.

31   Registration information includes the general categories of information listed

32   below. Definitions of specific fields within these categories are a function of the

33   specific implementation of the signature system or the parent system. The following

15

1   are exemplary:

2   1. Identity – unique identifier of the entity, the organization(s) with which the

3   entity is affiliated, other relevant information.

4   2. Contact information – information indicating how the entity can be

5   reached, how documents and mail messages can be routed to the entity.

6   3. Security Information – a password for each class of signature as described

7   further below.

8   2.   Classes of signatures – The device/method provides a variety of classes of

9   signature, each associated with a unique level of approval or level of commitment.

10   For example, a class of signature-authority can be defined that represents

11   individuals, for example, with authority to sign contracts only below a set amount,

12   or for expenses relating only to one department of an organization, or within certain

13   time constraints, etc. The signatory system maintains this taxonomy of possible

14   signature types in a database with a unique identifier for each level of authority

15   defined. The system allows the creation and deletion of classes. Each class is

16   preferably permitted to be named and a descriptive definition attached to each class.

17   3.   Defining a Set of Signatures – Using an appropriate user interface element, the

18   user of the system selects an information object (for example, a document, file, or

19   collection of such objects) requiring signature(s). The entity originating the signature

20   process then identifies the entity or entities required to sign the object. The

21   specification of the signers can proceed either by the selection of individuals from a

22   list supported by the above defined entity register. Alternatively, in an environment

23   where individuals are strongly bound to organizations, for example, it can proceed

24   by selecting the list of organizations that will sign and, within each organization, the

25   person who will sign. The list is built by a series of selections. After each selection

26   from the list, the user indicates his/her desire to add the selected individual to a list

27   of required signatories. The user interfaces provides for entries in which all the

28   selected signatories are required or only one of the selected signatories are required.

29   For example, if more than one entity is selected from the list prior to the

30   selection (e.g., clicking an "Add" button), the system may require a signature from

31   any of the people selected, but not all of them. To require signature from every

32   member of the group, the initiator may select one person, then "add", select the

33   second, then "add", and so on. Thus, adding a group with one "add" command would

1   provide an "any signature will suffice" list and adding members individually would

2   require a signature from that individual or entity. Note that this technique may also

3   be used to define combinations of required and "any of" groups.

4   For each signer or group of signers selected in a single "add" command, the

5   initiator of the signing sequence must specify the class of signature associated with

6   the person for the document being signed. This may be selected from a list of

7   signature classes (see item 2). If the specific implementation of the signature process

8   only supports one class of signature, the selection of class may be omitted.

9   4. Random or Serial Order of Signature – After or concurrent with the creation of a

10   signature list, the initiator specifies whether signatures must be in order or if a

11   specific order is not required. For purposes of defining the order of signature,

12   individuals who are selected as a group are considered as occupying a single place

13   in the sequence.

14   5. Document Authentication – Upon initiating a signature sequence, the information

15   object is authenticated by means of a secure hash algorithm. The specific hashing

16   algorithm is a matter of design choice or may made dependent on a user's choice.

17   There are several possible hash algorithms available in the public domain. The

18   electronic signature produced by the secure hash algorithm is archived with the

19   information object in a secure repository. If the information object is, for example,

20   a record in a database, the contents of the record are copied to a file in delimited

21   format for archival purposes. If the object is a table, the table is exported prior to

22   archive.

23   6. Document Routing – Upon initiation of a signature sequence, the initiator

24   specifies how the signatories are to be informed. The options are:

25   • No notification from the signature system

26   • Email message

27   • Email message with attachment of the information object.

28   • Posting on a signature web site

29   The system accepts and implements the chosen method, which may be connected to

30   the signature or a single choice applied to all signatories. Alternatively, the method

31   of notification may be stored with the signature class definitions. In a signature

32   process with no required order, e-mail notice may be sent simultaneously to all of the

17

1   designated individuals at the time of initiation. If the process is serial, only the first

2   person may be notified. The electronic signature of the information object may be

3   included in an e-mail message.

4   7. <u>Accessing the signature system</u> – The signature system can be implemented for

5   access via a web browser or database client-server software across the Internet, an

6   intranet, a LAN, or a WAN. Access to the system will typically require a password,

7   but this may not be necessary on a secure network. Upon access to the system a user

8   will have the option to display a list of all of the information objects which he or she

9   has signed or is being asked to sign. For each object, the display can include the

10 following information:

11   • Object name

12   • Description of object (text, mime, size, date)

13   • List of scheduled signatories

14   • Date each person signed

15   • Class of signature for each person

16   • Electronic signature produced by the secure hash algorithm

17 If the object is available (viewable) on line, the display may also include a link to

18 display or download the object.

19 8. <u>Validation of the Object at Time of Signature</u> – If the user downloads or views the

20 object, the system will execute the secure hash algorithm to calculate the electronic

21 signature. This will be displayed so that the potential signer can compare it to the

22 signature calculated at the time the process was initiated. If the user has previously

23 downloaded the object or received it as an attachment to an Email, the user may

24 access the secure hash code through the signature system and apply it to the version

25 on the user's disk.

26 9. <u>Signing a Document</u> – After the user has determined that an information object

27 is authentic and that the contents merit signature, he or she can affix a signature by

28 authenticating his or her identity. Various means of authentication may be used. The

29 means of authentication may be at the discretion of the manager of the signature

30 system. Such means may include personal identification numbers, passwords,

31 authentication based on computer address or information stored on the signer's

32 computer, third party validation using a public key or other security infrastructure,

18

1   or biogenic (fingerprint-recognition, retina scan) methods.

2   After a document is signed, the date of signature is recorded in a database so

3   that the display to other potential signers is updated. If the signature process is serial,

4   the next person in the sequence is notified. E-mail notice can be sent to all signers

5   when the last signature is collected.

6   10. Follow-up – At the time a signature process is initiated, the initiator can select

7   a time (in hours, days, or a time or date-certain) for automated follow-up. If a

8   document is not signed within the specified period after notice, a follow-up e-mail

9   can be sent as a reminder. Additional reminders may be sent at the same interval if

10   the object has not been signed. The reminders can be sent automatically by the

11   system according to user-input specifications.

12   11. Cancellation – The initiator of a signature sequence can modify the sequence at

13   any time, except that a signer can not be deleted from the list once they have signed

14   an object.

15   12. Transfer of authority – The individual initiating a sequence can transfer the right

16   to modify the list signature list to another individual in the system with appropriate

17   validation of identity.

18   Document Manager

19   Successfully conducting commerce over an electronic network requires the

20   exchange not only of messages, but of substantial blocks of information in the form

21   of documents and data. Beyond simply transferring files from hand to hand, it is

22   often necessary for multiple parties to work on a document simultaneously or

23   serially, to track changes, and to maintain a record of versions. Two general

24   architectures have emerged for document management, which can be termed a "mail

25   model" and a "repository model." Under the mail model, documents are attached to

26   messages and circulated person to person. Under the repository model, documents

27   are placed in a central location. There are advantages and disadvantages to each. At

28   a summary level:

	Mail Model	Repository Model

SUBSTITUTE SHEET (RULE 26)

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.

1
2   A browser-based document management model and tool combines the best
3 features of repository model and the mail model, for document dissemination and
4 sharing across the Internet or an intranet.
5 General Architecture – The general architecture of the system combines two basic
6 components: (1) a database of directories and documents and (2) a directory of users.
7 The directory of documents lists documents (of any type) contained in the system,
8 and folders that can contain documents or other folders. The directory of users
9 contains a list of individuals and organizations that can access the system, with
10 passwords and/or other information necessary to validate identity and to establish
11 authority.
12 Representation of document – The term "document" is used here in the broadest
13 sense of any file that can be stored magnetically or electronically. Preferably, each
14 file is given a unique name consisting of a string of no more than 256 characters.
15 Preferably, the character set is limited to those members of the ASCII character set

20

1   which are displayable or printable. Thus, such codes as "escape" which have no ·

2   visible representation, would be excluded. This is the file name that is displayed for

3   purposes of identifying the document to the users. There is also an actual file name

4   (which is not shown to users) to identify where copies of the file are stored in the

5   central repository. Certain other information is kept in addition to the name of the

6   file. This includes the following:

7   1. Data of creation

8   2. Date entered into repository

9   3. Person who entered the document into the repository

10   4. Description

11   5. Size of the document

12   6. Document type if known

13   7. Date of last update

14   8. Access password (optional) stored in encrypted form

15   9. File folder(s) where the document appears

16   10. Actual file name

17   In addition to the above information, data indicating whether the file is

18   checked-out and to what entity, and the identities of entities that have checked the

19   document out and returned it in the past are also stored. The term "checking out" is

20   described further below.   These functions related to file change control and

21   configuration management, which are discussed later.

22   User database – A database contains information on all individuals who can currently

23   access the system or who previously had access up to an administratively determined

24   retention period. This database includes standard contact information including

25   physical and electronic addresses. Security data such as passwords and/or encryption

26   keys is also maintained. In a combined system such as the presently described

27   system, the same database or registry of users can be employed for the document

28   manager as for the signature system.

29   High level directories – The entire document management system can be divided into

30   a number of high level directories that the user can display, one at a time. These

31   include, at a minimum, a "Private" directory of files and folders visible only to the

32   user, and a "Public" directory of files and folders visible to all users. Additional

33   high-level directories can be created by the system administrator as needed. These

21

SUBSTITUTE SHEET (RULE 26)

1   could correspond to projects, business units, or any other logical basis. At any point ·

2   in the use of the document management system, a user can see and select from the

3   high level directories to which the user has access. The name of the currently open

4   directory can be always displayed on the screen.

5   <u>Displaying the contents of a high-level directory</u> – When a user selects a high-level

6   directory, the repository displays a series of file folders against the left margin of the

7   active window. File folders whose contents are displayed are shown as open folders.

8   File folders who contents are not displayed are shown as closed folders. A folder is

9   opened or closed by clicking a single time. When a folder is opened, the contents are

10   shown with an indent to indicate the parent/child relationship between the folder and

11   its contents. Each folder can contain files, shown by an icon representing a printed

12   page and other folders, represented by an image of a closed folder.

13   <u>Information about a folder</u> – Information about each folder is displayed on the same

14   line, to the right of the folder icon. This information is as follows, from left to right:

15   1. Name of the folder

16   2. Number of files in the folder, or the word "empty"

17   3. Accessibility of the folder

18   Accessibility refers to user access rights to a folder which may private relative to the

19   entity that created it, restricted (limited to a subset of people who can access the high

20   level directory), or shared (available to everyone with access to the high-level

21   directory). The level of access to a directory is indicated by the words "private",

22   "restricted" or "shared."

23   If the directory is restricted, clicking on the word restricted displays a list of

24   the entities that have access to the folder. This list is a series of hyperlinks. Clicking

25   on the name of a person pulls up detailed contact information (discussed below). The

26   objective is to facilitate communications between people with a shared interest in a

27   file.

28   <u>Information about a file</u> – Information about a file is displayed to the right of the file

29   icon. From left to right, the first item displayed is the name. This is followed by the

30   word "details." Clicking on "details," causes the document management system to

31   display complete information about the file (see Item 2, above), the person who

32   placed the document in the file, (see Item 3, above), and the person who most

33   recently modified the file.

22

1   <u>Information about people/entities, and the link to communications</u> – Information.

2   about people/entities with access to the system is displayable at several points in the

3   document manager system:

4   1. by accessing the directory of users

5   2. when creating a new folder with "restricted" access

6   3. when displaying detailed information about a file (see #7)

7   4. when displaying information about a restricted directory (see #6)

8   Whenever such information is displayed, contact information from the database is

9   rendered along with the name. Depending on the implementation, this can include

10   complete contact info (multiple addresses, telephone and fax numbers, and email

11   addresses), or some of the contact information may be restricted, in which case it is

12   not displayed.

13   <u>Creating a new top level folder</u> – A new folder is created within a high-level

14   directory, for example by clicking a button labeled "new folder." This can bring up

15   a dialog in which the user assigns a name to the new folder and selects the type of

16   access (private, shared, or restricted) rights to be assigned. If the document is

17   restricted, the user specifies the entities (organizations and/or people) that can access

18   the folder. If the creator of the folder specifies that an organization has access to a

19   folder, all individuals associated with that organization may be granted access.

20   Folders to which a user does not have access may remain hidden or not displayed.

21   Alternatively, these folders can be shown with some indication that they are not

22   accessible, for example, by ghosting.

23   <u>Functions related to a folder</u> – Once a folder is defined, a user can execute the

24   following options.

25   1. Create a subfolder, using the same process described in 9

26   2. Add a document to the folder, using the process described in 11

27   3. Delete the folder, if it is empty

28   4. Modify access to the folder using the same tools used to specify access

29   initially

30   The functions can be invoked by, for example, clicking on the appropriate label to the

31   right of the name of the folder icon.

32   <u>Adding a file</u> – Users add a document using a dialog box that prompts for the

33   following information:

23

1   1. Location of file - may be entered by user, or selected through a standard

2   file browse dialog

3   2. Name to be used for the file in the repository

4   3. Version number or name (optional)

5   4. Password or encryption key (optional)

6   5. Description (optional)

7   6. Access rules (read only or read-write)

8   After entering the above information, the user either aborts or initiates upload.

9   The information listed above is recorded along with the name of the person entering

10   the document, and date and time.

11   File options – The following functions may be provided, preferably for every file in

12   the system:

13   1. Delete (with confirmation)

14   2. Archive. The file is removed from main repository, but a copy is retained

15   outside the repository. It may be restored though manual intervention.

16   3. View or download: a copy of the file is brought to the user's computer.

17   This file can be modified there for the individual user's use. A modified

18   version can be uploaded as a new file or different version of a current one, but

19   a file in the repository can only be replaced if the user has it checked out.

20   4. Check out / check in (see below)

21   5. Forward (see below)

22   6. Change Password. The old password must be entered followed by a new

23   password and confirmation.

24   7. Move: copy or more a document from one folder to another.

25   The functions may be invoked, for example by clicking on a label

26   corresponding to the function, which can be displayed to the right of the name of the

27   file. Not all options are shown to all users. If an entity does not have write-access

28   to a file, the entity may not delete it, archive it, check it in or out, or change the

29   password.

30   Check in / Check Out – All entities with write access to a file may check it out. By

31   checking the file out, the entity reserves the exclusive write to save changes to a file.

32   A person may not replace a file that is checked out. To check out a file, the user

33   selects this option from the list of functions associated with the file. The user can

24

1   then enter an expected return date and a reason that the file is checked out or the ·

2   changes to be made. This information is available to all others who can view the file.

3   Each check in or check out is recorded in a permanent log. After a file is checked

4   out, the "check out" button or link is changed to read "check in."

5   Each individual can check in only the files that he or she has checked out.

6   This is done by clicking "check in." The user may then upload a new version of the

7   file by specifying the location of the file on disk, or indicate that the version of the

8   file currently in the repository is to be retained. After a file is checked in, the check

9   button is changed back to "check out" and the file can be checked out by another

10   user.

11   Forwarding – A file can be forwarded to any other user of the system. When the

12   forward function is invoked, a list of users is displayed. The sender selects one or

13   more users. Upon confirmation, a copy of the document is placed in folder labeled

14   "in box" in each recipients private directory.

15   Referring to FIG. 5, a main screen for the document manager creates (using

16   server-side scripting) a user-interface display with some of the features of a Windows

17   Explorer® -type display. File and folder icons are shown along with an array

18   features arranged next to each. The similarities with Windows Explorer® fairly well

19   end there, however. Each of the properties shown next to each file/folder entry

20   invokes a feature.

21   A parameter object W "Details" invokes a detailed display of the

22   corresponding document object. The details can include contact information about

23   the creator of poster of the document or other data as desired. This data can be

24   hyperlinked and a return button can be provided to return the display back to the

25   screen shown in FIG. 5. Clicking the "details" button to the right of any document

26   brings up the display which can include the name, contact information, and other

27   details about the person who loaded the document into the system, similar

28   information about a person who has the document checked out, and, optionally, a

29   description of the document and information on its change history.

30   A parameter object X "Forward" simply sends the document to a selected

31   user. A selection screen can be invoked to allow selection of the recipient of the

32   document from the user registry. Of course, since most correspondence can be

33   handled on the server side, the user is, in reality, simply notified of the transfer and

25

1   the recipient's action to view the document simply invokes a server side feature to .

2   display the document. The document is not actually transferred bodily to the

3   recipient since the recipient, as a registrant logged in the user registry, can access it

4   through the server by requesting to do so.

5   A parameter object U "Check-in" checks in a document that has been checked

6   out. Other users may view the document, but not modify it when it is checked out.

7   This button is not accessible to users that have not checked the document out and

8   may be displayed ghosted or not displayed at all. A similar button can be displayed

9   if a document that is not checked out may be checked out by the user authorized to

10   see the document manager displayed shown in FIG. 5.

11   A parameter object T "Download" actually transfers a copy of the document

12   to the client computer. Another object S "Delete" allows the document to be deleted.

13   A new document can be added by clicking "New Document" Q. These are fairly

14   conventional notions, except for their placement on the screen and the fact that each

15   is filtered depending on the user's rights.

16   Note that when a folder is created, access to the folder can be restricted to the

17   creator, shared with everyone (in which case the folder is created in the public

18   directory), or shared with a select group of other users. The other users can be

19   selected by company or organization (providing access to all individuals in the

20   organization) or by individual within an organization. These are all selectable

21   through a linked selection control where if one selects a company in one selection

22   control, it shows employees in the linked selection control.

23   A parameter object P "Shared" displays a hyperlinked page that shows all

24   users with access rights to the document. This page allows a user that places a

25   document in the document manager or a user that has pertinent modify rights, to alter

26   the parties that have access to the document. Also, it allows a user with read-only

27   rights to see the list of users that can access that document. The names of the sharing

28   parties are hyperlinked to invoke the user's email client to allow fast sending of email

29   (which again may be performed server-side without actual transfer) or conventionally

30   or selectively. If a folder is shared, the word "Shared" appears to the right of the

31   folder. Clicking on "Shared" brings up the list of person who can access the folder,

32   as shown in FIG. 6. Each name is a hyperlink to detailed contact information.

33   FIG. 7 shows a list of all deals that were completed through the system. The

26

1 trade number (left column of the grid) is a hyper link to detailed information.

2   FIG. 8A shows detailed information about a completed trade. It shows the

3 party to the trade, the price or rate, and a description of what was traded. The

4 particular nomenclature is specific to a market. For insurance, for example, price is

5 termed rate, and the summary of a deal is the slip sheet. A complete contract can be

6 attached. Included documents can be downloaded to view on line. The intended

7 signatories to a deal are shown (there can be more than two).

8   If a signatory has actually signed the document electronically, the date and

9 time are shown. No date and time are shown for parties that have not yet signed.

10 The amount of information displayed on the screen is dependent on the identity of

11 the person viewing the screen. The viewer can be blocked from viewing any

12 information about a deal, or certain fields, such as the contract details or the name of

13 signatories.

14   Note that the detail screen of FIG. 8A would also show attached exhibits. The

15 FIG. 8A display is the basic device for signing deals. A similar device would be used

16 for signing documents.

17   Referring to FIG. 8B, all of the information necessary to document a deal is

18 pulled together through the screen below. The deal summary includes highly

19 structured information on parties, dates, terms, etc., as well as unstructured

20 information in the form of attachments. The bottom part of the page allows the

21 person registering the deal to designate the intended signatories. When the signers

22 affix their electronic signature, they are doing so to all of the documents in the deal,

23 including the attachments. These are archived and protected from tampering using

24 secure hash technology. In this way it is possible to create a reliable, on line

25 electronic signature to a complex deal, without risk of repudiation.

26   Note that any number of exhibits can be added to the UI device of FIG. 8B

27 since the list scrolls from the bottom each time a second exhibit is added. The user

28 interface has self-explanatory elements for defining information about the deal.

29 Anonymous Mail

30   For purposes of the following description, a "subscriber" is a person or entity

31 that subscribes to an anonymous mail system to be described below. Certain types

32 of negotiations and communications require anonymous initial contact, followed by

33 some period of anonymous discourse, leading to eventual disclosure of the parties'

27

1   identities. In the course of a typical sale or business deal, the initiating party begins .

2   either by contacting one or more targeted potential trading partners or advertising to

3   a community of potential partners. While the identity of the initial offeror is usually

4   clear in any direct contact, it need not be so in advertising. In certain cases it could

5   be problematic for the initiating party to reveal his or her identity:

6   A party to a deal can have difficulty controlling the method of contact once

7   the party's identity is known. If a company is known to be in the market for office

8   space, for example, the party may be subjected to badgering by real estate firms

9   outside the established bidding process. Executives of the company may be contacted

10   directly in an effort to influence the decision.

11   Disclosure of intent may adversely affect the market. If a large company

12   begins to acquire land in an area, the price can rise very quickly. Simple exploration

13   of an option can make the option more costly or even impossible.

14   Disclosure of intent may adversely impact the reputation or standing of a

15   company. An insurance company that determines that it is over exposed to a certain

16   peril (e.g. hurricane losses in the Southeastern U.S.) would reveal that situation to

17   their competitors and investors by a large public solicitation.

18   While anonymity can be crucial for the initiator of a deal, it can be equally

19   important for the respondent for the same reasons. The need for controlled anonymity

20   has been addressed by several methods that were initially developed for paper

21   communications and have been extended to analogues in telephonic and computer

22   communications.

23   •   Numbered mail boxes, including government and private

24   •   Communications through a mediator

25   •   Anonymous voice mail drops

26   •   The use of pseudonyms in computer e-mail and dialogs.

27   These methods have several serious shortcomings:

28   •   The method may only allow anonymity from one side.

29   •   There is no inherent mechanism to validate the credentials and intent

30   on an anonymous party

31   •   Use of a pseudonym may invalidate its future use by associating the

32   name with a specific party

28

1   •   Manually mediated communications are slow

2   •   The creation and deletion of pseudonyms may not be completely

3 within the control of the party, imposing an overhead cost (in cash or labor)

4 and/or delay in creating a new name

5   •   In most systems, a person with multiple pseudonymous mailboxes or

6 e-mail addresses will receive communications in several different places

7 (mailboxes or accounts), thus requiring multiple logons/passwords.

8   •   Routing of messages received anonymously requires manual

9 forwarding to all relevant parties by the individual with access to the

10 anonymous mail box or email account.

11   •   There is no mechanism to reveal actual identities in a secure and

12 mutually acceptable way.

13   The present invention addresses these deficiencies by providing two-way

14 anonymous communications, a central point of collection for messages sent to

15 multiple pseudonymous addresses, connection of multiple parties to a single

16 anonymous account, and a mechanism to reveal identities to all parties to a deal

17 simultaneously, by mutual consent. In summary, the anonymous mail system is a

18 server side system that allows clients to create anonymous handles on the fly. It also

19 allows them to share anonymous handles among multiple recipients so that the group

20 of recipients appears as a single recipient to the sender using the anonymous handle.

21 It is like a transparent mailing group. When mail is sent to an anonymous handle, it

22 is sent to all members of the group.

23 Multiple Systems – In contrast to the first-generation anonymous mail system, the

24 present system allows for multiple anonymous mail (Amail) systems. Each Amail

25 system operates in association with a conventional e-mail server, and uses the e-mail

26 server for communications with non-subscribers, subscribers to Amail systems other

27 than the local one, and for forwarding messages to the subscribers Email client

28 software.

29 Registration – Subscribers to an anonymous mail system (Amail) each complete a

30 registration that provides:

31   •   Contact information (name, address, telephone number, fax, etc.)

32   •   Information to determine whether they the party is qualified to

1   participate in the communications exchange. For example, if the system were

2   to be used between and among real-estate agents, registrants to the system

3   might be required to supply a real estate license number.

4   •   Association with an organization (if appropriate)

5   •   Additional information on the individual or organization that may be

6   of use to others in the Amail system to determine the suitability of the party

7   as a partner in negotiations.

8   The additional information can include such factors as credit ratings, assets, or the

9   region in which the company does business. The specific information required

10   depends on the application. Insurance, real estate, energy marketing, etc. would all

11   have different data of interest.

12   <u>Validation</u> – Depending on the business model and role of the organization operating

13   the Amail exchange, the organization can either accept the information provided by

14   the subscriber, or verify the information and provide verification as part of the

15   service. Upon acceptance of a subscription applications and validation of the

16   background information if necessary, the use is assigned an Amail user ID and

17   password.

18   In the first version of the Amail system, logon was automatic from the general

19   application (CATEX); there was no separate user ID and password. In alternative

20   versions, the Amail system can provide its own user ID and password, with the

21   ability to bypass logon when it accessed from other applications with acceptable user

22   validation. All of the actual contact information and validation information are

23   maintained in a database. Validation information was not provided in the first

24   version of CATEX.

25   <u>Assignment of an Email address</u> – Each subscriber must provide an Internet

26   accessible Email address or be assigned an e-mail address in the Amail system. The

27   first version of the Amail required that the user have an Email address on the system.

28   The new version works directly with e-mail systems other than the Amail.

29   <u>Logon</u> – Subscribers access the Amail system by connecting an Amail web page

30   provided either over the Internet or on an Intranet. The subscriber enters a user name

31   and password. The first version of Amail was not browser-based and worked only

32   over a LAN or WAN, not over the Internet or an intranet.

30

1   Available functions – After logon, the subscriber can access the following functions: ·

2   •   Manage aliases

3   •   Compose an anonymous message

4   •   Read Amail messages. In the original CATEX system, the user could

5   not access messages from within the Amail application.

6   •   Log off

7   Managing Aliases – Aliases are directly under user control. After logon, a user can:

8   •   Add a new aliases

9   •   Delete an existing alias

10   •   Create a free-form note associated with a new alias, or edit the note for an

11   existing alias that will be accessible to recipients from the alias.

12   •   Identify other subscribers to whom messages to alias should be forwarded

13   •   Identify other subscribers with permission to generate messages from the alias

14   These last two features make it possible for a group of subscribers to share an alias,

15   allowing them share communications and work together more effectively. The user

16   will:

17   Compose an anonymous message – After logon, a user can create and send an

18   anonymous message. After the option is selected, the system will display a message

19   creation screen with the following features:

20   1. A list of aliases currently owned by the user (i.e. created by the user and

21   not deleted), for the user to select the alias from which the message will

22   originate.

23   2. A subject box for the mail.

24   3. A list of the e-mail and alias addresses to which messages can be sent for

25   the user to select one or more. The original version could only send to one

26   alias. The user can also supply an Internet e-mail address off system.

27   4. A list of the e-mail and alias addresses to which copies of the messages

28   can be sent for the user to select one or more. The user may also supply an

29   Internet e-mail address off system. The original version did not include a

30   "CC" feature.

31   5. A space where the message can be typed, allowing for users to paste text

32   copies form another system using the Windows-based clipboard utility.

31

1   6. A check box to select whether the sender is willing to reveal his identify .

2   to the recipient on mutual consent.

3   7. A check box to select whether the copies of the message should be sent to

4   other subscribers who share the Alias. The original version allowed only one

5   subscriber to access an alias.

6   <u>Delivery of Messages</u> – After an Amail message has been composed (see step 7), it

7   is delivered as follows.

8   1. The body of the email message is modified by adding a header including

9   routing information and an indication of whether the sender is willing to reveal

10   identities if there is reciprocal concurrence. The message would appear as shown

11   below. The items in italics are new since the original (prior art) version. The first

12   generation of the anonymous mail system did not allow for communications between

13   multiple Amail systems and, hence, did not list the Amail system name in the list of

14   respondents. The first generation system also did not allow for multiple recipients.

32

SUBSTITUTE SHEET (RULE 26)

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.

Receipt of Messages – 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

33

1   read Amail messages (see above) the messages are rendered as an HTML page
2   through a browser. Messages to all of the aliases associated with the user are
3   displayed. Each message has a hotlink to respond to send a message back to the
4   sending alias. Each message also has a link to display the background and validation
5   information and note associated with the alias (see above). The original version did
6   not provide an Amail viewer nor did it provide for display of validation information.
7   Responding from off System from Amail – Individuals from off system can respond
8   to Amail messages using the standard reply feature of their mail server. Messages
9   will be returned to the reply address (see above). Messages received by the
10   conventional e-mail server supporting the Amail system will forward the message to
11   the Amail message repository for the alias listed in the return address. Responding
12   from a standard Email client was not provided in the original version.
13   Flip Widget
14   Increasingly, computer applications are delivered through browsers over the
15   Internet or an intranet. There are many design considerations in building a system
16   for browser delivery in contrast to delivery as conventional client server application.
17   Two related considerations are the graphic richness of a browser screen and the time
18   lag to render a new screen. Partly because good web pages contain complex graphics
19   and partly because the Internet can be a relatively slow network, it is important to
20   design a web application to make few unnecessary wholesale screen changes. It is
21   more economical from the perspective of data transmission and, hence, from response
22   time, to create a "flat" rather than "deep" hierarchy of screens, and change only the
23   part of a screen that is minimally necessary.
24   For example, it is better in a data query to provide a single screen that allows
25   a user to specify a state and city within the state than to provide a first screen for the
26   state, followed by a second screen for the city. As the function of screens becomes
27   more complex, however, it becomes an increasingly difficult challenge to fit all of
28   the options onto the screen (particularly when a user selects a lower screen
29   resolution) and while maintaining a clean appearance. The invention described here
30   provides a tool that allows the Internet application developer to display an effectively
31   unlimited number of options in a very small space using a very familiar and intuitive
32   display feature.
33   Appearance – The "Flip Widget" tool renders a graphical object representing two

34

1   rows of file folders, overlapping. The labels on the front row are visible, the labels.

2   on the second row are obscured by the front row of tabs, but the edges of the apparent

3   back tabs are visible. The number of the apparent tabs displayed in each row is a

4   function of the screen resolution and the length of the longest label entered by the

5   user.

6   The Flip Tab – In one embodiment, the rightmost tab on the front row is labeled

7   "FLIP". When a user actuates this tab, the response is as described below.

8   Database of labels and links – In creating the display, the application programmer

9   enters a set of paired values. Each pair consists of (1) text of the label to be displayed

10   and a tab, and (2) the name of an HTML link, either within or external to the page to

11   be rendered when the tab is selected.

12   Action – Upon rendering a page containing the flip widget, the two-row tab display

13   shows the first "n" options from the list of labels and links. The value of "n"

14   represents the maximum number that can be displayed while allowing room for the

15   flip tab. Upon clicking any of these tabs, the corresponding link is executed. Upon

16   clicking the flip tab, the two-row tab display is changed to reflect the next "n" options

17   from the list of labels and links, retaining the flip tab on the right. If there are fewer

18   than n options remaining, the flip widget will either display the last n options, or

19   whatever number remain supplement by as many options are needed from the start

20   of the list. Clicking the flip tab when the list has been completed starts the cycle over

21   again with the first option.

22   Referring to FIGS. 9 and 10, a flip widget in a first state is shown in FIG. 9.

23   In the first state, any of the tabs A through E can be selected and the corresponding

24   set of controls displayed. For example, in FIG. 9, tab B has been selected and the

25   controls 430-432 are displayed. If the flip tab 410 is selected, a next row of tabs is

26   brought forward so that the display appears as in FIG. 10 with tabs F through J

27   showing. In FIG. 10, tab G has been selected and the corresponding controls 435-

28   437 are displayed.

29   FIGs. 9A and 10A show a more detailed example of how a flip widget can be

30   used to organize functions available to a user. For example, suppose that one

31   application is a commodity futures trading system that permits a user to execute

32   trades, review prices, and obtain other information relating to various metals such as

33   gold, silver, and platinum. As shown in FIG. 9A, for example, controls or functions

35

1   430, 431, and 432 (e.g., execute a trade, review current prices, and the like) are .

2   associated with a "gold" category and can be invoked easily when that category is at

3   the forefront of the flip widget as shown. Clicking one of the other tabs (e.g., silver

4   tab 400) would bring the functions associated with that category to the forefront

5   while allowing the user to readily select other categories visible behind the front.

6   Clicking "other markets" tab 410 would change the selection of front-row tabs to a

7   different set of categories, as shown in FIG. 10A. The "other markets" tab 410 could

8   be continually clicked to rotate through a plurality of groupings of markets, each

9   having a set of functions or controls associated therewith.

10   A flip widget can be implemented in conjunction with the first or second

11   embodiments of the present invention in order to permit many different functions to

12   be displayed in a small screen space. The flip widget is a device to organize many

13   different functions in a logical way, and can be used as a tool for building an interface

14   to multiple applications.   As one example, in a DCE (described in more detail

15   below), there may exist n functions (e.g. bulletin boards, chat rooms, e-mail, a-mail,

16   transaction engines, and the like) the specific availability of which can be defined by

17   a user who creates the collaborative environment. This collection can change over

18   time. Accordingly, the interface cannot be "hard coded" for a particular user.

19   One way to represent an indefinite (and potentially large) number of functions

20   in a small space is with tabs resembling a file folder, with a graphic element

21   representing hidden cards, implying that the user can reach the functionality on the

22   cards by paging (i.e. flipping) to them. The flip widget makes it possible to provide

23   a link to a list of applications maintained in a database rather than requiring that they

24   be hard coded. Programming logic for storing folder labels in a database, linking

25   those labels with associated functions and activating them using browser-type

26   buttons, and for performing the display features described above, are conventional

27   and no further elaboration is necessary. Although the "flip widget" provides one

28   method of structuring a user interface to structure a user's view of application

29   functions, other methods can of course be used.

30   B. DYNAMIC COLLABORATIVE ENVIRONMENT EMBODIMENT

31   In a second embodiment of the invention, a dynamic, user-defined

32   collaborative environment can be created in accordance with a set of tools and

33   method steps. As explained previously, this system differs significantly from

1   conventional networked environments in that: (1) the environment (including access

2   and features) is user-defined, rather than centrally defined by a system administrator;

3   (2) each environment can be easily destroyed after completion of its intended

4   purpose; (3) users can specify a group of participants entitled to use the environment

5   and can define services available to those participants, including offering

6   participation to unknown potential users; (4) the networked environment (including

7   access features and facilities) can cross corporate and other physical boundaries; and

8   (5) the environment offers a broad selection of tools that are oriented to

9   communication, research, analysis, interaction, and deal-making among potential

10 group members.   Moreover, in a preferred embodiment, the environment is

11 implemented using web browser technology, which allows functions to be provided

12 with a minimum of programming and facilities communication over the Internet.

13   FIG. 11 shows various method steps that can be carried out to define, create,

14 and destroy an environment according to a second embodiment of the invention. The

15 term "environment" as used herein refers to a group of individuals (or computers,

16 corporations, or similar entities) and a set of functions available for use by that group

17 when they are operating within the environment. It is of course possible for one

18 individual to have access to more than one environment, and for the same functions

19 to be available to different groups of people in different environments.

20   The process of creating a collaborative environment involves the migration

21 of tools and information resources available in the library of the environment

22 generator into a specific collaborative environment.   The collaborative

23 environment can include / link to any application available to the environment

24 generator.   It can also include applications specific to the environment provided

25 that theses are accessible through Internet protocols.

26   Underlying the environment is a directory of users, information about

27 users, and their authorities.   The core structure for the environment user database

28 should conform to a directory standard – typically DAP (Directory Access

29 Protocol) or LDAP (the lightweight directory access protocol).   The environment

30 generator has access to its own directory of users and to the user directories of the

31 environments it has generated.   The directory of an environment can be populated

32 initially by selecting users from the environment generator's directories. These

33 are added to the directory of the environment in one of two ways depending on the

1   specific implementation. Directory records can be copies from the environment

2   generators user database to a separate database for the environment or a flag can

3   be added to the user data record in the environment generators users database to

4   indicate that the user has access to the environment. The second, simple model is

5   useful when all users in an environment have equal authority. A separate user

6   database (directory) is necessary for an environment when the environment has its

7   own security / authority model.

8   Additional members can be added through a set of standard application /

9   subscription routines. These then become known to the environment generator (as

10   well as the specific environment) providing the foundation for greater speed and

11   efficiency in creating subsequent environment.

12   Beginning in step 1101, a new group is created by identifying it (i.e., giving

13   it a name, such as "West High School Research Project," and describing it (e.g.,

14   providing a description of its purpose). The process of creating a group and defining

15   functions to be associated with the group can be performed by a user having access

16   to the system without the need for system administrator or other similar special

17   privileges (e.g., file protection privileges, adding/deleting application program

18   privileges, etc.).   In this respect, environments are, according to preferred

19   embodiments, completely user-defined according to an easy-to-use set of browser-

20   driven user input screens. The principles described herein are thus quite different

21   from conventional systems in which a central system administrator in a local area

22   network can define "groups" of e-mail participants, and can install application

23   programs such as spreadsheets, word processing packages, and the like on each

24   computer connected to the network. Moreover, according to various preferred

25   embodiments, the facilities provided to group members can be provided through a

26   web-based interface, thus avoiding the need to install software packages on a user's

27   computer.

28   It is also contemplated that various methods of obtaining payment for creating

29   or joining groups can be provided. For example, when a new environment or group

30   is created, the person or entity creating the group can be charged a fixed fee with

31   payment made by credit card or other means. Alternatively, a service fee can be

32   imposed based on the number of members that join, the specific functions made

33   available to the group, or a combination of these. Moreover, fees could be charged

38

1   to members that join the group. The amount of the fee could also be based on the

2   length of time that the environment exists or is used.

3   Although not specifically shown in FIG. 11, step 1101 can include the step

4   of creating a new entry in a database table (e.g., a relational or object-oriented

5   database) to store information concerning the new group and the environment in

6   which the group will operate. Database entries related to the group, including some

7   or all of the information described below, can be created as the environment is

8   defined. It is assumed that one or more computers are linked over a network as

9   described in more detail below in order to permit the environment to be created, used,

10   and destroyed, and that a database exists on one or more of these computers to store

11   information concerning the environment.

12   In step 1102, the group members are identified. According to various

13   embodiments, the group members can be identified in three different ways (or

14   combinations thereof), as indicated by sub-steps 1102a, 1102b, and 1102c in FIG. 11.

15   It is contemplated that group members can span physical networks and computer

16   systems, such as the Internet. Consequently, group members can include employees

17   of different corporations, government agencies, and the like.   In contrast to

18   conventional virtual private networks, both the group members and the functions

19   made available to those group members are entirely user-selected, thus permitting a

20   broad range of persons to easily create, use, and destroy virtual private networks and

21   associated functionality.

22   First, in step 1102a, group members can be identified by selecting them from

23   a list of known users that are to be included in the group. For example, within a

24   corporation or similar entity, a list of internal e-mail addresses can be provided, or

25   an electronic version of a phone list or other employee list can be provided. If the

26   hosting computer system is associated with a school, then a list of students having

27   accounts on the computer (or those in other schools that are known or connected to

28   the host) can be provided. From outside a corporate entity, users can be selected

29   based on their e-mail addresses (e.g., by specifying e-mail addresses that are

30   accessible over the Internet or a private or virtually private network). In this step, the

31   environment creator specifies or compels group members to belong to the group.

32   Second, in step 1102b, group members can be invited to join the group by

33   composing an invitation that accomplishes that purpose. For example, a group

39

1   creator may choose to send an invitation via e-mail to all members of the corporation, .

2   or all members of a particular department within the corporation, all students in a

3   school or region, or members of a previously defined group (e.g., the accounting

4   department, or all students in a particular teacher's class). The invitation would

5   typically identify the purpose of the group and provide a button, hyperlink, or other

6   facility that allows those receiving the invitation to accept or decline participation in

7   the group. As those invited to join the group accept participation, their responses can

8   be stored in a database to add to those members already in the group. Invitations

9   could have an expiration date or time after which they would no longer be accepted.

10   As invitees join the group, the group creator can be automatically notified via e-mail

11   of their participation.

12   Third, in step 1102c, group members can be solicited by way of an

13   advertisement that is sent via e-mail, banner advertisement on a web site, or the like.

14   Persons that see the advertisement can click on it to join the group. It is also possible

15   for advertisements to have a time limit, such that after a predetermined time period

16   no more responses will be accepted. The primary difference between advertising

17   participation in a group and inviting participation in a group is that invitations are

18   sent to known entities or groups, while advertisements are displayed to potentially

19   unknown persons or groups.

20   It will be appreciated that group members can be selected using combinations

21   of steps 1102a, 1102b, and 1102c. For example, some group members can be

22   directly selected from a list, while others are solicited by way of invitation to

23   specifically identified invitees, and yet others are solicited by way of an

24   advertisement made available to unknown entities.

25   In step 1103, the functions to be made available to the group are selected. For

26   example, the group can be provided with access to an auction transaction engine; a

27   survey tool; research tools; newswires or news reports; publication tools; blackboard

28   facilities; videoconferencing facilities; and bid-and-proposal packages. Further

29   details of these facilities and tools are provided herein. The group creator selects

30   from among these functions, preferably by way of an easy-to-use web browser

31   interface, and these choices are stored in a database and associated with the group

32   members. Additionally, the group creator can specify links to other web-based or

33   network-based applications that are not included in the list by specifying a web site

40

SUBSTITUTE SHEET (RULE 26)

1   address, executable file location, or the like. The group creator can also define shared .

2   data libraries that will be accessible to group members.

3   In step 1104, the environment is created (which can include the step of

4   generating a web page corresponding to the group and providing user interface

5   selection facilities such as buttons, pull-down menus or the like) to permit group

6   members to activate the functions selected for the group. In some embodiments,

7   access to the group may require authentication, such as a user identifier and password

8   that acts as a gateway to a web page on which the environment is provided. Other

9   techniques for ensuring that only group members access the group functions and

10   shared information can also be provided. A web page can be hosted on a central

11   computer at an address that is then broadcast to all members of the group, allowing

12   them to easily find the environment.

13   In step 1105, group members collaborate and communicate with one another

14   using the facilities and resources (e.g., shared data) available to group members. In

15   the example provided above, for example, a group of high school students

16   collaborating on a school research project could advertise for survey participants;

17   conduct an on-line survey; compile the results; communicate the results among the

18   group members; brainstorm about the results using various brainstorming tools;

19   conduct a videoconference including group members at various physical locations;

20   compile a report summarizing the results and exchange drafts of the report; and

21   publish the report on a web site, where it could optionally be offered for sale through

22   the use of an on-line catalog transaction engine. The group could even contact a

23   book publisher and negotiate a contract to publish the report in book form using bid

24   and proposal tools as described herein.

25   In step 1106, after the environment is no longer needed, it can be destroyed

26   by the person or entity that created the group. Again, in contrast to conventional

27   systems, the destruction of the environment is preferably controlled entirely by the

28   user that created the environment, not a system administrator or other person that has

29   special system privileges. Destruction of the environment would typically entail

30   deleting group entries from the database so that they are no longer accessible.

31   FIG. 12 shows one possible system architecture for implementing the steps

32   described above. As shown in FIG. 12, an Internet Protocol-accessible web server

33   1201 is coupled through a firewall 1202 to the Internet 1203. The web server includes

41

SUBSTITUTE SHEET (RULE 26)

1   an environment generator 1201a which can comprise a computer program that
2   generates user-defined environments as described above. Further details of this
3   computer program are provided herein with reference to FIG. 21.

4   Web server 1201 can include an associated system administrator terminal
5   1204, one or more CD-ROM archives 1205 for retaining permanent copies of files;
6   disk drives 1206 for storing files; a database server 1207 for storing relational or
7   object-oriented databases, including databases that define a plurality of user-
8   controlled environments; a mail server 1208; and one or more application servers
9   1209 that can host application programs that implement the tools in each
10 environment. Web server 1201 can also be coupled to an intranet 1210 using IP-
11 compatible interfaces. Intranet 1210 can in turn be coupled to other application
12 servers 1211 and one or more user computers 1212 from which users can create,
13 participate in, and destroy environments as described herein, preferably using
14 standard web browsers and IP interfaces. Web server 1201 can also be coupled to
15 other user computers 1217 through the Internet 1203; to additional application
16 servers 1215 through another firewall 1216; and to another IP-accessible web server
17 1213 through a firewall 1214.

18   It will be appreciated that the system architecture shown in FIG. 12 is only
19 one possible approach for providing a physically networked system in which user-
20 defined network environments can be created and destroyed in accordance with the
21 principles of the present invention. It is contemplated that application programs that
22 provide tools used in a particular user-defined environment can be located on web
23 server 1201, on user computers 1217, on application servers 1215, on application
24 servers 1209, on application servers 1211, or on any other computer that provides
25 communication facilities for communicating with web server 1201. It will also be
26 appreciated that web pages that provide access to each user-defined environment
27 need not physically reside on web server 1201, but could instead be hosted on any
28 of various computers shown in FIG. 12, or elsewhere.

29   Reference will now be made to exemplary steps and user interfaces that can
30 be used to carry out various principles of the invention, including steps of creating
31 a group, selecting group members, and defining functions to be made available to
32 group members in the environment.

33   FIGS. 13A through 13C show one possible user interface for creating a group

42

1 and identifying group members. In FIG. 13A, a user gains access to an environment

2 creation tool by way of an authentication process. This may be a simple username

3 and password device as shown in FIG. 13A, or it could be some other mechanism

4 intended to verify that the user has access to the environment creation tool. In the

5 case of a corporation, school, or other entity that already provides a log-in procedure

6 to access the entity's network, such log-in procedure could serve to authenticate the

7 user for the purpose of creating a new environment. It should be appreciated that

8 user authentication is not essential to carrying out the inventive principles.

9 Moreover, although it is contemplated that for ease of use (and to minimize

10 programming) web browsers and web pages be used to receive user-defined

11 information to create each environment, other approaches are of course possible.

12   In FIG. 13B, the user is prompted to create a new group by supplying a group

13 name (e.g., "Joe's Homework") and a brief description of the group. This

14 information is preferably stored in a database file and associated with group members

15 and functions available to those group members.

16   In FIG. 13C, the user is prompted to identify group members. As described

17 previously, group members are preferably identified in one of three ways (or

18 combinations of these): (1) selection from a list of known group members; (2)

19 inviting known candidates to join the group; or (3) advertising for new members.

20 When the user clicks one of the options in FIG. 13C, he or she is prompted to supply

21 additional information as shown in FIGS. 14A through 14C.

22   Beginning with FIG. 14A, for example, group members can be individually

23 specified by entering an e-mail address (e.g., an internal or external e-mail address)

24 in a text form data entry region and/or by selecting from a previously known list.

25 This screen permits the user to compel attendance in the group by specifying names

26 and/or e-mail addresses to which group messages will be sent. All those added to the

27 group in this manner will be provided with access to the environment corresponding

28 to the group. Aliases and pre-defined groups could also be specified as the basis for

29 membership (e.g., all those in the accounting department of a corporation, or all

30 students in a high school).

31   Each member of a group might have a group email account, or they may use

32 an off-system email account. Off-system email addresses can be maintained in a

33 database of users. Mail sent to the group email address is preferably forwarded off-

43

1   system, protecting the actual email address of the person unless that person wishes

2   to give out that address. New members can be added until the group is completed.

3   Although not explicitly shown in FIG. 14A, it is contemplated that new members

4   can be added to a previously defined group after the environment has already been

5   created.

6   When group members are selected or specified, the user creating the

7   environment can also create a password for each user in the group in order to enable

8   those in the group to access the environment. Alternatively, when a user visits the

9   environment, the environment can retrieve a "cookie" from the user's computer to

10 determine whether the user is authorized to access the environment. If no cookie is

11 available, the user could be prompted to supply certain authentication information

12 (e.g., the company for whom he or she works, etc.) In yet another approach,

13 authentication could occur by way of e-mail address (i.e., when the user first visits

14 the environment, he or she is prompted to enter an e-mail address). If the e-mail

15 address does not match one of those selected for the group, access to the environment

16 would be denied.

17   Turning to FIG. 14B, prospective group members can also be "invited" to join

18 the group. The user creating the environment can specify one or more e-mail

19 addresses to which an invitation will be sent. The invitation can be a simple text

20 message, or it could be a more sophisticated video or audio message. An expiration

21 date can also be associated with the invitation, such that responses to the invitation

22 received after the date will not be accepted. Software resident in web server 1201

23 (FIG. 12) receives responses to the invitations and adds members to the appropriate

24 group or drops them if the expiration date has passed or the prospective group

25 member declines participation. Prospective members can join the group by sending

26 a reply with a certain word in the message (e.g., "OK" or "I join"); by clicking on a

27 button in an e-mail message; or by visiting a web site identified in the invitation.

28   Turning to FIG. 14C, group members can also be solicited by creating an

29 advertisement directed primarily at potential group members that are unknown. The

30 advertisement could include, for example, a banner ad comprising text, video, and/or

31 audio clips. The graphic should conform to the size designated for the ad on the web

32 page. The ad could be posted on a web site by uploading the graphic through a web

33 interface and, optionally providing a URL on the screen of FIG. 14C to link to if the

44

1   advertisement is clicked. Software on the group page can render advertisements on .

2   a page either (a) every time the page is displayed, (b) in rotation with other ads; or

3   (c) when characteristics of the user match criteria specified for the ad.

4   The advertisement can include an expiration date after which responses would

5   no longer be accepted. Advertisements could range from the very specific (e.g., an

6   advertisement posted on a school's home page advertising participation in Joe's

7   research project on drug use at the school) to more general (e.g., an advertisement

8   that says "we're looking for minority contractors looking to establish a long-term

9   relationship with us" that is posted on web sites that cater to the construction

10 industry.

11   A qualification option can also be provided to screen prospective group

12 members. For example, if an advertisement seeks minority contractors to participate

13 on a particular construction project, selecting the "qualify" option would screen

14 responses by routing them to the user that created the group (or some other authority)

15 before the member is added to the group. Those responding to the advertisement

16 could be notified that they did not pass the qualifications for membership in the

17 group, or that further information is required (e.g., documents evidencing

18 qualifications) before participation in the group will be permitted. Alternatively, an

19 automatic qualification process can be provided to allow a prospective member to

20 join if the person fills in certain information on the response (e.g., e-mail address,

21 birthdate that meets certain criteria, or the like).

22   As shown in FIG. 15, a banner ad displayed on a web site invites minority

23 contractors to join a group that bids on information technology contracts. Those

24 interested in the advertisement click a button, which leads them to another site (not

25 shown) requiring that they provide certain information (qualification information,

26 name, age, company registration information, etc.) This information is then

27 forwarded to web server 1201 which either pre-screens the information according to

28 pre-established criteria, or notifies the user creating the group that a prospective

29 member has requested access to the group. In the latter case, the user could screen

30 the applicant and grant access to the group.

31   FIG. 16 shows one possible user interface for selecting communication tools

32 to be made available to group members. This screen can be presented to the user

33 creating the environment after the group has been identified and its members

45

1   selected. It is contemplated that a variety of communication tools can be provided,

2   including a bulletin board service; advertisements; white pages (e.g., a listing of

3   members, their e-mail addresses, telephone numbers, and the like); yellow pages

4   (e.g., a listing of services or companies represented by group members, with

5   promotional and contact information); document security (e.g., shared access secure

6   document storage services); anonymous e-mail (described above with respect to the

7   first embodiment); threaded dialogs; a group newsletter creation tool;

8   videoconferencing; and even other user-provided applications that can be specified

9   by name and location (e.g., URL). Details of these services are provided below.

10   According to various preferred embodiments, dynamic collaborative

11   environments are designed to integrate tools from multiple sources provided that they

12   are web-accessible (i.e., they operate according to Internet Protocol and/or HTML-

13   type standards). The categories listed above provide a reasonable taxonomy of the

14   tools necessary for collaboration, but this list can be extended to include virtually

15   every class of software such as computer-assisted design, engineering and financial

16   analysis tools and models, office applications (such as word processing and

17   spreadsheets), access to public or proprietary databases, multimedia processing and

18   editing tools, and geographic information systems. The following describes some

19   of the communication tools that can be provided:

20   Bulletin boards. A bulletin board (see, e.g., FIG. 2) lists notices posted by

21   group members, which may be offers to buy or sell, but need not be limited to such

22   offers. Many types of bulletin board services are of course conventional and no

23   further discussion is necessary in order to implement one of these services.

24   Nevertheless, in one embodiment the following data items (attributes) can be

25   provided for each notice appearing on the bulletin board: an item number, a title, the

26   date posted, and one or more special attributes defined by the user. The attributes

27   may include a field to indicate whether a listing is a "buy" or "sell " offer. The board

28   can be provided with an integrated sorting capability. By clicking on the heading

29   of each column, the user can sort the entries in, alternately, ascending or descending

30   order. Thus, it is possible to organize the records from oldest to newest or newest to

31   oldest, or to separate buy and sell offers. To limit the values on a board, a search

32   capability can also be provided, such that only those entries that meet the search

33   criteria are displayed.

46

1   Advertisements. In a typical environment of a dynamically created network

2   there are a number of fixed places for advertisements – the top of a page for a banner,

3   the bottom of a page for a banner, and space on the side for small ads. The creator

4   of the environment may choose to use none, any, or all of these spaces for

5   advertisements. Once a space is designated for advertising, group members may

6   place adds by completing a template that provides payment information (if required),

7   the text for the ad (any standard image format), and a link to be executed if the ad is

8   clicked by someone viewing the ad.

9   Each user is responsible for providing functionality behind the link. The ad

10   may be displayed persistently (every time a page is displayed), in rotation with other

11   ads for the same place, or may be triggered on the basis of user characteristics

12   including purchasing history. Revenue can be collected for placement (fixed price

13   regardless of how many times an ad is displayed), per time that the ad is displayed,

14   or per click on the ad. The virtual private network provides the front-end to facilitate

15   online placement of the ad. Display can be done by linking pages to standard ad

16   display code, available off the shelf from several sources. This code provides for

17   rotation of the ads. Software for customization (i.e. choosing the ad based on user

18   characteristics) is available commercially from several sources.

19   White pages. White pages provide a comprehensive listing or directory of

20   members with information about them and information regarding how to contact

21   them. Various types of commercially available software can be used to manage such

22   directories, and it is elementary to code typical directories that have fixed contents

23   for each member.

24   A web-accessible directory can be used in accordance with various

25   embodiments of the invention. One type of directory that can be provided differs

26   from directories having fixed structures. The key differences are as follows:

27   (a) User control over information Users enter and maintain their own

28   information directly, rather than through a central organization. This provides more

29   immediate update of data and reduces transcription errors. It makes it simple, for

30   example, for people to change their phone number when they are temporarily

31   working at another location.

32   (b) Multiple points for quality control. The data regarding each user can be

33   displayed to the user periodically (e.g.30, 60, and 90 days), and the user prompted to

47

1   update and verify the data. A feedback capability can be provided for members of

2   a group to report errors they find. Email addresses can be "pinged" periodically to

3   determine if they still exist. In addition, server management staff can periodically

4   review accounts that have had recent activity.

5   (c) Object structure. A directory entry consists of a collection of data

6   elements. These elements include such things as name for addressing (Dr. John D.

7   Smith), sort name (Smith, John D), or primary work telephone (800-555-1212).

8   Traditional mail systems have a fixed number of rigidly formatted elements. In one

9   embodiment, a more flexible approach can be used in that individuals identify which

10   elements they wish to add to the collection comprising their directory entry. For

11   example, a person can add 3, 4, 5 or more telephone numbers attaching a note to each

12   explaining its use (e.g. "for emergencies after 8PM").

13   (d) Direct link to communications tools. Where a directory refers to a contact

14   method (e.g. a telephone number), the method can be invoked directly from an entry

15   if the necessary software is available. For example, phone number can be dialed,

16   email messages initiated, or a word processing session initiated with letter and

17   envelope templates, preloaded with address information.

18   (e) Descriptive information. In addition to contact information, each directory

19   can contain information describing the entry (individual or business). The

20   description can be different in each group or it can be the same. The descriptive is

21   free form, with the exception that the user may drop in terms from a group-specific

22   lexicon. This lexicon can include terms specific to the industry (e.g. "fuel system")

23   for the automotive industry, or preferred forms of standard terms (e.g. "California"

24   rather than "CA", "Ca", or "Calif."). Standardization of terms in this way makes

25   search the directory more reliable.

26   Yellow pages. Conventional "yellow pages" products provide a one level

27   classification of directory entries designed to facilitate identification of and access

28   to an individual or organization with specific interests and capabilities. Within

29   industries, and particularly online, multi-level hierarchical directories are common,

30   with the multiple levels providing more precise classification. There are numerous

31   commercial products for maintaining online yellow page type classification systems.

32   Any web-accessible directory can be connected to a DVPN group. A

33   preferred method offered with the system integrates the classification system with the

48

1   descriptive field in a directory entry.   Every time a standard term pertaining to a .

2   classification is pulled from the lexicon, the entry is added to that classification in the

3   hierarchical sort. In addition to hierarchical access, this correspondence between the

4   traditional hierarchical sort and the free-form description with standardized terms

5   makes it possible to access records via search rather than browsing the hierarchy.

6   Searching makes it possible to identify an organization with multiple capabilities

7   (e.g. "brake repair" and "frame straightening"). This search capability is much like

8   a general web-search using a tool like AltaVista's or Inktomi's search engine and can

9   use the same search engine, but differs in that material being search is in a precisely

10   defined domain (group members), the information being searched is limited and

11   highly quality controlled (i.e. group directory entries), and has a precision rooted in

12   a precise vocabulary (the lexicon used in preparing the description).

13   Document repository. Any commercial web-enabled document repository

14   can be integrated into a group. Examples are Documentum and PC DOCs.   An

15   improved version offered specifically with the DVPN package was described above.

16   · Document security.   Within the document repository various tools can be

17   provided to protect the security of documents. These include (1) limiting access to

18   a document to certain people or groups; (2) only displaying the directory entry for

19   documents to people who can access it; (3) password protection; (4) encryption; (5)

20   secure archive in read only mode on a third-party machine; (6) time-limited access

21   and (7) a secure hash calculation.

22   All of the above are conventional except for time-limited access and the

23   secure hash calculation. Software for limiting access to a document to a certain

24   period is available from Intertrust, among others. A secure hash is a number that is

25   characteristic of the document calculated according to a precisely defined

26   mathematical algorithm. There are several secure hash algorithms, and implementers

27   can develop their won. They are "trap door" in nature. That is, the calculation can

28   be performed with reasonable effort, but the inverse of the function is

29   computationally intractable.   The classic example of a trap door function is

30   multiplication of very large prime number (on the scale of hundreds of digits). The

31   product can be calculated with relative ease, but factoring the product (the inverse

32   function) is very time consuming, making if effectively impossible with generally

33   available hardware. This method is used in public key encryption, but can be applied

49

SUBSTITUTE SHEET (RULE 26)

1 equally well in secure hash, though other trap door functions are preferred, in

2 particular, the one specified by the U.S. Department of Commerce as FIPS standard

3 180. Code to implement this standard can be developed from published algorithms.

4   Anonymous e-mail (described above with respect to the first embodiment);

5   Threaded dialogs. Threaded dialogs are a collection of messages addressing

6 a specific topic, added serially, not in real time. They are threaded in the sense that

7 new topics can branch off from a single topic, and topics can merge. According to

8 one embodiment, threaded dialogs differ from conventional news group functionality

9 in that (1) users can initiate new topics; (2) users can post a message to one topic,

10 then indicate that the message pertains to other topic as well; (3) browsers reading a

11 message may continue down the original thread or one of the alternates if other topics

12 are suggested.

13   Group newsletter creation tool. A newsletter creation tool can be used to link

14 columns provided by multiple users (and maintained as separate web documents) into

15 a whole through an integrating outline maintained by an "editor". The purpose of

16 the tool is to provide the look and feel of an attractive single document to a disparate

17 collection. To create the newsletter the editor generates an outline identifying an

18 author for each component and a layout. Art for the first page can be provided.

19 Through messaging, the authors are provided a link to upload their content. Content

20 is templated to include a title, date, a by line, one or more graphic elements, a

21 summary for the index, and text. The editor may allow documents to go directly to

22 "publication" or require impose a review and editing step.

23   Chat groups. Real time chat room software is widely available from many

24 sources including freeware and shareware.

25   Audio and videoconferencing. Commercially available tools for web-based

26 audio and video conferencing can be included in the group functionality. Examples

27 are Net Meeting and Picture Tel software.

28   FIG. 17 shows one possible user interface for selecting research tools to be

29 made available to group members. As shown in FIG. 17, various tools such as a

30 mortgage calculator, LEXIS/NEXIS access, news services, Valueline, and other

31 research tools can be provided by checking the appropriate box on the display. All

32 of these research tools are conventional and commercially available (via web-based

33 links and the like).

50

SUBSTITUTE SHEET (RULE 26)

1   FIG. 18 shows one possible user interface for selecting transaction engines

2   to be made available to group members. As shown in FIG. 18, many different types

3   of transaction engines can be provided to group members, including electronic data

4   interchange (EDI) ordering; online catalog ordering; various types of auctions; sealed

5   bids; bid and proposal tools; two-party negotiated contracts; brain writing (moderated

6   online discussion) and online Delphi (collaborative estimation of a numerical

7   parameter). The following describes various types of transaction engines in more

8   detail. Enhanced features (i.e., those that differ from conventional products) are

9   highlighted in gray text.

10   A. Order placement (online catalog) transaction engine

11   An order placement or online catalog engine allows the buyer to place an

12   order for a quantity of items at a stated fixed price, essentially ordering from an

13   online catalog. The catalog contains the description and specification of the

14   offerings. The catalog may be publicly accessible (Subtype 1a) or provided for a

15   specific customer (Subtype 1b). Prices are included in the catalog but may be

16   customer specific, may vary with quantity purchased, terms of delivery and

17   performance (e.g. cheaper if not required immediately). The catalog can represent

18   a single company's offering or an aggregate of the offerings from several companies.

19   The catalog can range from a sales-oriented web site designed for viewing by

20   customers, to a engine designed only accept orders sent via electronic data

21   interchange (EDI). Note that the catalog can be shopper oriented (i.e. designed to

22   sell) or a simple, machine-readable list of available items and prices. The following

23   describes in more detail steps that can be executed to create an online catalog:

24   1. Enter and maintain a framework for catalog

25   1.1. Enter / delete / edit categories. Categories are titles for groups of items, such

26   as "furniture" or "solvents"

27   1.2. Enter / delete / edit subcategories. Subcategories are categories within

28   categories, effectively establishing a hierarchy of products. Example:

29   furniture/dining room/tables.

30   1.3. Create groups of categories and subcategories (e.g. "see also....."). The

31   grouping allows a person browsing items to be referred to another category

32   that may contains items of interest. For example, someone may reach the

33   furniture/dining room/tables and then be referred to

51

1   furniture/office/conference room tables where other suitable tables may be

2   listed, or to furniture/dining room/chairs to buy chairs that make the table.

3   This cross-referencing transforms the hierarchical arrangement of

4   categories into a web.

5   2. Enter / edit / delete items in catalog by entering and updating the information

6   listed below.   The system allows users to enter this information and provides

7   basic quality assurance.

8   2.1. Catalog item number

9   2.2. Supplier part number(s)

10   2.3. Name of item

11   2.4. Description

12   2.5. Photos and drawings

13   2.6. Specifications (depends on item type).   Different items have different

14   specifications. For example, a computer printer can have color vs. black

15   and white, dots per inch resolution, paper size, etc.   In contrast to a fixed,

16   hard coded catalog, the specification section of the general purpose

17   catalog engine the user prepares the specification section by selecting

18   parameters from a list and then specifying a value for that parameter.

19   The parameter list contains values such as length, width, height, voltage,

20   color, resolution etc.   It is can be extended by the manager of the auction

21   environment.   A lister selects a necessary parameter (e.g. length, then

22   enter the value, such as 14"). The specification section is a concatenation

23   of individual specifications.

24   2.7. First available date

25   2.8. Last available date

26   2.9. Category (categories) into which the item fits

27   2.10.   Alternate suggestion(s) if product not available

28   2.11.   Related and associated products (e.g. printer supplies for a printer or other

29   household items with the same pattern.

30   2.12.   Additional information at the option of the individual or organization

31   listing the item.

32   3. Enter / update pricing information

33   3.1. Simple price.   The fixed prices is per item or per unit.   The price must

<center>52</center>

1   specify the

2   3.2. Pricing algorithm -- link to code for pricing algorithm

3   4. Take Orders

4   There are two variants: 4a: manual purchase in which a person browses a catalog

5   and selects and item for purchase and 4b: automated order in which a purchase

6   is initiated by an electronic message.

7   <u>Variant 4a: Manual Purchase</u>

8   4.1. Potential buyers access the catalog by drilling down through the category

9   / subcategory tree or

10   4.2. Buyers search fields in catalog to identify the appropriate item. The search

11   may examine the title, description, or any of the specification fields.

12   4.3. Display general information for item(s) meetings specifications

13   4.4. Allow user to modify search or to select specific item if the items displayed

14   to do not meet his requirements

15   4.5. Display detailed information for selected item

16   4.6. Display the fixed price or calculate price if prices is based on an algorithm.

17   The pricing algorithm may include parameter such as characteristics or

18   affiliation of the users (e.g. affiliated with a pre-negotiated discount

19   program) , delivery date and mode, and quantity.

20   4.7. Offer the option to purchase or search again if they choose not to purchase.

21   4.8. If the buyer opts to proceed with the purchase, then check the availability of

22   the item by linking to the sellers inventory system

23   4.8.1. If the item is available then execute an 'add to basket'. That is, place

24   it on a list of items designated for purchase.

25   4.8.2. If the item is not available, then execute the contingent response:

26   4.8.2.1.   Offer delivery at predicted date

27   4.8.2.2.   Terminate the sale, but offer to deliver or notify when next the

28   item is next available.

29   4.8.2.3.   Suggest alternate items

30   4.8.2.4.   Report 'sorry' and abort transaction

31   4.9. Offer option to purchase addition options

32   4.9.1. If offer is accepted, execute from step 4.1

33   4.9.2. If offer is not accepted, proceed with step 4.10

53

SUBSTITUTE SHEET (RULE 26)

1   4.10.   Conclude the transaction

2   4.10.1. Collect shipping information, offer options

3   4.10.2. Collect payment information

4   4.10.3. Validate payment

5   4.10.4. Summarize order

6   4.10.5. Obtain final authorization

7   4.10.6. Generate receipt

8   Variant 4b: automated order, done using an EDI (electronic data interchange)

9   message

10   4.1 Accept requests for item

11   4.2   Return price and confirmation of availability

12   Note that users may conduct transactions without employing EDI. It is

13   possible, however, for members to agree on a transaction EDI format either by

14   completing a template within the system or selecting a pre-established EDI format

15   from a library. This library can include formats developed by recognized standards

16   organizations (e.g. UNEDIFACT or ANSI) or formats developed specifically for an

17   industry or a trading environment. Once there is agreement on a format, transactions

18   can be initiated, concluded, and confirmed through the exchange of appropriate EDI

19   messages. As many commercial ordering, accounts payable, accounts receivable and

20   enterprise resource planning systems have an EDI interface the collaborative

21   environment should have the capability to forward the message to the order

22   fulfillment system.

23   B. English Auction Transaction Engine

24   In an English Auction, a single item is offered for sale to many buyers. The

25   auction can be open or limited to pre-qualified bidders. The buyers offer bids in turn,

26   each succeeding all prior bids. The highest bid received at any point in the auction

27   is visible to all buyers. The identity of the highest bidder may or may not be visible

28   to traders. Buyers may increase their bids in response to this information. Award

29   is to the highest bidder at the end of trading. The end of trading is reached when

30   there are no higher bids during an interval that may be formally defined or

31   determined by the manager of the auction at the time of execution.

32   There are two models for the access to the transactions. In the first model,

33   all buyers and sellers are members of the group. In the second model, all sellers are

1   members of the group, but buyers can include members and non-members. If non-

2   members are allowed to buy, the creator the transaction must enter a new URL for

3   buyers. This is a sub-URL of the main group URL. A registration process may be

4   established for the buyer URL.

5   In live auctions (as opposed to online) all traders are connected at the same

6   time, and the duration of the auction is brief – typically only a few minutes. In online

7   trading, it is not necessary for all of the bidders to be present (i.e. connected at the

8   same time). To distinguish between these two options they are designated (a)

9   concurrent (everyone bidding at the same time) and (b) batch (not everyone

10   connected simultaneously. The manager of the auction can set the minimum bid

11   and the minimum increment.

12   1. The first step in conducting an auction is to collect information on the items being

13   offered for sale. This is done online. The information collected includes:

14   1.1. Identity of seller. Note that the business rules of the auction may require

15   advance registration of sellers to verify their identity.

16   1.2. Descriptions, optionally including attachments and photographs, independent

17   certifications or appraisals, and anything else in digital form necessary or

18   useful in determining the value of the item.

19   1.3. Reserve price

20   1.4. Minimum increment

21   1.5. Time offered for sale

22   1.6. Time bidding is scheduled to end

23   1.7. Verify the seller's consent to the rules of the auction house regarding

24   delivery, payment, responsibility for non payment, etc.

25   2. If the business rule of the auction house is to require payment up front, collect

26   payment either by:

27   2.1. Debiting a deposit account

28   2.2. Charging to account for billing

29   2.3. Collecting online payment such as through a credit card.

30   3. Post information about auction, including:

31   3.1. Description of items to be auction

32   3.2. Auctions rules:

33   3.2.1. Qualification process for bidders

55

1   3.2.2. Time of bidding

2   3.2.3. Criterion for ending bidding – time between bids

3   3.2.4. Legal statement – responsibilities of buyer and seller, limitation of

4   liability

5   4. Execute qualification process (optional)

6   4.1. Admit bidders who are qualified based on past participation

7   4.2. Provide fill-in-the blank qualification form new bidders

8   4.3. Collect information

9   4.4. Conduct automated review or manual review

10   4.5. Inform prospective bidder of qualification or not

11 <u>Variant (a): concurrent auction</u>

12 5. Conduct Auction

13   5.1. Fifteen minutes prior to appointed time for auction, display "Welcome"

14   screen with space for qualified bidder to enter an alias or handle to be used

15   in the auction. Screen should have a description of the object. Show time

16   until auction starts. Auto refresh at 15 second intervals.

17   5.2. At appointed time, display the main auction page with the following

18   information:

19   5.2.1. Description / picture of item for auction stored in a separate, static

20   frame of the PC so that it does not need to be downloaded each cycle.

21   5.2.2. Current bid (initially the reserve price)

22   5.2.3. Suggested next bid (e.g. current + 3 * increment)

23   5.2.4. Button to accept suggested next bid

24   5.2.5. Field to enter bid higher than suggested next

25   5.2.6. Handle of the highest bidder

26   5.3. Refresh main auction page at 15 second intervals

27   5.4. Collect bids, either

28   5.4.1. Notice that the suggested bid was accepted

29   5.4.2. Bid higher than accepted bid

30   5.4.3. If new bid is lower than current highest, disgard

31   5.4.4. If higher than current highest then

32   5.4.4.1.   Log identity of highest bidder

33   5.4.4.2.   Update highest bid

56

1   5.4.4.3.   Update next suggested bid

2   6. If nobody accepts the suggested bid, then

3   6.1. Reduce suggested next bid

4   6.2. If accepted, resume normal sequence

5   6.3. If not accepted, reduce suggested next bid

6   6.4. If accepted, resume normal sequence

7   6.5. If not, begin close

8   6.6. "Going once ...", if response, resume normal sequence, else

9   6.7. "Going twice ..." if response, resume normal sequence, else

10   6.8. Done. Display closing screen

11 7. Settle with winning bidder, two models

12   7.1. Connect buyer to seller for direct settlement

13   7.2. Collect money from buyer, deduct fee, convey amount to seller

14 Variant (b): batch (i.e. time limited) auction

15   Conventional on-line batch (time limited) auctions are common. E-bay is

16 the most prominent example. This process description continues from step 4 of

17 the English auction description as the startup of the concurrent and batch auctions

18 are the same.

19 5. Conduct auction: Until closing time for an item:

20   5.1. On entry to system display the following for the potential buyer:

21   5.1.1. Latest listing

22   5.1.2. Categories

23   5.1.3. Search screen

24   5.2. On selection of categories:

25   5.2.1.   Execute dill down

26   5.2.2.   Retrieve count of items that meet criteria

27   5.2.3.   If more count is less than 25 (or other small number (n)

28   consistent with the layout of the screen) retrieve all items that meet

29   criterion

30   5.2.4.   If count is more than n, retrieve n auctions with nearest

31   expiration time

32   5.2.5.   Display link list to all items in list, sort order should be

33   auction with nearest deadline to most distant

57

1   5.2.5.1.   Item name

2   5.2.5.2.   Time till end of auction

3   5.2.5.3.   Highest current bid

4   5.2.6. On user selection of the item, display same information as above plus

5   5.2.6.1.   Description

6   5.2.6.2.   Photo (if any)

7   5.2.6.3.   Attachments (if any)

8   5.2.7.   If count is more than n, display further drill-down options as

9   well as item information above

10   5.3. Accept new bid through the display screen

11   5.3.1. Log bids in order, reject if bid is not higher than last high bid by

12   increment.

13   5.3.2. If bid is rejected, tell bidder that their bid is not sufficient

14   5.3.3. Update database recording highest bid, bidder, time of bid

15   5.3.4. Display screen to user to confirm that their bid is the highest

16   6. When the time limit is reached, determine if a new bid has been received in the

17   last 3 minutes (or other short time period). If so, extent the bidding time by 3

18   minutes (or other short time period) and execute step 5 with a new closing time.

19   7. When the time limit is reached, including all extensions under step 6, then

20   7.1. Email message to highest bidder that they won

21   7.2. Add transaction to completed deals

22   7.3. Update splash and add screens

23   7.4. Settle with winning bidder-- two models:

24   7.4.1. Connect buyer to seller for direct settlement

25   7.4.2. Collect money from buyer, deduct fee, convey amount to seller

26   C. Dutch Auction Transaction Engine

27   A Dutch auction, like a standard auction, involves the sale of a single item or

28 batch with fixed specifications. There is one seller, and many potential buyers. The

29 seller sets the prices, ideally higher than any buyer's maximum bid price. The

30 offered price is reduced by a fixed increment at fixed intervals until a buyer accepts

31 the price. The purchase goes to the first buyer in to accept the price. In the physical

32 world (as opposed to the online world), Dutch auctions are rarely if ever run

33 concurrently. In a live trading room, it could be difficult to determine which buyers

1   was first to commit to a price when several are willing to pay the same amount. The

2   Dutch auction is relatively simple to implement in an electronic environment. There

3   are, at present, no online Ducth Auctions of which the inventors are aware.

4   1. Enter and maintain a framework for catalog

5   1.1. Enter / delete / edit categories. Categories are titles for groups of items, such

6   as "furniture" or "solvents"

7   1.2. Enter / delete / edit subcategories. Subcategories are categories within

8   categories, effectively establishing a hierarchy of products.   Example:

9   furniture/dining room/tables.

10   1.3. Create groups of categories and subcategories (e.g. see also....).   The

11   grouping allows a person browsing items to be referred to another category

12   that may contains items of interest. For example, someone may reach the

13   furniture/dining   room/tables   and   then   be   referred   to

14   furniture/office/conference room tables where other suitable tables may be

15   listed, or to furniture/dining room/chairs to buy chairs that make the table.

16   This cross referencing makes transforms the hierarchical arrangement of

17   categories into a web.

18   2. Execute qualification process (optional)

19   2.1. Admit bidders who are qualified based on past participation

20   2.2. Provide fill-in-the blank qualification form new bidders

21   2.3. Collect information

22   2.4. Conduct automated review or manual review

23   2.5. Inform prospective bidder of qualification or not

24   3. Collect information on items to be auctioned and owners, including

25   3.1. Identity of seller

26   3.2. Descriptions, optionally including attachments and photographs, independent

27   certifications or appraisals, or other information necessary to establish the

28   value of the tiem

29   3.3. Categorization

30   3.4. Starting price

31   3.5. Increment, Interval for reduction

32   3.6. Minimum price

33   3.7. Obtain consent to rules (possibly as part of registration/qualification process)

59

1   3.8. Collect to conduct auction if item is

2   3.9. Calculate time to take item off auction by determining the number of steps

3   (intervals) necessary to reduce price from the starting price to the

4   minimum

5   3.10. Record all of the above information in the Dutch auction database

6   4. Cull expired options

7   4.1. Search database periodically for items where current time is later than time

8   to take item off auction (2.9)

9   4.2. Inform owner that item was not sold

10   4.3. Delete entry from database

11   4.4. Prompt for revised terms start of another auction, create new entry if user

12   takes option

13 5. When the buyer enters the system display a list of high level categories, a prompt

14   for search criteria, and/or a link to a search page.   Allow user to drill down

15   through categories or enter search parameters.

16   5.1. Retrieve count of items that meet criteria

17   5.2. If more count is less than 25 (or other small number (n) consistent with the

18   layout of the screen) retrieve all items that meet criterion

19   5.3. If count is more than n, retrieve n auctions with nearest expiration time

20   5.4. Display link list to all items in list, sort order should be auction with nearest

21   deadline to most distant

22   5.4.1. Item name

23   5.4.2. Time till end of auction

24   5.4.3. Current price:

25   5.4.3.1.   Retrieve starting price (SP) and increment (I$)

26   5.4.3.2.   Calculate number of intervals since start of auction (INT)

27   5.4.3.3.   Determine price = SP – (INT * $)

28   5.5. On click, display same information as above plus

29   5.6. Description

30   5.7. Photo (if any)

31   5.8. Attachments (if any)

32   5.9. The display screen should include a button that allows the buyer to purchase

33   the item at the selected price.

60

SUBSTITUTE SHEET (RULE 26)

1   6.   When the user clicks the "buy" button

2   6.1. Email message to highest bidder that they won

3   6.2. Add transaction to completed deals database

4   6.3. Settle with winning bidder-- two models:

5   6.3.1.   Connect buyer to seller for direct settlement

6   6.3.2.   Collect money from buyer, deduct fee if any for auction and

7   payment services, convey the remainder to seller.

8   D. Reverse English Auction Transaction Engine

9   In a reverse auction, there are multiple buyers to one seller.   Prices come

10   down rather than up. There are many variants of a reverse auction. The variant

11   discussed here is a reverse English auction. Reverse auctions have been

12   implemented on line in Open Markets.

13   The process for posting an item for bid and for qualifying bidders is the

14   same as for other auctions.   The difference here is that the buyer may optionally

15   set a maximum price.

16   1.   Accessing the list of items sought

17   Potential bidders access items sought by working through a hierarchy of

18   categories and subcategories or entering search criteria, as for other auctions.   A

19   list of items within the category/subcategory and/or meeting the search criteria

20   is displayed.   The user may then

21   1.1. Terminate the session on finding no suitable items ·

22   1.2. Revise the search criteria

23   1.3. Select an item on which to bid

24   2.   If the user selects an item on which that may wish to bid, detailed information

25   about the items is displayed. This item may include the following information:

26   2.1. Name

27   2.2. Seller

28   2.3. Description

29   2.4. Detailed specifications for items

30   2.5. Delivery requirements

31   2.6. Proposed terms

32   2.7. Current low bid

33   3.   If the user determines that they should bid, he accesses the bid entry screen from

61

1   the detailed description in Step 2 above. Making a bid consists of entering the
2   following information:

3   3.1. New, lower bid

4   3.2. Comments pertaining to any special terms, features, or conditions

5   3.3. Attachments containing relevant additional information and any
6   certifications required by the buyer

7   4. On receipt of bid, there are two options – either all bids are accepted, or bids are
8   accepted only after review of information by the buyer.

9   4.1. Case 1: all bids are accepted

10   4.1.1. New bid is checked to determine if it is lower than prior bid

11   4.1.2. If so, then

12   4.1.2.1.   bidder is notified that their bid is currently the lowest

13   4.1.2.2.   seller is notified of new low bid

14   4.1.2.3.   bid database is updated

15   4.1.3. If not, then

16   4.1.3.1.   Bidder is notified that their bid is not the lowest

17   4.1.3.2.   Bid screen is displayed so that bidder may lower bid

18   4.2. Case 2: bids are accepted after review by buyer

19   4.2.1. Buyer is notified of bid via email or online message

20   4.2.2. Buyer accesses complete information on the proposed bid through the
21   system

22   4.2.3. Buyer select accept bid or reject bid.

23   4.2.4. If bid is accepted, then

24   4.2.4.1.   Bidder is notified that their bid is currently the lowest

25   4.2.4.2.   Bid database is updated

26   4.2.5. If bid is not accepted, then

27   4.2.5.1.   Buyer enters reason for not accepting bid

28   4.2.5.2.   Bidder is informed that bid is rejected with reason stated
29   above

30   4.2.5.3.   Bidder may access the bid screen to revise offer

31   5. When time period has expired and there have been no bids within a short
32   specified interval, then

33   5.1. If at least one bid less than the maximum has been received, then:

1   5.1.1. Notify low bidder that their offer was successful

2   5.1.2. Add transaction to completed deals database

3   5.1.3. Settle with winning bidder-- two models:

4   5.1.3.1.   Connect or introduce buyer to seller for direct settlement

5   5.1.3.2.   Collect money from buyer, deduct fee if any for auction and

6   payment services, and convey the remainder to seller.

7   5.2. If no bid less than the maximum has been received, the

8   5.2.1. Notify buyer

9   5.2.2. Allow buyer to revise bid criteria

10   E. Sealed Bid Transaction Engine

11   In a sealed bid system, the buyer publishes or distributes detailed, fixed

12   specification to a number of potential bidders (who may or may not be

13   prequalified). Bidders submit binding bids by a specified deadline, in a specific

14   format that allows ready comparison.   The competitive bidding process is

15   distinguished from the bid and proposal process by the complexity of the

16   specifications and the bids.   In a simple competitive bid, competition among the

17   bidders is along one or two readily quantified dimensions (always including price)

18   and there is little or no room for variation in the form or specifications of the

19   offering.   Comparison of the bids is elementary.

20   The process for posting an item for bid and for qualifying bidders is the

21   same as for other transactions as is the method to identify items on which to bid

22   either using the hierarchy of categories and subcategories or a search engine.

23   1. If the user selects an item on which he may wish to bid, detailed information

24   about the items is displayed. This item may include the following information:

25   1.1. Name

26   1.2. Seller

27   1.3. Description

28   1.4. Detailed specifications for items including all information necessary to

29   prepare a bid

30   1.5. Bid instruction including specification for any documentation the buyer may

31   required with a bid (e.g. proof of bonding or license)

32   1.6. Notice of any fees for bid registration

33   1.7. Delivery requirements

63

1   1.8. Proposed terms

2   2. After review of the bid requirements, the user may choose not to bid or may enter

3   a bid.   The process for entering a bid consists of preparing a bid package,

4   including the price offered and any necessary supporting documentation. This

5   is done by completing an online form, with provision for attachments. The bid

6   is submitted through the system where it goes into a database of bids that are not

7   opened to the closing time for the bidding process.

8   3. At the closing time, all bid packages are conveyed to the buyer.

9   3.1. If there are no bids, the buyer is offered the opportunity to revise the request

10   for bids.

11   3.2. If there are multiple bids, the buyer reviews the bids and selects the lowest

12   priced qualifying bid. They buyer informs the seller and arranges payment

13   and delivery in accord with the terms stated in the bid package.

14   F. Order Matching Transaction Engine

15   In an order-matching system there are many potential buyers. Each posts

16   binding offer to buy (bid amount) or sell (asked amount).   The process proceeds in

17   real time. The order matching system constantly compares bid and asked and, when

18   a match is found within a specified spread, the deal is concluded. No accepted offer

19   can be repudiated, but offers may be withdrawn before a deal is consummated. The

20   strike price is posted so that buyers and sellers can modify their offerings in real time.

21   The items traded are fungible so that price is the only decision. For the market to

22   operate efficiently the items traded must be tightly defined and the terms of sale must

23   be fixed and determined in advance. This is typically done by the operation or an

24   exchange, with the order-matching engine operating in the background. To insure

25   that the items traded are well defined, and the terms of sale are rigid example of an

26   order matching process in stock trading on an exchange.

27   Users of an order-matching engine are all potential buyers and seller. They

28   are qualified in advance using a process like that outlined by for auction with the

29   extension that deposit accounts are frequently required given the speed of

30   transactions in exchange environments.

31   1. Establish and maintain items to be traded. All functions in this category are

32   reserved to the manager of the exchange or a designee.

33   To add (i.e. "list" and idem), enter

1   1.1. Unique item number or symbol

2   1.2. Description of item (e.g. Sears Class A Common Stock)

3   1.3. Terms and conditions ownership (e.g. who can own) if any

4   1.4. Trading units (e.g. shares, blocks, etc.)

5   1.5. Additional information as required by the rules of the exchange

6   To delete (i.e. "delist" and item)

7   1.6. Select the item to be deleted

8   1.7. Confirm deletion

9   2. On entry to the system, potential buyers and sellers can review the price of the

10   last transaction of any item, either through a list or a search by item name or

11   symbol. The current highest asked and lowest bid price are also shown.

12   3. An offer to sell is posted by entering the following information:

13   3.1. Item number or symbol

14   3.2. Quantity offered

15   3.3. Proposed price ("asked")

16   3.4. Seller

17   3.5. Offers may be revise at any time prior to consummation of a deal

18   4. An offer to buy is posted by entering the following information

19   4.1. Item number or symbol

20   4.2. Quantity offered

21   4.3. Proposed price ("asked")

22   4.4. Buyer

23   4.5. Offers may be revised at any time prior to consummation of a deal

24   5. Offers to buy and sell are constantly reviewed by the software. When there is an

25   offer to buy and sell at a price within a preset difference. When prices match,

26   buyers and sellers are notified of the transaction, and the transaction is recorded.

27   The display of the last transaction price, the highest bid and the lowest asked

28   price is updated.

29   6. The transaction is conveyed to the backend accounting system of the exchange.

30   G. Bid and Proposal

31   The bid and proposal process is typically used for procurement of large or

32   complex products or services, in which cost is not the only factor. Cost must be

33   weighed against the buyer's assessment of the quality and suitability of an offering

<center>65</center>

1   and the ability of the bidder to deliver the product or perform the specified services. .

2   The bid and proposal process is conducted between one buyer (possibly

3   representing a consortium) and many potential sellers, sometimes organized into

4   teams.   The buyer issues specifications that may be general or highly specific, brief

5   or very lengthy.   The specifications may be distributed freely or to a list of qualified

6   buyers.

7   With physical RFPs, the size and the associated cost of distribution make it

8   common practice to advertise the availability of the RFP first, sending copies only

9   to those that request it. Frequently, the requestors are required to supply information

10   to establish their qualifications to bid.   While cost is not an issue in electronic

11   dissemination of RFPs, the model of advertising prior to distribution is still useful in

12   managing the qualification process.   This is addressed as variant (a) is this

13   description.   Variant (b) requires no prequalification.

14   In a competitive bid on fixed requirements (sealed bid or auction), there is

15   typically very little communication between buyer and seller between publication of

16   the request and submission of the bids.   The requirements are comparatively simple,

17   clear, and unambiguous.   In contrast, the bid and proposal process may involve

18   considerable communication between buyer and seller.   The process may begin with

19   a bidders' conference to answer questions about the requirements.   Additional

20   questions from bidders may be accepted, though not all need be answered.

21   Questions and answers may be made available to all bidders or the response may be

22   in private.   This dialog is crucial for two reasons.   First, it helps the bidders

23   understand the requirements and to be responsive in their bids.   Second, it is not

24   unusual for the bidders' questions to identify some point of ambiguity, error, or

25   contradiction in the specifications, leading to a modification of the RFP.   The

26   diverse perspectives of the bidders, and the close attention required on their part to

27   prepare a bid inherently provides an excellent review of the RFP.

28   The initial phase of the RFP process concludes with submission of the bids,

29   but this is far from the conclusion of the process.   Commonly, questions arise from

30   the review of the proposals.   These may relate to a specific submission or have

31   broader implications, leading to modification of the requirements.   The list of

32   bidders can be culled to the best candidates. These are asked to answer questions

33   about their proposals and to provide additional and clarifying information.

66

SUBSTITUTE SHEET (RULE 26)

1   The process described here is built around the document repository described.

2   elsewhere in this application. Through this process of refinement, the list of bidders

3   is narrowed to one or two with whom a contract is negotiated. The process of

4   negotiation is addressed as a separate transaction type (Negotiation Engine) as it may

5   be conducted without the bid and proposal process.

6   Variant (A): with pre-qualification

7   1. Software supports the user in creating a web site for the proposal process.

8   Initially this site manages the process for requesting the request for proposal

9   (RFP), qualifying bidders, and disseminating the RFP.

10   2. Supported by the system software, the bidder creates and RFP advertisement by

11   2.1. entering a summary of the RFP.

12   2.2. entering a summary of the information needed to qualify as a bidder or

13   2.3. attaching a form (HTML web page or template for paper form) for entering

14   qualifying information

15   3. The RFP advertisement includes file transfer software for uploading qualifying

16   information to the repository.

17   4. Disseminate RFP advertising

18   4.1. Post on public bulletin board or

19   4.2. Disseminate via mail to selected users

20   5. When users access the system, issue them an encryption key and PIN to be used

21   for subsequent uploads and communications to verify their identity.

22   6. Receive requests for RFP in repository

23   6.1. Prompt for key

24   6.2. Encrypt submission

25   6.3. Upload

26   6.4. Generate receipt – should include an authentication number

27   7. Disseminate RFP to selected user, either:

28   7.1. Attach to return Email or

29   7.2. Post the RFP in a repository from which qualified prospective bidders may

30   download the file. If the repository model is used, provide notice of the

31   posting via email including any necessary PINs and codes to access the

32   repository

33   7.3. When a prospective bidder downloads an RFP, issue an encryption key to be

67

1   used in submitting proposal

2   8. The RFP site also includes a page through which prospective bidders can submit

3   questions. Questions and answers are posted to the site.

4   9. Updates to the schedule and amendments to the RFP are posted to the site

5   10. All access to the site is recorded to verify that prospective bidders have received

6   critical information. Direct contact may be used when it is determined that a

7   bidder had not accesses the site since critical new information was posted.

8   11. Bidders prepare their proposal and then upload them to a repository for proposals

9   using software built into the proposal site.

10   11.1.   Prompt for key

11   11.2.   Encrypt submission

12   11.3.   Upload

13   11.4.   Generate secure hash number to prevent tampering with the

14   submission

15   11.5.   Generate receipt including secure hash number and authentication

16   code

17   12. After initial proposals are received, the process moves into a phase commonly

18   termed the "best and final process" in which the proposals are reviewed, the list

19   narrowed, and the proposals refined.

20   12.1.   Create separate secure environment (i.e. web site with repository) for

21   each respondent

22   12.2.   Exchange materials through repository (described elsewhere in this

23   filing)

24   12.3.   Records and receipt each access

25   12.4.   Generate key for revised proposal

26   12.5.   Receive proposal using process in 11

27   12.6.   Repeat from step 11 as many times as necessary

28

29   The remainder of the process is completed as a negotiated deal, described below.

30   Variant B: no pre-qualification:

31   Proceed as above, beginning with Step 6 and not requiring a key for download of the

32   RFP.

1   <u>H. Negotiation Deal Engine</u>

2   An engine for negotiating a deal can be built around the capability of the

3   system to create a temporary virtual private network through the web. A temporary

4   network is created for the negotiation. Access to the network is limited to the parties

5   of the negotiation, their advisors and counsel, and, potentially, arbitrators and

6   regulators. The members of the negotiating environment have access to the complete

7   set of tools described in this filing including those for communications (email,

8   anonymous mail, online chat, threaded dialogs, and audio and video collaboration),

9   the library of standard contract instruments, the tools for document signature and

10   authentication, and the document repository. Using these tools in a secure

11   environment they can negotiate, close, and register a deal.

12   FIG. 19 shows one possible user interface for selecting participation engines

13   to be made available to group members. The term "participation engine" refers

14   generally to collaboration tools that provide features beyond merely communicating

15   among group members. Various services such as an on-line survey tool, a DELPHI

16   model tool; brain writing tool; and real-time polling can be provided.

17   <u>A. Online Survey</u>

18   In online polling or surveying, the person creating the poll uses and

19   automated tool (new to this application) to build simultaneously an online

20   questionnaire and a database to collect the results. The user builds the questionnaire

21   by entering a series of questions and an associated data collection widget for each.

22   The polling tool builds the database and the data entry screen. The data entry

23   screen consists of two columns. The left column is a series of questions. The right

24   column is the data entry tool appropriate to the question. Various data entry tools

25   can be provided to respond to the query, including such things as:

26   1.   yes / no radio buttons

27   2.   true / false radio buttons

28   3.   slider with scale from 1-5, 1-10, etc.

29   4.   fill-in-the-blank text box

30   5.   numeric field

31   6.   multiple check boxes (e.g. strongly disagree, disagree, agree, strongly

32   agree)

33   Other data entry types may be added.

69

1   As each question / data collection widget is added, the polling tool creates the .

2 database. The database includes one record per data collection form. Creating the

3 database structure simply means adding one new field to each record definition for

4 each question. The type of data collection widget defines the format of the field, as

5 follows:

6   1.   yes / no radio buttons: one character field, limited to "y" or "n"

7   2.   true / false radio buttons: one character field, limited to "y" or "n"

8   3.   slider: real number field, with appropriate range check

9   4.   fill-in-the-blank text box: text box

10   5.   numeric field: real number or integer

11   6.   multiple check boxes: integer field with range check from 1 to number of

12   boxes

13 Every data entry screen provides a "save" and "cancel" button. Save writes to the

14 database. Cancel exits the entry screen without saving.

15   The survey, once composed as described above exists as a web page. This

16 page can be embedded in web applications. It can be made available on a site

17 available to the entire Internet, on an Intranet, or in a dynamically created

18 environment. Alternatively, it can be distributed via e-mail. When the form is

19 completed, the submit button transmits the value entered to the database that is

20 created at the time the form is generated. Access to the database is controlled by the

21 rules of the database system. It may be limited to the individual who creates the

22 survey form and database, but it may be accessible other users in the survey

23 developers organization, as determined by the database administrator. Distribution

24 of the result of the analysis is at the discretion and control of the individual managing

25 the survey. This manager may be the individual who creates the survey, but the

26 actual creator may be acting on behalf of the survey manager. Results may be kept

27 private, posted to the Internet, and intranet, or a collaborative environment,

28 distributed via e-mail within an organization, or, if the information is available, sent

29 via e-mail to the participants in the survey.

30   B. Online Delphi Engine

31   The online Delphi engine allows real-time collaboration in estimating or

32 predicting an outcome that can be expressed numerically. For example, the method

33 can be used to develop a consensus forecast of grain prices. The method has been

70

1   in used since the 1970s, but has not previously been adapted to online processes.

2   One possible method is as follows:

3   1. Establish the session

4   1.1. Within an online community, the moderator of the session creates the brain

5   writing session by entering the following information:

6   1.1.1. Name of moderator

7   1.1.2. Title of the session

8   1.1.3. Description of the session

9   1.1.4. Background reading as references or attachments

10   1.1.5. Start date for the session

11   1.1.6. Scheduled end for the session

12   1.1.7. Access to the session:

13   1.1.7.1.   URL for access

14   1.1.7.2.   Open to all or invitees only for observation

15   1.1.7.3.   Open to all or invitees only for participation

16   1.1.8. Payment information if required

17   2. Optionally, the session may be advertised on line

18   3. If the session is private, invitations with logon keys must be distributed via email,

19   actual mail, or download.

20   4. Optionally, the moderator may run on online applications and qualification

21   process

22   5. Prior to the start of the session, the moderator must describe precisely the value

23   to be estimated. The definition must be completely unambiguous.

24   6. Each participant connects at the start of the session. On connecting, they question

25   is posed (e.g. "What will be the price of West Texas intermediate oil in

26   December?")

27   7. Each participant enters a number a brief (1 paragraph maximum) explanation of

28   their reasoning.

29   8. When the participant is done entering their estimate, they click "Done".

30   9. Each participant's estimate and explanation is recorded.

31   10. Each participant then sees the summary screen.

71

1   11. Estimates are arrayed graphically from top to bottom of the screen, from lowest .

2   to highest. The value is stated as is the associated comment, but the source of

3   the comment is not revealed.

4   12. Participants can review the estimates and comments, send an anonymous message

5   to the author or any comment, or amend their answers.

6   13. The session terminates when the time expires, or when the moderator determines

7   that there it is no longer appropriate to continue. The operator may determine

8   this is based on declining participation or, if participation is high, the moderator

9   may extend the deadline.

10 14. Participants and observers may access the final display of estimates, again

11   arrayed from top to bottom, lowest to highest.

12   C. Brain Writing

13   Brain writing is a variant of a method for facilitated group discussion termed

14 brainstorming.   The objective of brainstorming is to maintain the focus of the

15 discussion while encouraging creative input and recognizing the contributions of all

16 members of the group. It seeks to avoid problems with a few individuals dominating

17 the discussion, with junior staff deferring to senior staff, and with new ideas being

18 abandoned before than can be developed fully. Brain storming has been commonly

19 used since the late 1960s. Brain writing is a more intense method that relies on joint

20 writing rather than discussion. What is presented here is adaptation of that method

21 to an online environment. It is believed to be the first such adaptation.

22 1. Establish the session

23   1.1. Within an online community, the moderator of the session creates the brain

24   writing session by entering the following information:

25   1.1.1. Name of moderator

26   1.1.2. Title of the session

27   1.1.3. Description of the session

28   1.1.4. Background reading as references or attachments

29   1.1.5. Start date for the session

30   1.1.6. Scheduled end for the session

31   1.1.7. Access to the session:

32   1.1.7.1.   URL for access

33   1.1.7.2.   Open to all or invitees only for observation

72

SUBSTITUTE SHEET (RULE 26)

1   1.1.7.3.   Open to all or invitees only for participation

2   1.1.8. Payment information if required

3   2. Optionally, the session may be advertised on line

4   3. If the session is private, invitations with logon keys must be distributed via email,

5   actual mail, or download.

6   4. Optionally, the moderator may run on online applications and qualification

7   process

8   5. Prior to the start of the session, the moderator must list some number (typically

9   5-10) of questions or hypotheses to be explored. (e.g. " Our company should

10   create a spinoff to develop and commercialize the new breast cancer vaccine")

11   This may be done by the moderator alone, in consultation with the   participants,

12   or with other outside the session.

13   6. Each question or hypothesis becomes a "Card".

14   7. Participants may enter the session any time after the start.   A password may be

15   required if the session is not open.

16   8. On entry into the system, a user if given a card at random.   The card consists of

17   the initial question or hypothesis plus all comments entered on the card by other

18   participants.

19   9. After reviewing the card, the participant may add his or her own comments to the

20   bottom.   After entering comments, the participant clicks "Done" to return the

21   card to the pile.

22   10. When a participant returns a card to the pile, they received another card, chosen

23   at random (preferably) or selected by the user. This process continues until the

24   opt to exit. They may reenter at any time up to the conclusion of the session.

25   11. When a card is returned to the pile, it is become available for assignment to the

26   next participant.   The card includes the additions of the most recent participant.

27   12. A participant may opt to return the card without addition if he or she has nothing

28   to add.

29   13. Participants may create new cards when new ideas come to mind.   These are

30   treated in exactly the same way as original cards.

31   14. Observers may view any card but may not add to them.

32   15. The moderator may limit participation to a set number at any time so that there

33   is a sufficient number of cards to keep the participants fully occupied.

73

SUBSTITUTE SHEET (RULE 26)

1   16. The session terminates when the time expires, or when the moderator determines.

2   that there it is no longer appropriate to continue. The operator can determine this

3   based on declining participation or, if participation is high, the moderator may

4   extend the deadline.

5   17. The raw cards are distributed at the conclusion to all participants. The moderator

6   or another individual is charged preparing a summary and arranging follow-up.

7   FIG. 22 shows one possible scheme for storing brain card writing data

8   elements. In accordance with one embodiment, each brain writing card comprises

9   a data structure including the following elements:

10   1.   Brain writing session number: Serially assigned number to differentiate

11   brainwriting sessions.   A session is the set of all cards pertaining to a

12   particular topic.

13   2.   Card number: A Serially assigned sequence number

14   3.   Initial Comment : The question or comment used to initiate the discussion

15   (e.g. "SAIC should purchase a company that produces Internet server

16   software"

17   4.   Date and time card started

18   5.   Date and time card closed

19   6.   Comments: A collection (i.e. a set of unlimited length) containing the

20   comments added by participants in the brainwriting session.

21   7.   Date of additional comment: Date and time that each additional comment

22   was added.

23   8.   Commenter: Name or user ID of the person adding each additional

24   comment. Ideally, brainwriting should be anonymous to encourage open

25   dialog. Accordingly, this field may be omitted from an implementation.

26   Some organizations, however, may wish to track this information

27   without making it visible to users, or in some cases to attribute comments.

28   When the user has finished defining the group and specifying its functions,

29   environment generator 1201a (FIG. 12) creates an environment accessible to the

30   group members and including the functions specified during the environment

31   definition process. As shown in FIG. 20A, for example, a web page can be created

32   for the newly created environment, including those functions that were selected by

33   the user that created the group. All group members are notified of the existence and

74

1   location of the environment, and each group member can use the functions provided

2   in the environment to collaborate on a project or conduct business.

3   FIG. 20B shows what an environment might look like to a group member

4   after entering the environment. As shown in FIG. 20B, for example, a news banner

5   announces the latest news for the group. Additionally, specific communication tools,

6   research tools, transaction engines, and participation engines are made available to

7   group members, which can be executed by appropriate mouse clicks in accordance

8   with the inventive principles. According to various inventive principles, each tool

9   shown on the web page is accessible through a hyperlink to a web-based program that

10   performs predefined functions as set forth above. For example, clicking on "online

11   catalog" would link the group member to a web page that implements an online

12   ordering engine as described previously. Users can navigate through the various

13   tools using conventional web browser features (i.e., forward, backward, etc.). It may

14   be desirable to implement some or all of such software using server-side scripting or

15   other similar means consistent with the system configuration of FIG. 12.

16   FIG. 21 shows how environment generator 1201a can create multiple

17   environments including virtual private facilities, which can be implemented through

18   web pages that contain hyperlinks to functions available to members of each group

19   or environment. An environment definition software component 2106 implements

20   steps 1101 through 1103 of FIG. 11 in order to create one or more environments

21   2107. (In one embodiment, each group can also be provided with a copy of an

22   environment generator 2106 in order to create sub-groups that draw on the

23   applications and directory structure created for the group). As a user identifies group

24   members and selects functions to be provided for the environment in which the group

25   will collaborate, environment definition component 2106 stores information relating

26   to the selected members and functions in databases. Each environment can include

27   a web page (not shown in FIG. 21) and directories, tools and other applications

28   specific for each created group.

29   Based on user selections of the type illustrated in FIGS. 13 through 19,

30   environment generator 2106 creates an environment 2107 containing one or more

31   web pages with links to the selected tools. Environment generator 2106 retrieves

32   information from various information sources including a directory of

33   communication tools 2101 (e.g., including descriptions of tools and URL/IP

75

1   addresses of web applications to set up each communication tool); directory of

2   transaction engines 2102 (e.g., including descriptions of transaction engines and the

3   URL/IP addresses of web-based applications to set up each transaction engine);

4   directory of research tools 2103 (similar to above); list of global data objects 2104

5   (e.g., a dictionary of data elements from which the directory of each group can be

6   composed); and a directory of applications 2105 (e.g., a description of available

7   applications and URL/IP addresses of pages to set up access to applications).

SUBSTITUTE SHEET (RULE 26)

1   <u>WE CLAIM:</u>

2   1. A method of negotiating a deal over a network of computers, the network

3   including at least one or more computers connected to the Internet, the method

4   comprising the steps of:

5   (1) posting, on an electronic list that can be viewed over the Internet,

6   information regarding one or more offers to form a contract;

7   (2) posting on the electronic list one or more responses to the one or more

8   offers;

9   (3) researching the one or more responses to determine whether they satisfy

10   one or more contract criteria;

11   (4) negotiating over the network between at least two parties to accept or

12   modify one or more of the responses; and

13   (5) electronically signing a document to consummate the contract.

14   2. The method of claim 1, wherein step (1) comprises the step of displaying

15   offers and responses in a parent-daughter spatial relationship on a computer display.

16   3. The method of claim 1, further comprising the step of sorting the one or

17   more offers and one or more responses according to a user-selected sort order.

18   4. The method of claim 1, wherein steps (1) and (2) are done anonymously,

19   such that each party to the contract cannot determine the identity of the other party

20   to the contract.

21   5. The method of claim 4, further comprising the step of simultaneous

22   revealing the identity of each party prior to step (5).

23   6. The method of claim 4, wherein steps (1) and (4) comprise the step of

24   sharing a single anonymous e-mail alias among a plurality of users.

25   7. The method of claim 1, further comprising the steps of:

26   (6) registering keywords with an electronic agent that monitors the one or

27   more offers and providing an e-mail address to be notified upon a keyword match;

28   and

29   (7) in response to the electronic agent detecting the keyword match,

30   transmitting a message to the e-mail address provided in step (6).

31   8. The method of claim 1, wherein step (2) comprises the step of clicking on

32   a hyperlink linking the information posted in step (1) to a reply card.

33   9. The method of claim 7, wherein step (2) comprises the step of requiring

77

SUBSTITUTE SHEET (RULE 26)

1   the submission of certain information before the reply card will be accepted.

2   10.   The method of claim 1, wherein steps (3) and (4) are performed a

3   plurality of times for a single contract, such that modifications are made to the one

4   or more responses.

5   11.   The method of claim 1, further comprising the step of electronically

6   registering a plurality of entities that have signatory authority and correlating the

7   registered entities with one or more documents to which signatures can be affixed.

8   12.   A method of displaying information on a computer display,

9   comprising the steps of:

10   (1) displaying a first plurality of graphical objects each having a shape of a

11 file folder comprising a folder face and a labeled tab, wherein the first plurality of

12 graphical objects are stacked in a cascading arrangement; and

13   (2) in response to user activation of a "flip" tab, changing the graphical

14 objects displayed in step (1) to show a second plurality of graphical objects each

15 having a shape of a file folder comprising a folder face and a labeled tab,

16   wherein each of the first and second plurality of graphical objects can be

17 brought to a foreground position in front of other graphical objects by clicking on

18 a corresponding labeled tab.

19   13.   The method of claim 12, wherein each of the first and second plurality

20 of graphical objects has associated therewith one or more functions displayed on

21 the folder face thereof, wherein user can activate the one or more functions by

22 clicking thereon.

23   14.   A method of creating a user-defined networked environment across a

24 plurality of computers without requiring system administrator-level privileges,

25 comprising the steps of:

26   (1) creating a group by providing a group identifier, a group description,

27 and by specifying a plurality of group members entitled to use the user-defined

28 networked environment;

29   (2) selecting a plurality of web-based communication, collaboration, and

30 transaction tools from a list of available tools, wherein the selected tools are to be

31 made available to the plurality of group members specified in claim 1; and

32   (3) through the use of computer software, automatically creating the user-

33 defined networked environment by creating a web page accessible to the plurality

78

1   of group members selected in step (1), wherein the web page provides access to

2   the plurality of tools selected in step (2).
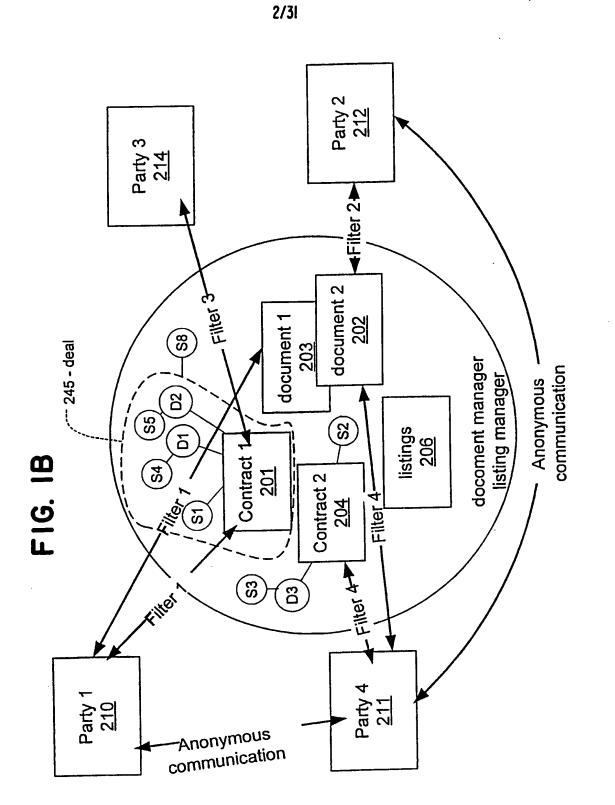
3   15. The method of claim 14, wherein step (1) comprises the step of

4   inviting a plurality of individuals to join the group by transmitting an invitation to

5   prospective group members.

6   16. The method of claim 14, wherein step (1) comprises the step of

7   advertising an invitation to join the group by posting an advertisement for

8   prospective group members, wherein at least some of the prospective group

9   members are unknown to the user creating the networked environment.

10   17. The method of claim 14, further comprising the step of screening

11   prospective members that respond to the advertisement in order to determine

12   whether they should be added to the group.

13   18. The method of claim 14, further comprising the steps of electronically

14   collaborating among group members using the user-defined networked

15   environment.

16   19. The method of claim 14, further comprising the step of destroying the

17   user-defined networked environment when it is no longer needed.

18   20. The method of claim 14, wherein step (2) comprises the step of

19   selecting a transaction engine that implements an auction to members of the

20   group.

21   21. The method of claim 14, wherein step (2) comprises the step of

22   selecting a transaction engine that implements an on-line electronic survey

23   comprising survey questions that are to be answered electronically by survey

24   participants.

25   22. The method of claim 14, wherein step (2) comprises the step of

26   selecting a transaction engine that implements a bid-and-proposal tool that permits

27   group members to electronically submit bids on one or more proposals.

28   23. The method of claim 14, wherein step (2) comprises the step of

29   selecting an online ordering engine that permits group members to electronically

30   order goods or services in the user-defined networked environment.

31   24. The method of claim 14, wherein step (2) comprises the step of

32   selecting an Electronic Data Interchange (EDI) compatible interface that executes

33   electronic commercial transactions between two or more group members.

79

SUBSTITUTE SHEET (RULE 26)
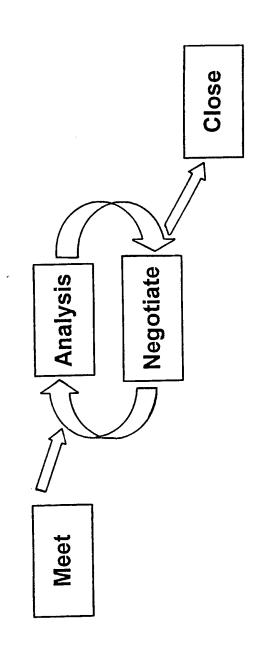
1   25. The method of claim 14, wherein step (2) comprises the step of a

2 selecting an electronic brain-writing tool that permits participants to brainstorm

3 using electronic idea cards.

4   26. A system for implementing a user-defined networked environment

5 that can be created without the need for system administrator-level privileges,

6 comprising:

7   a plurality of networked computers that communicate using Internet

8 Protocol;

9   a plurality of web browsers executing on the plurality of networked

10 computers;

11   a database that stores information concerning the user-defined networked

12 environment; and

13   a computer program executing on one or more of the plurality of

14 networked computers, wherein the computer program performs the steps of:

15   (1) permitting a user to create a group comprising a plurality of group

16 members;

17   (2) permitting the user to select a plurality of web-based communication,

18 collaboration, and transaction tools from a list of available tools, wherein the

19 selected tools are to be made available to the plurality of group members; and

20   (3) automatically generating a web page accessible to the plurality of

21 group members, wherein the web page provides access to the plurality of tools

22 selected in step (2) to the plurality of group members.

FIG. IA

FIG. IB

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
236
239
240
313
312

FIG. 3

\#
Date
Title
Market
Action

Description

— 301

Attachment 1

Attachment 2 —— 304

Attachment 3   — 302

Instructions for responding

Reply   — 303

FIG. 4

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:

O Small Business

O Minority owned business

O Woman owned business

FIG. 5

International | Deals | Communications | Recyclables | Proposals | Research | Job Bank | Products & Services

US Domestic

Offers

Documents

4:11A-

Opcom.GIF (2100bytes)

This is a private folder (1 document) | New Document

letter re lease sales.doc | Download | Details | Forward | Delete — X

demonstration for jeff (1 document) | Shared | New Document

foy streetman.doc | Download | Details | Forward | Check-In — U | Delete — S

Test (empty) | Delete | New Document

Demo for Rowan (1 document) | New Document

In Box (1 document) — I

letter re lease sales.doc | Download — P | Shared | Details | Forward | Delete

zelenica repowering (empty) | Shared | Delete | New Document

just fooling around (empty)

Create a New Folder

FIG. 6

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

International   US Domestic   Proposals   Job Bank   Products & Services

Offers   Deals   Communications   Recyclables   Documents   Research

4:11A

SAIC.
Science Applications
International Corporation
An Employee Owned Company
Copyright SAIC,1998. All rights reserved

FIG. 7

Trade Number	Date Created	Trade Title
357	Sep 22 1998	mmmmmmmmmm
356	Sep 22 1998	World Trade Center
353	Sep 22 1998	Florida Windstorm Coverage
352	Sep 22 1998	vvv
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

FIG. 8A

FIG. 8B

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:

Names: John Q. Client
            Craig Miller   Delete   None : None

FIG. 9

Flip

410

400

A   B   C   D   E

430

431

432

FIG. 9A

FIG. 10

410

Flip

435

J

I

437

436

H

400

G

F

FIG. 10A

FIG. 11

FIG. 12

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 C:\WINDOWS\DESKTOP\Figure 13(a).htm

Go   Links"

CREATE NEW ENVIRONMENT

User Name: JoeUser

Password: SloppyJoe

Next

Done   My Computer

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\WRITE\WORK\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. 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\WRITE\WORK\0479\75982\Figure 13(c).htm

Links

Go

IDENTIFY GROUPMEMBERS

Name of Group: JOE'S HOMEWORK

o Select from list or provide contact

o Compose invitation

o Compose advertisement

Next

Done   Local Intranet

FIG. 14A

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\0479\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. 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\0479\75982\Figure 14(b).htm

INVITE GROUP MEMBERS

INVITE GROUP MEMBERS

1. Enter address: bob@biqisp.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

FIG. 14C

CREATE NEW ENVIRONMENT - Microsoft Internet Explorer

File " | Back Forward Stop Refresh Home | " | Links " | Address TWORK\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

Qualify

Click for qualifications:

Done

Done | Local Intranet

FIG. 15

H:\USERS\WRIGHT\WORK\0479\75982\Figure 15.htm - Microsoft Internet Explorer

File " Back Forward Stop Refresh Home " Links " Address HT\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.

More!

Click here for more information and to submit qualifications

TODAY"S NEWS

Eskimos eating less ice cream!

Hulk runs for President!

Stock market hits new high!

Done   Local Intranet

FIG. 16

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\0479\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. 17

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. 18

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. 19

FIG. 20A

SELECT COMMUNICATIONS TOOLS - Microsoft Internet Explorer

File » | ⇦ ▾ ⇨ ▾ 🛇 ⟳ 🏠 » | Links » | Address 🔗 T:\WORK\0479\75982\Figure 20(a).htm ▾ ⤤Go
Back Forward Stop Refresh Home

WELCOME TO JOE'S RESEARCH PROJECT!

Please log in:

User Name:
Password:

Send a message to the group
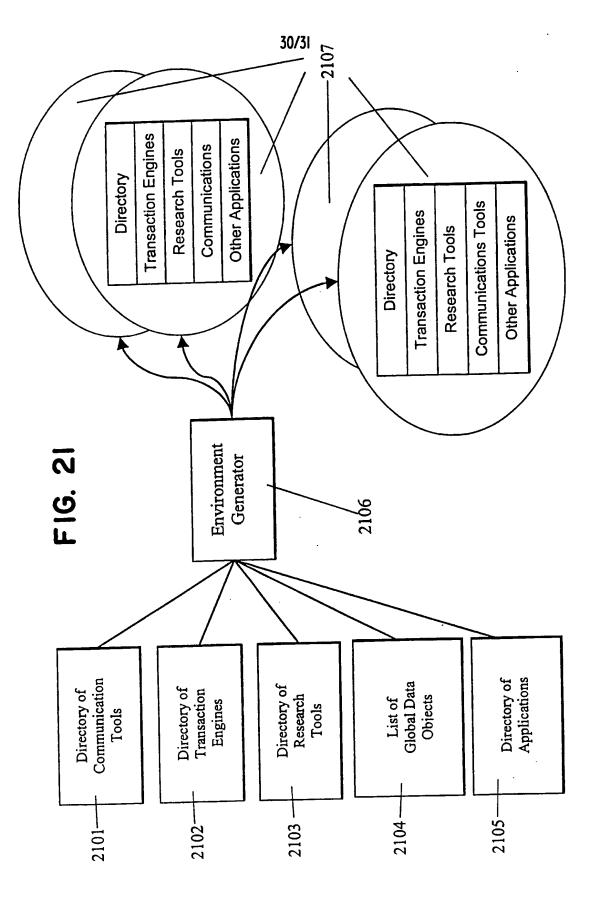
Cancel

Done | Local Intranet

FIG. 20B

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   Communication tools
Yellow Pages
Newsletter

BC News   Research Tools
Washington Post

Online Catalog   Transaction Engines
Bid and Proposal

Online Survey   Participation Engines
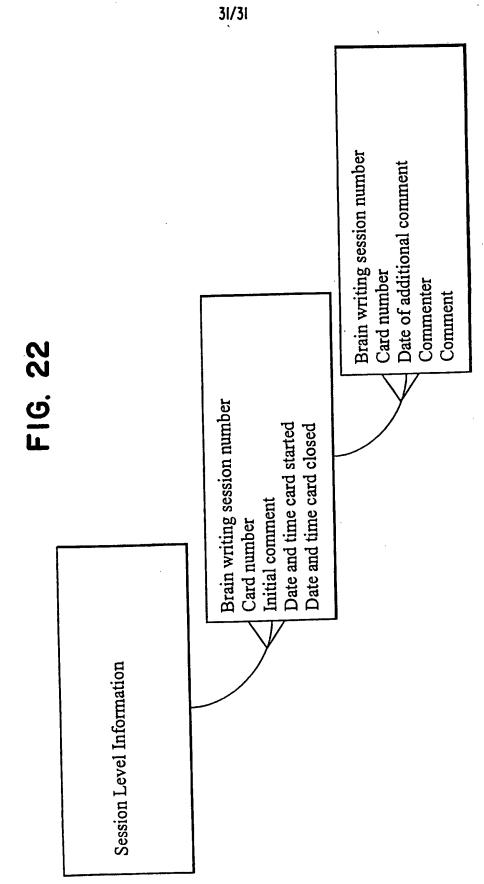
Done   Local Intranet

FIG. 21

30/31 2107

Directory
Transaction Engines
Research Tools
Communications
Other Applications

Directory
Transaction Engines
Research Tools
Communications Tools
Other Applications

2106

Environment
Generator

2101 — Directory of Communication Tools

2102 — Directory of Transaction Engines

2103 — Directory of Research Tools

2104 — List of Global Data Objects

2105 — Directory of Applications

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

A 12-18-07

10/714,849

PTO-1449 (Modified)	ATTY. DOCKET NO. 000479.00111	SERIAL NUMBER TBD
U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	APPLICANT Victor Larson et al.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT	FILING DATE Herewith	GROUP ART UNIT TBD

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
/KL/	6,016,512	1/2000	Christian Huitema	—		

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO	
/KL/	WO 98 55930	12/10/98	PCT	—			

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

	/KL/	Search Report (dated 10/7/02), International Application No. PCT/US01/13261
		F. Halsall, "Data Communications, Computer Networks And Open Systems", Chapter 4, Protocol Basics, 1996, pages 198-203
		Reiter, Michael K. and Rubin, Aviel D. (AT&T Labs – Research), "Crowds: Anonymity for Web Transmissions", pages 1-23
		Dolev, Shlomi and Ostrovsky, Rafil, "Efficient Anonymous Multicast and Reception" (Extended Abstract), 16 pages
		Rubin, Aviel D., Greer, Daniel, and Ranum, Marcus J. (Wiley Computer Publishing), "Web Security Sourcebook", pages 82-94
		FASBENDER, KESDOGAN, and KUBITZ: "Variable and Scalable Security" Protection of Location Information in Mobile IP", IEEE publication, 1996, pages 963-967

EXAMINER KRISNA Linn	DATE CONSIDERED 3/6/07

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.

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/714,849	11/18/2003	Victor Larson	007170.00025	3154

7590 01/08/2008	EXAMINER
BANNER & WITCOFF, LTD.	LIM, KRISNA
1100 13th STREET, N.W.	
SUITE 1200	
WASHINGTON, DC 20005-4051	

ART UNIT	PAPER NUMBER
2153	

MAIL DATE	DELIVERY MODE
01/08/2008	PAPER

NOTICE OF NON-COMPLIANT INFORMATION DISCLOSURE STATEMENT

An Information Disclosure Statement (IDS) filed 11/8/07 in the above-identified application fails to meet the requirements of 37 CFR 1.97(d) for the reason(s) specified below. Accordingly, the IDS will be placed in the file, but the information referred to therein has not been considered.

The IDS is not compliant with 37 CFR 1.97(d) because:

☑ The IDS lacks a statement as specified in 37 CFR 1.97(e).

☐ The IDS lacks the fee set forth in 37 CFR 1.17(p).

☐ The IDS was field after the issue fee was paid. Applicant may wish to consider filing a petition to withdraw the application from issue under 37 CFR 1.313(c) to have the IDS considered. See MPEP 1308.

Charles Bowen

PTO-1449 (Modified)	ATTY. DOCKET NO. 000479.00111	SERIAL NUMBER TBD 10/714,849
U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	APPLICANT Victor Larson et al.	
INFORMATION DISCLOSURE STATEMENT BY APPLICANT	FILING DATE Herewith	GROUP ART UNIT TBD 2153

U.S. PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	NAME	CLASS	SUB CLASS	FILING DATE
/LL	6,119,171	9/2000	Alkhatib			
/LL	5,588,060	12/24/96	Aziz			
/LL	5,689,566	11/18/97	Nguyen			
/LL	5,842,040	11/24/98	Hughes et al.			
/LL	4,933,846	06/12/90	Humphrey et al.			

FOREIGN PATENT DOCUMENTS

EXAMINER INITIAL	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUB CLASS	TRANSLATION YES/NO
/LL	199 24 575	12/2/99	DE			
/LL	0 838 930	4/29/98	EPO			
/LL	2 317 792	4/1/98	GB			
/LL	0 814 589	12/29/97	EPO			
/LL	WO 98/27783	6/25/98	PCT			

OTHER DOCUMENTS (Including Author, Title, Date, Pertinent Pages, Etc.)

/LL	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
	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 3/6/07

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.

JAN 2 9 2008
PTO/SB/30 (09-04)
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 contains a valid OMB control number.

Request for Continued Examination (RCE) Transmittal	Application Number	10/714,849
	Filing Date	November 18, 2003
	First Named Inventor	Victor Larson
Address to: Mail Stop RCE Commissioner for Patents P.O. Box 1450 Alexandria, VA 22313-1450	Art Unit	2153
	Examiner Name	Lim, Krisna
	Attorney Docket Number	077580-0042 (VRNK-1CP3CN)

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   Consider the IDS filed November 8, 2007

   b. ☒ Enclosed

      i. ☐ Amendment/Reply   iii. ☒ Information Disclosure Statement (IDS)

      ii. ☐ Affidavit(s)/Declaration(s)   iv. ☒ Other   Power of Attorney and Change of Address

2. **Miscellaneous**

   a. ☐ Suspension of action of the above-identified application is requested under 37 CFR 1.103(c) for a period of _____ months. (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)

      ii. ☐ One-Month 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)

01/30/2008 VBUI11   00000046 501133   10714849
01 FC:1801   810.00 DA

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	January 29, 2008
Name (Print/Type)	Atabak R. Royaee	Registration No.	59,037

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	January 29, 2008

This collection of information is required by 37 CFR 1.114. 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 1.11 and 1.14. This collection is estimated to take 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: Mail Stop RCE, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

RECEIVED
CENTRAL FAX CENTER

JAN 2 9 2008

Docket No.: 077580-0042   **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	Victor Larson et al.	
Appl. No.	:	10/714,849	
Filed	:	November 18, 2003	
Title	:	AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES	

Customer No.: 23,630
Confirmation No.: 3154

CERTIFICATE OF MAILING (37 CFR. § 1.10)

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as "Express Mail Post Office to Addressee" under 37 CFR 1.10 in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on January 29, 2008.

Express Mail Mailing Label: EV942455055US

Jacqueline Andrew

Grp./A.U.	:	2153
Examiner:	:	LIM, Krisna

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Mail Stop RCE
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 concurrently with an RCE for the above-referenced application.

BST99 1558649-1.077580.0042

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.

No fees are believed to be due with the filing of this paper. However, the Commissioner is hereby authorized to charge any required fees to Deposit Account 50-1133.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Atabak R. Royaee
Registration No. 26,418

28 State Street
Boston, MA 02109
Phone: 617-535-4108
Facsimile: 617-535-3800
Date: January 29, 2008

Please recognize our Customer No. 23630 as our correspondence address.

RECEIVED
CENTRAL FAX CENTER

JAN 2 9 2008

SHEET 1 OF 1

INFORMATION DISCLOSURE CITATION IN AN APPLICATION	ATTY. DOCKET NO. 077580-0042	SERIAL NO. 10/714,849
	APPLICANT Larson et al.	
(PTO-1449)	FILING DATE Nov. 18, 2003	GROUP 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
	A91	US 5,164,988 A	11/17/1992	Matyas	
	A92	US 5,790,548	8/4/1998	Sitaraman et al.	
	A93	US 6,101,182 B2	8/8/2000	Sitaraman et al.	
	A94	US 6,425,003 B1	7/23/2002	Herzog et al.	
	A95	US 6,606,708 B1	8/12/2003	Devine et al.	
	A96	US 6,751,738 B1	6/15/2004	Wesinger, Jr. et al.	
	A97	US 2003/0196122 A1	10/16/2003	Wesinger, Jr. et al.	
	A98	US 2006/0059337 A1	3/16/2006	Polyhonen et al.	

FOREIGN PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Foreign Patent Document Country Code3-Number 4-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

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.	

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.

BST99 1558649-1.077580.0042

PAGE 8/8 * RCVD AT 1/29/2008 4:18:32 PM [Eastern Standard Time] * SVR:USPTO-EFXRF-5/43 * DNIS:2738300 * CSID:617 535 3869 * DURATION (mm-ss):03-06

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.

RECEIVED
CENTRAL FAX CENTER

JAN 2 9 2008

FACSIMILE

Date: January 29, 2008

Time Sent:

To:	Company:	Facsimile No:	Telephone No:
Mail Stop: RCE	**USPTO**	**571-273-8300**	

From:	Atabak R. Royaee	*Direct Phone:*	617.535.4108
E-Mail:	aroyaee@mwe.com		
Sent By:	Jackie Andreu	*Direct Phone:*	617-535-4110
Client/Matter/Tkpr:	077580-0042 (VRNK-1CP3CN)	*Original to Follow by Mail*	No
		Number of Pages, Including Cover:	8

Message:

BST99 1559874-1.074280.0016

RECEIVED
CENTRAL FAX CENTER

JAN 2 9 2008

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Victor Larson
Application No.:	10/714,849
Filing Date:	November 18, 2003
Docket Number:	077580-0042 (VRNK-1CP3CN)
Title:	AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES
Examiner:	Lim, Krisna
Art Unit:	2153

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that this correspondence is being deposited with the U.S. Postal Service via Express Mail Label No. EV942455055US 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) to the USPTO, on the date indicated below.

Date: January 29, 2008

Jacqueline Andreu

MAIL STOP RCE
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER

Applicants transmit herewith the following document in the above-identified application:

1) Request for Continued Examination (RCE) (1 page);
2) Power of Attorney (1 page);
3) Statement Under 37 CFR 3.73(b) (1 page);
4) Supplemental Information Disclosure Statement (2 pages); and
5) Information Disclosure Statement by Applicant (Form 1449) (1 page).

The Commissioner is authorized to charge the RCE fee of $810.00, and/or any other fees that may due or credit any fees to our Deposit Account Number 50-1133.

Respectfully submitted,

Date: January 29, 2008

Atabak R. Royaee, Reg. No. 59,037
McDermott Will & Emery LLP
28 State Street
Boston, Massachusetts 02109-1775
Telephone: (617) 535-4108
Facsimile: (617) 535-3800

BST99 1563433-1.077580.0010

Doc Code: 077580 - 0042

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 | 23,830

OR

☐ Practitioner(s) named below (if more than 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 | 23,830

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) 535-4066	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	10/19/07
Name	Randall Carson	Telephone	831. 608.5698
Title	President		

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: **VirnetX Inc.**

Application No./Patent No.: **10/714,849**   Filed/Issue Date: **Nov. 18, 2003**

Entitled: **Agile Network Protocol For Secure Communications Using Secure Domain Names**

_____**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: _____**Victor Larson et al. (Inventors)**_____   To: _**Science Applications International Corporation**_
The document was recorded in the United States Patent and Trademark Office at
Reel ___**014711**___ , Frame ___**0008**___ , or for which a copy thereof is attached.

2. From: ____**Science Applications International Corp.**____   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.

_____   _____1/29/08_____
              Signature   Date

Toby H. Kusmer, PC — Reg. No. 26,418   **(617) 535-4065**
Printed or Typed Name   Telephone number

Attorney at McDermott Will & Emery LLP
Title.

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
10/714,849	11/18/2003	Victor Larson	007170.00025

CONFIRMATION NO. 3154

22907
BANNER & WITCOFF, LTD.
1100 13th STREET, N.W.
SUITE 1200
WASHINGTON, DC 20005-4051

POWER OF ATTORNEY NOTICE

OC000000028041082

Date Mailed: 01/31/2008

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 01/29/2008.

- 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).

/cbowen/

Office of Initial Patent Examination (571) 272-4000 or 1-800-PTO-9199

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
10/714,849	11/18/2003	Victor Larson	

CONFIRMATION NO. 3154

23630
MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

POA ACCEPTANCE LETTER

OC000000028041179

Date Mailed: 01/31/2008

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 01/29/2008.

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.

/cbowen/

Office of Initial Patent Examination (571) 272-4000 or 1-800-PTO-9199

UNITED STATES PATENT AND TRADEMARK OFFICE

Bib Data Sheet

CONFIRMATION NO. 3154

SERIAL NUMBER 10/714,849	FILING OR 371(c) DATE 11/18/2003 RULE	CLASS 709	GROUP ART UNIT 2153	ATTORNEY DOCKET NO.

APPLICANTS

Victor Larson, Fairfax, VA;
Robert Durham Short III, Leesburg, VA;
Edmund Colby Munger, Crownsville, MD;
Michael Williamson, South Riding, VA;

** CONTINUING DATA *************************

This application is a CON of 09/558,210 04/26/2000 ABN
which is a CIP 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
** 02/12/2004

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 VA	SHEETS DRAWING 40	TOTAL CLAIMS 23	INDEPENDENT CLAIMS 5

ADDRESS
23630

TITLE

AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

FILING FEE RECEIVED 2444	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

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Victor Larsen et al.
Serial No: 10/714,849
Filing Date: November 18, 2003
Group Art Unit: 2153
Examiner: Krisna Lim
Confirmation No.:   3154
Title: An Agile Network Protocol for Secure Communications Using Secure Domain Names
Docket No: 77580-042 (VRNK-1CP3CN)

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: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted (571-273-8300) to the USPTO, on the date indicated below.

Date: 12 March 2008

Jacqueline Andreu
Jacqueline Andreu

MS Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

AMENDMENT "C" AFTER FILING AN RCE

This paper is a response to the Notice of Allowance dated 29 October 2007 and is being filed as a part of a Request for Continued Prosecution filed on 29 January 2007 in order to insure that the Information Disclosure Statement filed on November 8, 2007 is considered by the Examiner, and to add additional dependent claims to claims previously allowed.

The Applicants request reconsideration and further examination in view of the following:

Amendments to the Claims, as reflected in the listing of claims beginning on page 2 of this paper; and

Remarks, beginning on page 14 of this paper.

Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

<u>Listing of Claims:</u>

1. (Canceled).

2. (Previously presented) The system of claim 29, wherein the top-level domain name is a non-standard top-level domain name.

3. (Original) The system of claim 2, wherein the non-standard top-level domain name is one of .scom, .sorg, .snet, .sgov, .sedu, .smil and .sint.

4. (Previously presented) The system of claim 28, wherein the communication network includes the Internet.

5. (Previously presented) The system of claim 28, wherein the domain name service system comprises an edge router.

6. (Previously presented) The system of claim 29, wherein the domain name service system is configured to authenticate the query using a cryptographic technique.

7. (Previously presented) The system of claim 28, wherein the domain name service system is connectable to a virtual private network through the communication network.

8. (Previously presented) The system of claim 7, wherein the virtual private network is one of a plurality of secure communication links in a hierarchy of secure communication links.

9. (Previously presented) The system of claim 7, wherein the virtual private network is based on inserting into each data packet communicated over a secure communication link one or more data values that vary according to a pseudo-random sequence.

10. (Previously presented) The system of claim 7, wherein the virtual private network is based on a network address hopping regime that is used to pseudorandomly change network addresses in packets transmitted between a first device and a second device.

11. (Previously presented) The system of claim 7, wherein the virtual private network is based on comparing a value in each data packet transmitted between a first device and a second device to a moving window of valid values.

12. (Previously presented) The system of claim 7, wherein the virtual private network is based on a comparison of a discriminator field in a header of each data packet to a table of valid discriminator fields maintained for a first device.

13. (Withdrawn) A method for registering a secure domain name, comprising steps of:
   receiving a request for registering a secure domain name;
   verifying ownership information for an equivalent non-secure domain name corresponding to the secure domain name;
   registering the secure domain name in a secure domain name service when the ownership information for the equivalent non-secure domain name is consistent with ownership information for the secure domain name.

14. (Withdrawn) The method according to claim 13, wherein the step of verifying ownership information includes steps of:
   determining whether the equivalent non-secure domain name corresponding to the secure domain name has been registered in a non-secure domain name service; and

querying whether the equivalent non-secure domain name should be registered in the nonsecure domain name service when the equivalent non-secure domain name has not been registered in the non-secure domain name service.

15. (Withdrawn) A computer-readable storage medium, comprising:

a storage area; and computer-readable instructions for a method for registering a secure domain name, the method comprising steps of:

receiving a request for registering a secure domain name;

verifying ownership information for an equivalent non-secure domain name corresponding to the secure domain name; and

registering the secure domain name in a secure domain name service when the ownership information for the equivalent non-secure domain name is consistent with ownership information for the secure domain name.

16. (Withdrawn) The computer-readable medium according to claim 15, wherein the step of verifying ownership information includes steps of:

determining whether the equivalent non-secure domain name corresponding to the secure domain name has been registered in a non-secure domain name service; and

querying whether the equivalent non-secure domain name should be registered in the non-secure domain name service when the equivalent non-secure domain name has not been registered in the non-secure domain name service.

17. (Withdrawn) A method for registering a domain name, comprising steps of:

(i) receiving a request for registering a first domain name;

(ii) verifying ownership information for a second domain name corresponding to the first domain name; and

(iii) registering the first domain name when the ownership information for the second domain name is consistent with ownership information for the first domain name.

18. (Withdrawn) The method of claim 17, wherein the first domain name comprises a nonstandard top-level domain and the second domain name comprises a standard top-level domain.

19. (Withdrawn) The method of claim 17, further comprising the step of storing information corresponding to the registration performed in step (iii) in a database separate from a database storing information for standard domain name registrations.

20. (Withdrawn) The method according to claim 17, wherein the step of verifying ownership information includes steps of:

(a) determining whether the second domain name has been registered in a domain name service; and

(b) querying whether the second domain name should be registered in the domain name service when the second domain name has not been registered in the domain name service.

21. (Withdrawn) A computer-readable medium, comprising computer-readable instructions for a method for registering a domain name, the method comprising steps of:

(i) receiving a request for registering a first domain name;

(ii) verifying ownership information for a second domain name corresponding to the first domain name; and

(iii) registering the first domain name when the ownership information for the second domain name is consistent with ownership information for the first domain name.

22. (Withdrawn) The computer readable medium of claim 21, wherein the first domain name comprises a non-standard top-level domain and the second domain name comprises a standard top level domain.

23. (Withdrawn) The computer-readable medium of claim 21, wherein the step of verifying ownership information includes steps of:

(a) determining whether the second domain name has been registered in a domain name service; and

(b) querying whether the second domain name should be registered in the domain name service when the second domain name has not been registered in the domain name service.

24. (Withdrawn) The method of claim 13, wherein the secure domain name has a top-level domain reserved for secure network connections.

25. (Withdrawn) The computer-readable storage medium of claim 15, wherein the secure domain name has a top-level domain reserved for secure network connections.

26. (Canceled).

27. (Canceled).

28. (Previously presented) A system for providing a domain name service for establishing a secure communication link, the system comprising:

a domain name service system configured to be connected to a communication network, to store a plurality of domain names and corresponding network addresses, to receive a query for a network address, and to comprise an indication that the domain name service system supports establishing a secure communication link.

29. (Previously presented) The system of claim 28, wherein at least one of the plurality of domain names comprises a top-level domain name.

30. (Previously presented) The system of claim 28, wherein the domain name service system is configured to respond to the query for the network address.

31. (Previously presented) The system of claim 28, wherein the domain name service system is configured to provide, in response to the query, the network address

corresponding to a domain name from the plurality of domain names and the corresponding network addresses.

32. (Previously presented) The system of claim 28, wherein the domain name service system is configured to receive the query initiated from a first location, the query requesting the network address associated with a domain name, wherein the domain name service system is configured to provide the network address associated with a second location, and wherein the domain name service system is configured to support establishing a secure communication link between the first location and the second location.

33. (Previously presented) The system of claim 28, wherein the domain name service system is connected to a communication network, stores a plurality of domain names and corresponding network addresses, and comprises an indication that the domain name service system supports establishing a secure communication link.

34. (Previously presented) The system of claim 28, wherein at least one of the plurality of domain names is reserved for secure communication links.

35. (Previously presented) The system of claim 28, wherein the domain name service system comprises a server.

36. (Previously presented) The system of claim 35, wherein the domain name service system further comprises a domain name database, and wherein the domain name database stores the plurality of domain names and the corresponding network addresses.

37. (Previously presented) The system of claim 28, wherein the domain name service system comprises a server, wherein the server comprises a domain name database, and wherein the domain name database stores the plurality of domain names and the corresponding network addresses.

38. (Previously presented) The system of claim 28, wherein the domain name service system is configured to store the corresponding network addresses for use in establishing secure communication links.

39. (Previously presented) The system of claim 28, wherein the domain name service system is configured to authenticate the query for the network address.

40. (Previously presented) The system of claim 28, wherein at least one of the plurality of domain names comprises an indication that the domain name service system supports establishing a secure communication link.

41. (Previously presented) The system of claim 28, wherein at least one of the plurality of domain names comprises a secure name.

42. (Previously presented) The system of claim 28, wherein at least one of the plurality of domain names enables establishment of a secure communication link.

43. (Previously presented) The system of claim 28, wherein the domain name service system is configured to enable establishment of a secure communication link between a first location and a second location transparently to a user at the first location.

44. (Previously presented) The system of claim 28, wherein the secure communication link uses encryption.

45. (Previously presented) The system of claim 28, wherein the secure communication link is capable of supporting a plurality of services.

46. (Previously presented) The system of claim 45, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof.

47. (Previously presented) The system of claim 46, 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.

48. (Previously presented) The system of claim 45, wherein the plurality of services comprises audio, video, or a combination thereof.

49. (Previously presented) The system of claim 28, wherein the domain name service system is configured to enable establishment of a secure communication link between a first location and a second location.

50. (Previously presented) The system of claim 49, wherein the query is initiated from the first location, wherein the second location comprises a computer, and wherein the network address is an address associated with the computer.

51. (Previously presented) The system of claim 28, wherein the domain name service system comprises a domain name database connected to a communication network and storing a plurality of domain names and corresponding network addresses for communication,

wherein the domain name database is configured so as to provide a network address corresponding to a domain name in response to a query in order to establish a secure communication link.

52. (Previously presented) A machine-readable medium comprising instructions executable in a domain name service system, the instructions comprising code for:

connecting the domain name service system to a communication network;

storing a plurality of domain names and corresponding network addresses;

receiving a query for a network address; and

supporting an indication that the domain name service system supports establishing a secure communication link.

53. (Previously presented) A method of providing a domain name service for establishing a secure communication link, the method comprising:

connecting a domain name service system to a communication network, the domain name service system comprising an indication that the domain name service system supports establishing a secure communication link;

storing a plurality of domain names and corresponding network addresses; and

receiving a query for a network address for communication.

54. (New) The machine-readable medium of claim 52, wherein the instructions comprise code for storing the plurality of domain names and corresponding network addresses including at least one top-level domain name.

55. (New) The machine-readable medium of claim 52, wherein the instructions comprise code for responding to the query for the network address.

56. (New) The machine-readable medium of claim 52, wherein the instructions comprise code for providing, in response to the query, the network address corresponding to a domain name from the plurality of domain names and the corresponding network addresses.

57. (New) The machine-readable medium of claim 52, wherein the instructions comprise code for receiving the query for a network address associated with a domain name and initiated from a first location, and providing a network address associated with a second location, and establishing a secure communication link between the first location and the second location.

58. (New) The machine-readable medium of claim 52, wherein the instructions comprise code for indicating that the domain name service system supports the establishment of a secure communication link.

59. (New) The machine-readable medium of claim 52, wherein the instructions comprise code for reserving at least one of the plurality of domain names for secure communication links.

60. (New) The machine-readable medium of claim 52, wherein the code resides on a server.

61. (New) The machine-readable medium of claim 52, wherein the instructions comprise code for storing a plurality of domain names and corresponding network addresses so as to define a domain name database.

62. (New) The machine-readable medium of claim 52, wherein the code resides on a server, and the instructions comprise code for creating a domain name database configured to store the plurality of domain names and the corresponding network addresses.

63. (New) The machine-readable medium of claim 52, wherein the instructions comprise code for storing the corresponding network addresses for use in establishing secure communication links.

64. (New) The machine-readable medium of claim 52, wherein the instructions comprise code for authenticating the query for the network address.

65. (New) The machine-readable medium of claim 52, wherein at least one of the plurality of domain names includes an indication that the domain name service system supports the establishment of a secure communication link.

66. (New) The machine-readable medium of claim 52, wherein at least one of the plurality of domain names includes a secure name.

67. (New) The machine-readable medium of claim 52, wherein at least one of the plurality of domain names is configured so as to enable establishment of a secure communication link.

68. (New) The machine-readable medium of claim 52, wherein the domain name service system is configured to enable establishment of a secure communication link between a first location and a second location transparently to a user at the first location.

69. (New) The machine-readable medium of claim 52, wherein the secure communication link uses encryption.

70. (New) The machine-readable medium of claim 52, wherein the secure communication link is capable of supporting a plurality of services.

71. (New) The machine-readable medium of claim 70, wherein the plurality of services comprises a plurality of communication protocols, a plurality of application programs, multiple sessions, or a combination thereof.

72. (New) The machine-readable medium of claim 71, 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.

73. (New) The machine-readable medium of claim 70, wherein the plurality of services comprises audio, video, or a combination thereof.

74. (New) The machine-readable medium of claim 52, wherein the domain name service system is configured to enable establishment of a secure communication link between a first location and a second location.

75. (New) The machine-readable medium of claim 74, wherein the instructions include code for receiving a query initiated from the first location, wherein the second

location comprises a computer, and wherein the network address is an address associated with the computer.

76. (New) The machine-readable medium of claim 52, wherein the domain name service system comprises a domain name database connected to a communication network and storing a plurality of domain names and corresponding network addresses for communication,

wherein the domain name database is configured so as to provide a network address corresponding to a domain name in response to the query in order to establish a secure communication link.

REMARKS

Claims 2-25 and 28-76 remain in the application. Claims 1, 26 and 27 have been previously canceled. Claims 54-76 have been added by this amendment. Claims 13-25, drawn to a non-elected invention, are withdrawn from consideration. Applicants note with appreciation that claims 2-25 and 28-53 have been previously allowed.

New claims 54-76 are all dependent either directly or indirectly from claim 52. Since claim 52 was previously allowed, the dependent claims are also believed to be allowable. 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.

Accordingly, all of the pending claims currently under consideration, claims 2-25 and 28-76, are believed to be in condition for allowance. An early and favorable action thereon is therefore earnestly solicited.

The Commissioner is hereby authorized to charge any necessary fees with regard to this filing to our Deposit Account No. 50-1133.

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: __March 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

INFORMATION DISCLOSURE CITATION IN AN APPLICATION	ATTY DOCKET NO 077580-0042	SERIAL NO 10/714,849
	APPLICANT Larson et al.	
(PTO-1449)	FILING DATE Nov. 18, 2003	GROUP 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
	A91	US 5,164,988 A	11/17/1992	Matyas	
	A92	US 5,790,548	8/4/1998	Sitaraman et al.	
	A93	US 6,101,182 B2	8/8/2000	Sitaraman et al.	
	A94	US 6,425,003 B1	7/23/2002	Herzog et al.	
	A95	US 6,606,708 B1	8/12/2003	Devine et al.	
	A96	US 6,751,738 B1	6/15/2004	Wesinger, Jr. et al.	
	A97	US 2003/0196122 A1	10/16/2003	Wesinger, Jr. et al.	
	A98	US 2006/0059337 A1	3/16/2006	Polyhonen et al.	

FOREIGN PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO	Foreign Patent Document Country Code3 -Number 4 -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

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	

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

Electronic Acknowledgement Receipt

EFS ID:	2985054
Application Number:	10714849
International Application Number:	
Confirmation Number:	3154
Title of Invention:	AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	23630
Filer:	Atabak R Royaee/Jacqueline Andreu
Filer Authorized By:	Atabak R Royaee
Attorney Docket Number:	
Receipt Date:	12-MAR-2008
Filing Date:	18-NOV-2003
Time Stamp:	11:36:43
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	Miscellaneous Incoming Letter	Transmittal.pdf	75134 1a38261885b0787948c8fa9ec79281ce53ac7819	no	1

Warnings:

Information:

2	Amendment After Final	Amendment.pdf	636661	no	14
			72370af620e5e4836b3231277cf27d4a6b1a49d2		
Warnings:					
Information:					
3	Information Disclosure Statement Letter	SupplementalStatement.pdf	79031	no	2
			2abb0598bf11c5153cd9a30020b5dc15d2d6fc61		
Warnings:					
Information:					
4	Information Disclosure Statement (IDS) Filed	IDS.pdf	82771	no	1
			e22a1d6fccdbfab7df4dab7a6894b08d22d00b8a		
Warnings:					
Information:					

This is not an USPTO supplied IDS fillable form

	Total Files Size (in bytes):	873597

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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	Victor Larson et al.	Customer No.: 23,630
Appl. No.	:	10/714,849	Confirmation No.: 3154
Filed	:	November 18, 2003	**CERTIFICATE OF MAILING (37 CFR. § 1.8(a))**
Title	:	An Agile Network Protocol for	

Title : An Agile Network Protocol for
        Secure Communications Using
        Secure Domain Names

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 USPTO, on **March 12, 2008.**

Grp./A.U. : 2153
Examiner: : LIM, Kristina

Jacqueline Andreu

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL LETTER

Enclosed for filing in connection with the above-referenced patent application are the following documents:

1) Amendment "C" After Filing An RCE
2) Supplemental Information Disclosure Statement
3) IDS Form PTO-1449 (1 page)

The Commissioner is hereby authorized to charge any fees that may be required for filing of the above-listed papers to our Deposit Account No. 50-1133.

Respectfully submitted,

McDERMOTT WILL & EMERY LLP

Date: March 12, 2008

Atabak R. Royaee, Reg. No.: 59,037
McDermott Will & Emery LLP
28 State Street
Boston, MA 02109-1775
Telephone: (617) 535-4108
Facsimile: (617) 535-3800

Docket No.: 077580-0042 (VRNK-1CP3CN)   **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant   :   Victor Larson et al.

Appl. No.   :   10/714,849

Filed   :   November 18, 2003

Title   :   AN AGILE NETWORK PROTOCOL
                 FOR SECURE COMMUNICATIONS
                 USING SECURE DOMAIN NAMES

Grp./A.U.   :   2153

Examiner: :   LIM, Krisna

SUPPLEMENTAL INFORMATION DISCLOSURE STATEMENT

Mail Stop RCE
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 form PTO-1449 filed January 29, 2008. 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 Statement is being filed as a substitution of the statement submitted on January 29, 2008, in which the undersigned agent's registration number was incorrect. A replacement form PTO-1449 for the one filed on Jan. 29, 2008 is also submitted herewith.

BST99 1566689-1 077580.0042

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.

No fees are believed to be due with the filing of this paper. However, the Commissioner is hereby authorized to charge any required fees to Deposit Account 50-1133.

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: March 12, 2008

Please recognize our Customer No. 23630 as our correspondence address.

Document code: WFEE

United States Patent and Trademark Office
Sales Receipt for Accounting Date: 03/27/2008

KWATSON   SALE #00000001   Mailroom Dt: 03/12/2008   501133   10714849
                 01   FC : 1202   1,150.00 DA

PATENT APPLICATION FEE DETERMINATION RECORD
Substitute for Form PTO-875

Application or Docket Number	Filing Date	
10/714,849	11/18/2003	☐ To be Mailed

APPLICATION AS FILED – PART I

OTHER THAN

	(Column 1)	(Column 2)	SMALL ENTITY ☐	OR	SMALL ENTITY

FOR	NUMBER FILED	NUMBER EXTRA	RATE ($)	FEE ($)		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))							
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	

APPLICATION AS AMENDED – PART II

OTHER THAN

	(Column 1)	(Column 2)	(Column 3)	SMALL ENTITY	OR	SMALL ENTITY

AMENDMENT

03/12/2008	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	RATE ($)	ADDITIONAL FEE ($)		RATE ($)	ADDITIONAL FEE ($)
Total (37 CFR 1.16(i))	* 73	Minus	** 50	= 23	X $ =		OR	X $50=	1150
Independent (37 CFR 1.16(h))	* 7	Minus	*** 7	= 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	1150

AMENDMENT

	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA	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:
/KIM WATSON SAUNDERS/

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	04/10/2008

MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

EXAMINER
LIM, KRISNA

ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 04/10/2008

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/714,849	11/18/2003	Victor Larson		3154

TITLE OF INVENTION: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

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	07/10/2008

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.

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   04/10/2008

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.
10/714,849	11/18/2003	Victor Larson		3154

TITLE OF INVENTION: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

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	07/10/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-226000

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. _____

This collection of information is required by 37 CFR 1.311. 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 take 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, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

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.

UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/714,849	11/18/2003	Victor Larson		3154

			EXAMINER
23630	7590	04/10/2008	LIM, KRISNA

MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

ART UNIT	PAPER NUMBER
2153	

DATE MAILED: 04/10/2008

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 663 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 663 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.

	Application No.	Applicant(s)
Notice of Allowability	10/714,849	LARSON 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 3/12/08*.

2. ☒ The allowed claim(s) is/are *2-12 and 28-76*.

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 claims 13-25.

Pursuant to 37 C.F.R 1.109 and M.P.E.P 1302.14, the following is an Examiner's Statement of Reasons for Allowance:

The prior arts of record do not teach or a domain name service system configured to be connected to a communication network, to store a plurality of domain names and corresponding network addresses, to receive a query for a network address, and to comprise an indication that the domain name service system supports establishing a secure communication link.

The examiner considers the applicants' claims 2-12 and 28-76 to be allowable based on the claim interpretation and the aforesaid prior arts of record.

Any comments considered necessary by applicant must be submitted no later than the payment of the Issue Fee and, to avoid processing delays, should preferably **accompany** the Issue Fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

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

KL

March 31, 2008

/Krisna Lim/

Primary Examiner, Art Unit 2153

	Application/Control No.	Applicant(s)/Patent Under Reexamination
Index of Claims	10714849	LARSON ET AL.
	Examiner	**Art Unit**
	Krisna Lim	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	03/31/2008									
	1	-									
3	2	=									
4	3	=									
6	4	=									
7	5	=									
5	6	=									
8	7	=									
9	8	=									
10	9	=									
11	10	=									
12	11	=									
13	12	=									
	13	-									
	14	-									
	15	-									
	16	-									
	17	-									
	18	-									
	19	-									
	20	-									
	21	-									
	22	-									
	23	-									
	24	-									
	25	-									
	26	-									
	27	-									
1	28	=									
2	29	=									
14	30	=									
15	31	=									
16	32	=									
17	33	=									
18	34	=									
19	35	=									
20	36	=									

		Application/Control No.	Applicant(s)/Patent Under Reexamination
Index of Claims		10714849	LARSON ET AL.
		Examiner	**Art Unit**
		Krisna Lim	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	03/31/2008									
21	37	=									
22	38	=									
23	39	=									
24	40	=									
25	41	=									
26	42	=									
27	43	=									
28	44	=									
29	45	=									
30	46	=									
31	47	=									
32	48	=									
33	49	=									
34	50	=									
35	51	=									
36	52	=									
60	53	=									
37	54	=									
38	55	=									
39	56	=									
40	57	=									
41	58	=									
42	59	=									
43	60	=									
44	61	=									
45	62	=									
46	63	=									
47	64	=									
48	65	=									
49	66	=									
50	67	=									
51	68	=									
52	69	=									
53	70	=									
54	71	=									
55	72	=									

U.S. Patent and Trademark Office   Part of Paper No. : 20080331

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10714849	LARSON ET AL.
	Examiner	**Art Unit**
	Krisna Lim	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	03/31/2008								
56	73	=								
57	74	=								
58	75	=								
59	76	=								

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10714849	LARSON ET AL.
	Examiner	**Art Unit**
	Krisna Lim	2153

ORIGINAL		INTERNATIONAL CLASSIFICATION												
CLASS	**SUBCLASS**	**CLAIMED**					**NON-CLAIMED**							
709	226	G	0	6	F	15 / 173 (2006.01.01)								
CROSS REFERENCE(S)														
CLASS	**SUBCLASS (ONE SUBCLASS PER BLOCK)**													

NONE			**Total Claims Allowed:**	
(Assistant Examiner) (Date)			60	
/Krisna Lim/	3/31/08		O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)		1	1

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	10714849	LARSON ET AL.
	Examiner	**Art Unit**
	Krisna Lim	2153

SEARCHED

Class	Subclass	Date	Examiner
709	226, 221	3/31/08	kl
713	201	3/31/08	kl

SEARCH NOTES

Search Notes	Date	Examiner
EAST, Inventor	3/31/08	kl

INTERFERENCE SEARCH

Class	Subclass	Date	Examiner
709	226	3/31/08	kl
713	201	3/31/08	kl

UNITED STATES PATENT AND TRADEMARK OFFICE

BIB DATA SHEET

CONFIRMATION NO. 3154

SERIAL NUMBER 10/714,849	FILING or 371(c) DATE 11/18/2003 RULE	CLASS 709	GROUP ART UNIT 2153	ATTORNEY DOCKET NO.

APPLICANTS

Victor Larson, Fairfax, VA;
Robert Durham Short III, Leesburg, VA;
Edmund Colby Munger, Crownsville, MD;
Michael Williamson, South Riding, VA;

** CONTINUING DATA **************************

This application is a CON of 09/558,210 04/26/2000 ABN
   which is a CIP 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 **
   02/12/2004

		STATE OR COUNTRY	SHEETS DRAWINGS	TOTAL CLAIMS	INDEPENDENT CLAIMS
Foreign Priority claimed ☐ Yes ☑ No 35 USC 119(a-d) conditions met ☐ Yes ☑ No Verified and /KRISNA LIM/ Acknowledged Examiner's Signature	☐ Met after Allowance Initials	VA	40	23	5

ADDRESS

MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775
UNITED STATES

TITLE

AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

FILING FEE RECEIVED 2444	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

BIB (Rev. 05/07).

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	190	(("5164988") or ("5790548") or ("6101182") or ("6425003") or ("6606708") or ("6751738") or ("4933846") or ("4988990") or ("5276735") or ("5311593") or ("5329521") or ("5341426") or ("5367643") or ("5559883") or ("5561669") or ("5588060") or ("5625626") or ("5654695") or ("5682480") or ("5689566") or ("5740375") or ("5774660") or ("5787172") or ("5796942") or ("5805801") or ("5842040") or ("5845091") or ("5867650") or ("5870610") or ("5878231") or ("5892903") or ("5898830") or ("5905859") or ("5918019") or ("5996016") or ("6006259") or ("6006272") or ("6016318") or ("6016512") or ("6041342") or ("6052788") or ("6055574") or ("6061736") or ("6079020") or ("6092200") or ("6119171") or	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2008/02/29 19:29

("6119234") or
("6147976") or
("6157957") or
("6158011") or
("6168409") or
("6175867") or
("6178409") or
("6178505") or
("6179102") or
("6222842") or
("6226751") or
("6233618") or
("6243360") or
("6243749") or
("6243754") or
("6256671") or
("6263445") or
("6286047") or
("6301223") or
("6308274") or
("6311207") or
("6324161") or
("6330562") or
("6332158") or
("6353614") or
("6430155") or
("6430610") or
("6487598") or
("6502135") or
("6505232") or
("6510154") or
("6549516") or
("6557037") or
("6571296") or
("6571338") or
("6581166") or
("6618761") or
("6671702") or
("6687551") or
("6714970") or
("6717949") or
("6760766") or
("6826616") or
("6839759") or
("7010604") or
("7133930") or
("7188180") or
("7197563")).PN.

2/ 29/ 08 7:38:54 PM
C:\ Program Files\ USPTO\ EAST\ Bin\ default.wsp

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	45	((VICTOR) near2 (LARSON)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2008/03/01 12:51
L2	150	((ROBERT) near2 (SHORT)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2008/03/01 12:51
L3	23	((EDMUND) near2 (MUNGER)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2008/03/01 12:51
L4	61	((MICHAEL) near2 (WILLIAMSON)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2008/03/01 12:52
L5	221	l1 or l2 or l3 or l4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 13:04
L6	25	l5 and (DNS or domain adj3 name adj4 service)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 13:24
L7	1146861	(secure domain name service).ti,ab,clm. or DSN.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 13:44
L8	1146861	l7 and 2ad<="19990607"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 13:49
L9	11	l8 and (scom.ti,clm. or sorg.ti,ab,clm.)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 13:50
L10	6931	709/226, "221".ccls. or 713/201.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 14:12

L11	138	l10 and (DNS or domain adj3 name adj4 service).ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 14:14
L12	32	l11 and @ad<="19990607"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 14:14
S1	190	((("5164988") or ("5790548") or ("6101182") or ("6425003") or ("6606708") or ("6751738") or ("4933846") or ("4988990") or ("5276735") or ("5311593") or ("5329521") or ("5341426") or ("5367643") or ("5559883") or ("5561669") or ("5588060") or ("5625626") or ("5654695") or ("5682480") or ("5689566") or ("5740375") or ("5774660") or ("5787172") or ("5796942") or ("5805801") or ("5842040") or ("5845091") or ("5867650") or ("5870610") or ("5878231") or ("5892903") or ("5898830") or ("5905859") or ("5918019") or ("5996016") or ("6006259") or ("6006272") or ("6016318") or ("6016512") or ("6041342") or ("6052788") or ("6055574") or ("6061736") or ("6079020") or ("6092200") or ("6119171") or ("6119234") or ("6147976") or ("6157957") or ("6158011") or ("6168409") or ("6175867") or ("6178409") or ("6178505") or ("6179102") or ("6222842") or ("6226751") or ("6233618") or ("6243360") or ("6243749") or ("6243754") or ("6256671") or ("6263445") or ("6286047") or ("6301223") or ("6308274") or ("6311207") or ("6324161") or ("6330562") or	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2008/02/29 19:29

| | | ("6332158") or ("6353614") or ("6430155") or ("6430610") or ("6487598") or ("6502135") or ("6505232") or ("6510154") or ("6549516") or ("6557037") or ("6571296") or ("6571338") or ("6581166") or ("6618761") or ("6671702") or ("6687551") or ("6714970") or ("6717949") or ("6760766") or ("6826616") or ("6839759") or ("7010604") or ("7133930") or ("7188180") or ("7197563")).PN. | | | | | |

3/ 1/ 08 2:14:50 PM
C:\ Program Files\ USPTO\ EAST\ Bin\ default.wsp

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	45	((VICTOR) near2 (LARSON)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2008/03/01 12:51
L2	150	((ROBERT) near2 (SHORT)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2008/03/01 12:51
L3	23	((EDMUND) near2 (MUNGER)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2008/03/01 12:51
L4	61	((MICHAEL) near2 (WILLIAMSON)).INV.	US-PGPUB; USPAT; USOCR	OR	ON	2008/03/01 12:52
L5	221	l1 or l2 or l3 or l4	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 13:04
L6	25	l5 and (DNS or domain adj3 name adj4 service)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 13:24
L7	1146861	(secure domain name service).ti,ab,clm. or DSN.ti,ab,clm.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 13:44
L8	1146861	l7 and 2ad<="19990607"	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 13:49
L9	11	l8 and (scom.ti,clm. or sorg.ti,ab,clm.)	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 13:50
L10	6931	709/226, "221".ccls. or 713/201.ccls.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2008/03/01 14:12

| L11 | 138 | l10 and (DNS or domain adj3 name adj4 service).ti, ab,clm. | US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB | OR | ON | 2008/03/01 14:14 |
| L12 | 32 | l11 and @ad<="19990607" | US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB | OR | ON | 2008/03/01 14:14 |

3/ 1/ 08 2:15:03 PM
C:\ Program Files\ USPTO\ EAST\ Bin\ default.wsp

SPW

STATEMENT UNDER 37 CFR 1.97(e) ACCOMPANYING INFORMATION DISCLOSURE STATEMENT	Docket No. 77580-042 (VRNK-1CP3CN)

In Re Application Of: Victor Larson, et al.

Application No. 10/714,849	Filing Date November 18, 2003	Examiner Krisna Lim	Customer No. 23630	Group Art Unit 2153	Confirmation No. 3154

Invention: AN AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS

JUL 16 2007

COMMISSIONER FOR PATENTS:

This is a statement under the provisions of 37 CFR 1.97(e) in the above-identified application.

Check applicable statement herebelow:

Statement Under 37 CFR 1.97(e)(1)

☒   Each item of information contained in the accompanying 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.

Statement Under 37 CFR 1.97(e)(2)

☐   No item of information contained in the accompanying Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the undersigned person, after making reasonable inquiry, no item of information contained in the accompanying 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.

_____
Signature

Dated: 7/12/07

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

cc:

Certificate of Mailing by First Class Mail

I 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 "Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)] on

7-12-07
_____
(Date)

_____
Signature of Person Mailing Correspondence

Cynthia Joseph
Typed or Printed Name of Person Mailing Correspondence

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH.   /K.L./

INFORMATION DISCLOSURE CITATION *(Use several sheets if necessary)*	Docket Number (Optional) 77580-042 (VRNK-1CP3CN)		Application Number 10/714,849
	Applicant(s) Victor Larson, et al.		
	Filing Date November 18, 2003		Group Art Unit 2153

JUL 16 2007

U.S. PATENT DOCUMENTS

*EXAMINER INITIAL	REF	DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE
/K.L./		6,557,037	04/29/2003	Provino	709	227	05/29/1998

U.S. PATENT APPLICATION PUBLICATIONS

*EXAMINER INITIAL	REF	DOCUMENT NUMBER	DATE	NAME	CLASS	SUBCLASS	FILING DATE IF APPROPRIATE

FOREIGN PATENT DOCUMENTS

	REF	DOCUMENT NUMBER	DATE	COUNTRY	CLASS	SUBCLASS	Translation YES	NO

OTHER DOCUMENTS *(Including Author, Title, Date, Pertinent Pages, Etc.)*

/K.L./		Eastlake, D. E., "Domain Name System Security Extensions", Internet Draft, April 1998 (1998-04), XP002199931, Sections 1, 2.3 and 2.4.

EXAMINER /Krisna Lim/	DATE CONSIDERED 02/21/2008

EXAMINER: Initial if citation considered, whether or not citation is in conformance with MPEP Section 609; Draw line through citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

Form PTO-A820
(also form PTO-1449)   ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /K.L./

P09A/REV05   Patent and Trademark Office * U.S. DEPARTMENT OF COMMERCE
SHEET 1 OF 1

/K.L./
10/714,849 /K.L./

/K.L./ SHEET 1 OF 4

02/21/2008

INFORMATION DISCLOSURE CITATION IN AN APPLICATION (PTO-1449)	ATTY. DOCKET NO. 077580-0042	SERIAL NO. 10/714,849
	APPLICANT Larson et al.	
	FILING DATE Nov. 18, 2003	GROUP 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
	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.

3

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /K.L./

INFORMATION DISCLOSURE CITATION IN AN APPLICATION

ATTY. DOCKET NO. **077580-0042**	SERIAL NO. **10/714,849**

APPLICANT **Larson et al.**	

(PTO-1449)

FILING DATE **Nov. 18, 2003**	GROUP **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
	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 /Krisna Lim/	DATE CONSIDERED 02/21/2008

*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.

4

INFORMATION DISCLOSURE CITATION IN AN APPLICATION	ATTY. DOCKET NO. 077580-0042	SERIAL NO. 10/714,849
(PTO-1449)	APPLICANT Larson et al.	
	FILING DATE Nov. 18, 2003	GROUP 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
	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 -Number 4 -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

EXAMINER /Krisna Lim/	DATE CONSIDERED 02/21/2008

*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.

5

INFORMATION DISCLOSURE CITATION IN AN APPLICATION	ATTY. DOCKET NO. 077580-0042	SERIAL NO. 10/714,849
	APPLICANT Larson et al.	
(PTO-1449)	FILING DATE Nov. 18, 2003	GROUP 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

FOREIGN PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Foreign Patent Document Country Code3 -Number 4 -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
	B1	EP 836306A1	4/15/1998	Sasyan et al.			
	B2	WO 00/17775	3/30/2000	Miller et al.			

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	RFC 2401 (dated Nov. 1998) Security Architecture for the Internet Protocol (RTP)	
	C2	RFC 2543-SIP (dated March 1999): Session Initiation Protocol (SIP or SIPS)	
	C3	Search Report, IPER (dataed Nov. 13, 2002), International Applicatoin No. PCT/US01/04340.	
	C4	Search Report, IPER (dated Feb. 06, 2002), International Application No. PCT/US01/13261.	
	C5	Search Report, IPER (dated Jan. 14, 2003), International Application No. PCT/US01/13260.	
	C6	Shankar, A.U. "A verified sliding window protocol with variable flow control". Proceedings of ACM SIGCOMM conferece on Communications architectures & protocols. pp. 84-91, ACM Press, NY,NY 1986.	
	C7	W. Stallings, "Cryptography And Network Security", 2nd, Edition, Chapter 13, IP Security, Jun. 8, 1998, pp. 399–440.	

EXAMINER /Krisna Lim/	DATE CONSIDERED 02/21/2008

*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.

BST99 1556883-1.077580.0042 ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /K.L./

RECEIVED
CENTRAL FAX CENTER

JAN 2 9 2008

SHEET 1 OF 1

INFORMATION DISCLOSURE CITATION IN AN APPLICATION	ATTY. DOCKET NO. 077580-0042	SERIAL NO. 10/714,849
	APPLICANT Larson et al.	
(PTO-1449)	FILING DATE Nov. 18, 2003	GROUP 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
	A91	US 5,164,988 A	11/17/1992	Matyas	
	A92	US 5,790,548	8/4/1998	Sitaraman et al.	
	A93	US 6,101,182 B2	8/8/2000	Sitaraman et al.	
	A94	US 6,425,003 B1	7/23/2002	Herzog et al.	
	A95	US 6,606,708 B1	8/12/2003	Devine et al.	
	A96	US 6,751,738 B1	6/15/2004	Wesinger, Jr. et al.	
	A97	US 2003/0196122 A1	10/16/2003	Wesinger, Jr. et al.	
	A98	US 2006/0059337 A1	3/16/2006	Polyhonen et al.	

FOREIGN PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO.	Foreign Patent Document Country Code3-Number 4-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

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.	

EXAMINER /Krisna Lim/	DATE CONSIDERED 02/21/2008

*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.

BST99 1558649-1.077580.0042

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /K.L./

INFORMATION DISCLOSURE CITATION IN AN APPLICATION	ATTY DOCKET NO 077580-0042	SERIAL NO 10/714,849
	APPLICANT Larson et al.	
(PTO-1449)	FILING DATE Nov. 18, 2003	GROUP 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
	A91	US	5,164,988 A	11/17/1992	Matyas	
	A92	US	5,790,548	8/4/1998	Sitaraman et al.	
	A93	US	6,101,182 B2	8/8/2000	Sitaraman et al.	
	A94	US	6,425,003 B1	7/23/2002	Herzog et al.	
	A95	US	6,606,708 B1	8/12/2003	Devine et al.	
	A96	US	6,751,738 B1	6/15/2004	Wesinger, Jr. et al.	
	A97	US	2003/0196122 A1	10/16/2003	Wesinger, Jr. et al.	
	A98	US	2006/0059337 A1	3/16/2006	Polyhonen et al.	

FOREIGN PATENT DOCUMENTS

EXAMINER'S INITIALS	CITE NO	Foreign Patent Document Country Code3 -Number 4 -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

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	

EXAMINER /Krisna Lim/	DATE CONSIDERED 03/31/2008

*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

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: <u>Mail</u> Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

or <u>Fax</u> (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   04/10/2008

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.

Jacqueline I. Andreu	(Depositor's name)
Jacqueline Andreu	(Signature)
July 9, 2008	(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/714,849	11/18/2003	Victor Larson		3154

TITLE OF INVENTION: AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES

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	07/10/2008

EXAMINER	ART UNIT	CLASS-SUBCLASS
LIM, KRISNA	2153	709-226000

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 **McDermott Will & Emery, LLP**

(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

VirnetX, Inc.

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Scotts Valley, 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 _____5_____

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 **50-1133** (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 _*Toby H. Kusmer*_	Date	**July 9, 2008**
Typed or printed name **Toby H. Kusmer**	Registration No.	**26,418**

This collection of information is required by 37 CFR 1.311. 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 take 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, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.

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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:)
)
Larson et al.   ) Group Art Unit: 2153
)
Serial No.: 10/714,849   ) Examiner: LIM, Krisna
)
Filed: November 18, 2003   )
) Confirmation No.: 3154
For: Agile Network Protocol For Secure   )
      Communications Using Secure   )
      Domain Names   )

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 filed online via EFS-Web to the USPTO, on the date indicated below.

Date: July 9, 2008

Jacqueline Andreu

Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AFTER A NOTICE OF ALLOWANCE
Under 37 C.F.R. §§ 1.312

Sir:

Pursuant to 37 C.F.R. § 1.312, this amendment is filed concurrently with the payment of the issue fee. Consideration of the following amendment remarks is respectfully requested.

Amendment to the Specification is reflected in this paper at page 2.

Remarks follow the amendment section of this paper at page 3.

AMENDMENTS TO THE SPECIFICAION:

In the specification, on page 1, after the first paragraph, before the title

"BACKGROUND OF THE INVENTION," 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

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. Entry of the amendment is respectfully requested.

Pursuant to MPEP § 2732 explaining 37 C.F.R. § 1.104(c)(10), this amendment (which is a letter regarding government interests) does not constitutes a failure of the applicants to engage in reasonable efforts to conclude processing or examination of the application and should not result in reduction of Patent Term Adjustment for the above-referenced application. See MPEP 2732 and Clarification of 37 CFR 1.704(c)(10) - Reduction of Patent Term Adjustment for Certain Types of Papers Filed After a Notice of Allowance has been Mailed, 1247 Off. Gaz. Pat. Office 111 (June 26, 2001).

No fee is believed to be due with the filing of this paper. However, the Commissioner is hereby authorized to charge our deposit account 50-1133 for any fee required for consideration and entry of this amendment.

To the extend necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please grant any extension of time required to enter this response and charge any additional required fees to our deposit account 50-1133.

Respectfully submitted,

McDERMOTT, WILL & EMERY, LLP

Dated: July 9, 2008   By: _____

Toby H. Kusmer, Reg. No. 26,418
Atabak R. Royaee, Reg. No. 59,037
McDERMOTT, WILL & EMERY, LLP
28 State Street
Boston, Massachusetts 02109-1775
Tel. (617) 535-4065
Fax: (617) 535-3800

(2)

Electronic Patent Application Fee Transmittal

Application Number:	10714849
Filing Date:	18-Nov-2003
Title of Invention:	AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES
First Named Inventor/Applicant Name:	Victor Larson
Filer:	Toby H. Kusmer./Jacqueline Andreu
Attorney Docket Number:	

Filed as Large Entity

Utility   Filing Fees

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
Utility Appl issue fee	1501	1	1440	1440
Publ. Fee- early, voluntary, or normal	1504	1	300	300
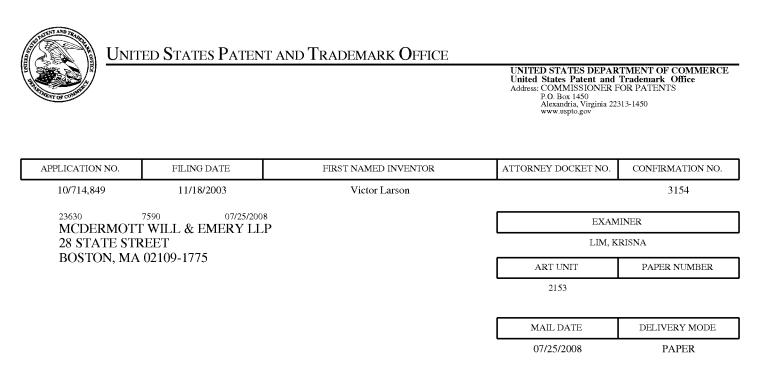
Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
Total in USD ($)				1740

Electronic Acknowledgement Receipt

EFS ID:	3586347
Application Number:	10714849
International Application Number:	
Confirmation Number:	3154
Title of Invention:	AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	23630
Filer:	Toby H. Kusmer./Jacqueline Andreu
Filer Authorized By:	Toby H. Kusmer.
Attorney Docket Number:	
Receipt Date:	09-JUL-2008
Filing Date:	18-NOV-2003
Time Stamp:	11:31:29
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	8004
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 1.21 (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)	IssueFeePayment.pdf	195729 / 8107622e7ebf0af94dcce4f94033a6e85389dfa3	no	1

Warnings:

Information:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
2	Amendment after Notice of Allowance (Rule 312)	Amendment.pdf	217784 / 33a6cccba216530e5fad977ba030ba7fc6940f99	no	3

Warnings:

Information:

Document Number	Document Description	File Name	File Size(Bytes) /Message Digest	Multi Part /.zip	Pages (if appl.)
3	Fee Worksheet (PTO-06)	fee-info.pdf	8316 / 84cad4f07943433d632e837a47f9b313c1741feb	no	2

Warnings:

Information:

	Total Files Size (in bytes):	421829

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 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.
10/714,849	11/18/2003	Victor Larson		3154

23630   7590   07/25/2008
MCDERMOTT WILL & EMERY LLP
28 STATE STREET
BOSTON, MA 02109-1775

EXAMINER
LIM, KRISNA

ART UNIT	PAPER NUMBER
2153	

MAIL DATE	DELIVERY MODE
07/25/2008	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.

	Application No.	Applicant(s)
Response to Rule 312 Communication	10/714,849	LARSON ET AL.
	Examiner	**Art Unit**
	Krisna Lim	2153

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address –

1. ☒ The amendment filed on <u>09 July 2008</u> under 37 CFR 1.312 has been considered, and has been:

  a) ☒   entered.

  b) ☒   entered as directed to matters of form not affecting the scope of the invention.

  c) ☐   disapproved because the amendment was filed after the payment of the issue fee.
      Any amendment filed after the date the issue fee is paid must be accompanied by a petition under 37 CFR 1.313(c)(1)
      and the required fee to withdraw the application from issue.

  d) ☐   disapproved. See explanation below.

  e) ☐   entered in part. See explanation below.

/Glenton B. Burgess/
Supervisory Patent Examiner, Art Unit 2153

/Krisna Lim/
Primary Examiner, Art Unit 2153

OK TO ENTER: /K.L./

07/22/2008

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	)
	)
Larson et al.	) Group Art Unit: 2153
	)
Serial No.: 10/714,849	) Examiner: LIM, Krisna
	)
Filed: November 18, 2003	)
	) Confirmation No.: 3154
For: Agile Network Protocol For Secure	)
Communications Using Secure	)
Domain Names	

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 filed online via EFS-Web to the USPTO, on the date indicated below.

Date: July 9, 2008

Jacqueline Andreu

Mail Stop Issue Fee
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

AMENDMENT AFTER A NOTICE OF ALLOWANCE
Under 37 C.F.R. §§ 1.312

Sir:

Pursuant to 37 C.F.R. § 1.312, this amendment is filed concurrently with the payment of the issue fee. Consideration of the following amendment remarks is respectfully requested.

Amendment to the Specification is reflected in this paper at page 2.

Remarks follow the amendment section of this paper at page 3.

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.
10/714,849	08/26/2008	7418504		3154

23630   7590   08/06/2008
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 646 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):

Victor Larson, Fairfax, VA;
Robert Durham Short III, Leesburg, VA;
Edmund Colby Munger, Crownsville, MD;
Michael Williamson, South Riding, VA;

IR103 (Rev. 11/05)

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. 6:10-cv-417	DATE FILED 8/11/2010	U.S. DISTRICT COURT Eastern District of Texas
PLAINTIFF VirnetX Inc.,		DEFENDANT 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

10714849

IN THE UNITED STATES
PATENT AND TRADEMARK OFFICE

In re patent of
VirnetX Inc.
Patent No. 7418504
Issued: August 26, 2008
For: Agile network protocol for secure communications using secure domain names

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	US5412717	US5400334	US5390247	US5201000	US5177788
US7861166	US7853723	US7849393	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	US7743248	US7734923	US7734789	US7730312	US7730311
US7730310	US7730299	US7730190	US7720076	US7716349	US7715371	US7707408	US7706332
US7702908	US7702540	US7699220	US7698567	US7694024	US7689826	US7680879	US7673072
US7669055	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	US7584358	US7584260	US7583971	US7583668	US7583665	US7580919
US7580715	US7575158	US7571850	US7558407	US7554995	US7546251	US7539942	US7526644

US7523072	US7522542	US7515712	US7509360	US7S09270	US7S02869	US7502406	US7496198
US7486660	U57484004	US7483417	US7477900	US7475156	US7475137	US7472156	US7469339
US7462746	US7461160	US74S4709	US7451193	US74438S8	US7424737	US7415617	US7404014
US7401286	US7392395	US7389270	US7386880	US7380273	US7366900	US7360244	US7359881
US7359137	US7342581	US7337233	US7336788	US7333485	US7325063	US7315893	US7307956
US7302473	US7299501	US7298851	US7296283	US7287271	US7283630	US7281133	US7277248
US7272625	US7269577	US7269576	US7269256	US7266529	US7265927	US7260556	US7260518
US7254518	US7251784	US7249378	US7249376	US7248719	US7248588	US7246148	US7236486
US7236320	US7225249	US7225160	US7224798	US7218625	US7210035	US7209902	US7200574
US7187686	US7181613	U57165174	US7162635	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	US7028187	US7028088	US7024392	US7023875
US7020700	US7020111	US7017046	US7010702	US7003587	US6983327	US6983009	US6978017
US6971008	US6968571	US6963923	US6957427	US69S6845	US6945457	US6944657	US6944600
US6940859	US6940840	US6938263	US6937729	US6937703	US6925644	US6925564	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	US67S4601	US6754212	US6754181	US6748082	US6745229	US6744892
US6742715	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	US6659354	US6658568	US6657956	US6654793
US6654697	US6650631	US6647418	US6640243	US6633S72	US6631402	US6618484	US6618353
US6611528	US6606378	US6603857	US6603763	US6601005	US6598081	US6594257	US6590894
US6587474	US6584094	US6578088	US6578079	US6577734	US6575372	US6571290	US6S64240
US6560340	US6560236	US6553002	US6546011	US6542897	US6542830	US6542569	US6539267
US6S39015	US6538988	US6S35517	US6532540	US6532392	US6529517	US6526508	US6519675
US6519615	US6519546	US6519224	US6516412	US6515996	US6515968	US6513116	US6S13061
US6512766	US6510519	US6510461	US6510151	US6505302	US6505301	US6504844	US6501767
US6501753	US6488211	US6487195	US6484244	US6480893	US6473803	US6473407	US6473406
US6470386	US6470340	US6467091	US6466780	US6463475	US6463443	US6463057	US6457039
US6456716	US6456608	US6456594	US6453350	US645334S	US6452925	US6449648	US6449574
US6449367	US6449344	US6446164	US6445710	US6445704	US6445703	US6442689	US6442170
US6438127	US6434504	US6427140	US6424992	US6424717	US6424627	US6418466	US6415335
US6414952	US6412074	US6411716	US6408367	US6408341	US6405272	US6405253	US6400371
US6397114	US6396831	US6393526	US6393486	US6393270	US6389480	US6389402	US6389004
US6385723	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	US6328217	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	US6263442	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	US6240187	US6240185
US6240091	US6240084	US6237786	US6237099	US6237096	US6237009	US6236660	US6233246
US6233234	US6233232	US6230245	US6230203	US6230194	US6226748	US6223209	US6222822
US6220510	US6219803	US6219726	US6219715	US6219707	US6219699	US6219421	US6212558

US6212506	US6212192	US6212183	US6212182	US6209098	US6209091	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
US6175626	US6173399	US6173322	US6172615	US6170057	US6170012	US6164549	US6163844
US6163843	US6163779	US6163543	US6161180	US6161145	US6161128	US6161126	US6161102
US6157721	US6157641	US6154744	US6151679	US61S1628	US6151325	US6148400	US6148083
US6147991	US6147987	US6145004	US6145001	US6144934	US6144667	US6141749	US6141351
US6138119	US6137869	US6134678	US6134658	US6134591	US6134235	US6131169	US6131116
US6130933	US6128666	US6128664	US61286S7	US6128642	US6128298	US6128296	US6122272
US6122258	US6119196	US6119109	US6118760	US6115780	US6115737	US6115458	US6115376
US6115040	US6112242	US6112239	US6112238	US6111883	US6108787	US6108786	US6108782
US6108700	US6108692	US61086S5	US6108300	US6105132	US6105131	US610S065	US6104716
US6104704	US6101616	US6101552	US6101189	US6098172	US6098123	US6098111	US6098108
US6097719	US6097718	US6097705	US6096094	US6094715	US6094655	US6094435	US6091951
US6091808	US6091725	US6088797	US6088728	US6088717	US6088451	US6088368	US6088356
US6084892	US6081807	US6081533	US6081522	US6081512	US6078590	US6075860	US6075858
US6075783	US6073241	US6073202	US6073185	US6073181	US6073178	US6073176	US6073122
US607310S	US6072942	US6072781	US6070243	US6070242	US6070191	US6070184	US6069890
US6067S73	US6067S61	US60650S9	US6065046	US6065002	US6064671	US6064667	US6063129
US6061797	US6061794	US6061734	US6061721	US6061362	US6061357	US6061346	US6061330
US6058478	US6058423	US6058412	US6058381	US60S8170	US6058106	US6055638	US605S562
US6055224	US6052819	US6052738	US6052718	US6052629	US60S2450	US6049872	US6049825
US6049602	US6049528	US6047376	US6047338	US6046988	US6044418	US6044402	US6044362
US6044224	US6044155	US6044144	US6044087	US6041379	US6041357	US6041356	US604135S
US6041325	US6041041	US6038664	US6038S96	US6035414	US603S360	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	US600267S	US5999972	US5999970
US5999965	US5999940	US5999629	US5999525	US5996077	US5996000	US5995726	US599S725
US599560S	US5995489	US5995021	USS991881	US5991876	US5991824	US5991814	US5991810
USS991809	US5991806	US5991746	US5991733	US5991406	US5991302	US5991292	US5991264
US5987523	US5987S21	US5987498	USS987497	US5983327	US5983281	USS983275	US5983270
US5983233	US5983176	USS982997	US5982864	US5980078	US5978880	US5978853	US5978813
US5978594	US5978568	US5978360	USS974S62	US5974460	US5974441	US5974396	US5970477
USS970467	US5970064	US5969632	US5969433	US5968177	US5968158	US5968116	US5966705
US5966695	US5966S28	US5966509	USS966431	US5966163	US5964841	US5963925	US5963746
US5963745	USS963556	US5961644	US5961620	US5961608	US5961602	US5960411	US5960179
US59S9990	US59S9972	US5958053	USS958052	US5958016	US5958012	US5956714	US5956409
US5956404	USS956403	US5954798	US59S3732	US5953507	US5953389	US5953005	US5951694
USS951652	US59516S1	US5951643	US5949976	US5949883	US5949882	US5949876	US5949874
US5949784	US5949761	USS949753	US5948069	US5946464	US5946295	US5944824	US5944782
US5943422	US5943230	US5941957	US5941947	US5941944	US5940591	US5940396	US5940394
US5938736	USS938732	US5938729	US5937165	US5937163	US5937162	USS933827	US5933606
US5933600	US5933596	US5933591	USS933503	US5933490	US5933428	US5933142	US5931900
USS930804	US5930786	US5930764	US5930257	US5928351	US5926463	US5926458	US5925109
US5923854	US55923853	US5923016	US5922073	US5922049	US5920861	US5920705	USS920695
USS920566	USS920542	US5919248	US5918021	US5918017	US5918016	US5917997	US5917912
US5917911	US55917821	US5917820	USS915101	US5915087	US5915019	US5915001	USS913041
US5913024	US5910987	US5910955	US5909682	US5909679	US5907717	USS907685	US5907681
USS907680	US5907620	US5907597	US5905714	US5903651	US5903572	US5901332	US5898780
US5896508	US5894479	US5892922	US5892919	US5892910	US5892909	US5892812	US5889958
US5889953	US55889776	US5884312	US5884284	US5884246	US5884038	US5884033	US5884032
US5884024	US5883956	US5883948	US5881243	US5881238	US5881237	US5878241	US5878212
US5878144	US55877759	US5875343	US5875329	US5875236	US5872931	US5872849	US5872847
US5872783	US5870631	US5870562	US5870558	US5870550	US5870545	US5870386	US5867667
US5867666	US5867665	US5867660	US5867647	US5867578	US5867495	US5867484	US5864854
US5864843	US55864654	US5862490	US5862344	USS862339	US5862325	USS859852	US5857022
US5856974	US55855020	US5854901	US5854900	US5854899	US5852812	US5852810	US5852724

US5852607	US5850S17	US5850451	US5850446	US5848399	US5848396	US5848258	US5848233
US584S267	US5845087	US5845080	US5845073	US5845070	US5845067	US5844896	US5844888
USS842043	US5842031	US5841865	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	US5825880
US5825871	US5825774	US5825772	US5825769	US5825283	US5822608	US5822531	US5822435
US5822433	US5822319	US5822303	US5819271	US5819225	US5819045	US5819033	US5819028
US5818930	US5818845	US5815723	US5815665	US5815664	US5815080	U5812819	US5812784
US5812775	US5812771	US5812750	US5812671	US5812666	US5812654	US5812552	US5812533
US5809292	US5809235	US5809128	US5808886	U55808671	US5805915	US5805822	US5805820
US5805818	US5805803	US55805785	US5805596	US5805595	US5805591	US5802590	US5802554
US5802367	US5802320	US5802316	US5802306	US5802304	US5802302	US5802291	US5802290
US5802285	US5802283	US55802053	US5802047	US5802043	US5799154	US5799090	US5799086
US5799016	US5799002	US5796952	US5796951	US5796944	US5796833	US5796824	US5796718
US5796393	US5794059	US5793983	US5793978	US5793965	US5793964	US5793763	US5793762
US5793759	US5793694	US5790809	US5790806	USS790805	US5790800	US5790797	US5790789
US5790780	US5787412	US5787160	U55787070	US5784566	US5784547	US5784464	USS784003
US5781743	US5781632	US5781551	US5781550	US5781534	US5778419	US5778377	US5778178
US55778174	US55777989	US5774689	US5774670	US5774668	US5774640	US5771459	US5771353
US5771291	US55768528	US5768501	US5768280	US5768271	US5765015	US5765012	US5764935
US5764915	US5764909	US5764789	U55764756	US5761669	US5761523	US5761306	US5758087
US5758083	US5757916	US55757900	US5754871	US5754830	US5754803	US5754774	US5754657
US5754656	US5754646	US55752260	US5752078	US5752067	US5751971	US5751954	US5751707
US5751706	U55751287	U55748901	US5748893	US5748871	US5748738	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	US55734709	US5734654	US5732406	US5732214	US5732213	US5732133	US5729689
US5729682	US55729452	US5727175	U55727155	US5727147	US5727129	US5727061	US5724510
US5724492	US5724355	US55721913	US5721908	US5721876	US5721780	US5720025	US5717944
US5717943	US5717686	US55715399	US5713037	US5712981	US5712914	U55712903	US5712897
US5710935	U55710884	U55710883	US5710882	US5710814	US5708836	US5708780	US5708659
US5708655	US5708654	US5706507	US5706502	US5706437	US5706427	US5706277	US5701465
US5699532	US5699528	US5699521	US5699513	US5699500	US5699403	US5696906	US5696898
US5696827	US5696825	US5696702	US5694546	US5694472	US5694335	US5692181	US5692126
US5692049	US55692030	US5689688	US5689645	US5689564	US5687235	US55685004	US5684950
US5684451	US55682553	US5682478	US55680470	US5680461	U55680456	US5680437	US55678045
US5678041	US55678006	US5675782	US5675732	US5675723	US5673322	US5671365	US5671354
US5671279	US55668992	US5668952	US5668878	US5668876	US5668857	U55666530	US5666486
US5666481	US55664199	US55664185	US5661803	US5661719	US5659601	US5659542	US5659350
US5657452	US55657450	US55657390	US5657320	US5657314	US5655077	US5654886	US5654746
US5654531	US55651068	US5651066	US5650994	US5649182	U55649099	U55644751	US5644733
US5644720	US5644706	US5644571	US5642515	US5640504	U55638515	US5636371	US5636216
US5636210	US55634099	US5634074	US5634015	US5633869	US5633371	US5632029	US5632011
US5630162	US5630081	US5630066	US5625836	US5625622	US5623605	US5623601	US5623600
US5623492	US55621889	US5621727	US5619716	US5619574	US5619498	US5617577	US5617547
US5617540	US5617421	U55615277	US5614891	US5613204	US5613096	US5612959	US5612958
US5612897	U55612865	US5612730	US5611075	US5610981	US5610915	U55610910	US5608908
US5608786	U55608738	US5608726	US5608446	US5606668	US5604807	US5604803	US5604729
US5604528	US55603029	US5602918	US5600820	US5600644	US5598536	US5594918	US5594798
US5594490	US55592669	US55592470	US5590299	US5590285	US5590199	US5588152	US5588059
US5587726	US55586263	U55586260	US5586257	US5586121	US55586112	US5583996	US5583933
US5583868	US55581559	US5577209	US5577196	US5574859	US55572643	US5572640	US5572533
US5570466	U55570361	US5570360	US5568613	US5568487	US5568471	US5566351	US5566170
US5564001	US55563805	U55561709	US5560013	US5559986	US5558339	US5557747	US5557742
US5555416	US55555304	US55555290	US5553287	US5553239	US5551052	US5551025	US5550816
US5550551	US55548756	US5548726	US5548724	US5548721	US5548649	US5548646	US5546389
US5544356	US55544340	US5544325	US5544322	US5542046	US5541919	US5539883	US5539826
US5539734	US55538255	US5537611	US5537592	US5537535	US5537417	US5537099	US5535403
US5535336	US55535334	US5535217	US5535206	US5534913	US5533108	US5533033	US5533029
US5530852	US55530758	US5530744	US5528763	US5526489	US5526257	US5524238	US5524227
US5522077	US5521925	US5521923	US5519858	US5519704	US5517618	US5517502	U55517488

US5515508	US5515441	US5515418	US5515361	US5513346	US5513337	US5509120	US5509006
US5508998	US5506973	US5506961	US5506956	US5506893	US5504921	US5504866	US5504814
US5502766	US5499297	US5497422	US5495580	US5495533	US5495426	US5493650	US5491800
US5491779	US5491752	US5491693	US5490252	US5490212	US5490208	US5490060	US5488715
US5488412	US5488411	US5485579	US5485576	US5485510	US5485465	US5483661	US5483654
US5483631	US5483596	US5481613	US5481604	US5481542	US5479395	US5478993	US5477531
US5476259	US5475836	US5475823	US5475758	US5475753	US5475687	US5473692	US5473599
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	US5420405	US5418922
US5418854	US5416842	US5414833	US5414694	US5410541	US5408506	US5406643	US5406628
US5406557	US5404562	US5404534	US5404505	US5402415	US5402394	US5400335	US5398248
US5396494	US5394408	US5392357	US5392280	US5390336	US5390326	US5390181	US5386548
US5386542	US5386413	US5384906	US5384777	US5381541	US5379289	US5379057	US5375219
US5375207	US5373559	US5371877	US5371852	US5371794	US5371494	US5369707	US5369688
US5369571	US5367517	US5361259	US5361256	US5359717	US5359660	US5355476	US5355472
US5355453	US5351243	US5349693	US5349686	US5347642	US5347450	US5347304	US5343477
US5341499	US5341496	US5341477	US5341293	US5339356	US5337360	US5337320	US5337309
US5333183	US5329619	US5327554	US5327486	US5327428	US5327426	US5325504	US5325423
US5325290	US5323146	US5321841	US5319642	US5315591	US5315093	US5313598	US5313465
US5309562	US5309437	US5307490	US5307347	US5305385	US5303303	US5303234	US5301337
US5301329	US5301247	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
US5200999	US5200949	US5199069	US5195092	US5195089	US5193151	US5193149	US5191611
US5187780	US5185860	US5185796	US5185795	US5182554	US5181107	US5179704	US5179591
US5178246	US5175416	US5173938	US5166978	US5166931	US5166930	US5166678	US5164986
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	US5086467	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	US5005200	US5005011	US5003597
US5003593	US5003591	US5001753	US5001752	US5001628	US4995082	US4995074	US4992646
US4991204	US4987571	US4983961	US4982430	US4982324	US4980913	US4980886	US4979100
US4975944	US4972504	US4968873	US4965804	US4965788	US4964120	US4962532	US4962531
US4962449	US4952930	US4949340	US4947390	US4942574	US4941176	US4935962	US4933970
US4933938	US4933937	US4932056	US4932021	US4930159	US4926479	US4926375	US4924515
US4924514	US4924303	US4922523	US4920484	US4918728	US4916737	US4916704	US4914571
US4912721	US4908861	US4907224	US4906828	US4901348	US4899333	US4893338	US4893248
US4891781	US4888801	US4885777	US4881264	US4878218	US4877950	US4873517	US4866421
US4864615	US4860201	US4859837	US4858224	US4853961	US4853950	US4852151	US4851993
US4850017	US4847837	US4846653	US4843026	US4825354	US4825050	US4823386	US4823338
US4817091	US4817050	US4811393	US4811337	US4811201	US4809265	US4807286	US4804248
US4803725	US4799156	US4799153	US4792973	US4787082	US4782326	US4779193	US4771461

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	U54641304	U54638356	U54635254	US4635194	US4634808	US4633462	US4633434
US4633036	US4630201	U54629872	U54627052	US4627045	US4621362	U54614861	US4613935
US4613901	US4601028	US4593282	US4587627	U54586175	US4578531	US4578530	US4577313
US4577224	US4575842	US54575579	US4574350	U54570261	U54567600	US4536791	US4533948
US4531929	US4531021	US4529870	U54528589	US4520488	US4491983	U54484025	US4479228
US4475123	US54470114	US4455605	U54454414	US4449181	US4445833	US4438493	US4424414
U54420068	U54417249	US4417245	U54413315	U54405829	US4403297	U54403282	US4400770
US4396984	US4393269	US54376299	U54371929	US4361851	US4345315	US4332027	US4326098
US4325116	US4309569	US4303904	US4282572	US4277837	US4264782	US4238854	U54227253
US4223380	US4218582	US4207609	US4207431	US4200770	US4186380	US4164787	US4160129
US4139737	US4063220	US4034347	US3985962	US3963102	US3956615	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
EP0828208	EP0825784	EP0825512	EP0821508	EP0818007	EP0813133	EP0809387	EP0752674
EP0740439	EP0737907	EP0732654	EP0731406	EP0729256	EP0729252	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
WO9639765	WO9630840	WO9618160	WO9613774	WO9613113	WO9605549	WO9604741	WO9602993
WO9600485	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	AU0698454	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

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. 6:11-cv-18	DATE FILED 1/12/2011	U.S. DISTRICT COURT Eastern District of Texas - Tyler Division
PLAINTIFF VirnetX, Inc.		DEFENDANT Mitel Networks Corp., et al.

PATENT OR TRADEMARK NO.	DATE OF PATENT OR TRADEMARK	HOLDER OF PATENT OR TRADEMARK
1 6,502,135	12/31/2002	VirnetX, Inc.
2 7,418,504	8/26/2008	VirnetX, Inc.
3		
4		
5		

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

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re *Inter Partes* Reexamination of:	)
	)
Victor LARSON et al.	) Control Nos.: 95/001,851; 95/001,788
	)
U. S. Patent No. 7,418,504	) Group Art Unit: 3992
	)
Issued: August 26, 2008	) Examiner: Roland G. Foster
	)
For: AGILE NETWORK PROTOCOL FOR	) Confirmation Nos. 1688; 5823
SECURE COMMUNICATIONS	)
USING SECURE DOMAIN NAMES	)

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. 7,418,504 ("the '504 patent"), hereby revokes all previous powers of attorney or authorization of agent granted in the '504 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 '504 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,**

-1-

L.L.P., Customer Number 22,852, to transact all business in the Patent and Trademark Office

connected with the '504 patent, including the reexamination proceedings assigned control nos.

95/001,851 and 95/001,788, and in any other proceedings involving the '504 patent.

Please also send all future correspondence concerning the '504 patent to the address

associated with **Finnegan, Henderson, Farabow, Garrett & Dunner, L.L.P., Customer**

Number 22,852.

Dated: _____11/30/12_____   By: _____

Sameer Mathur
Vice President, Corporate Development and Product
Marketing
VirnetX Inc.

-2-

Electronic Acknowledgement Receipt

EFS ID:	14368453
Application Number:	10714849
International Application Number:	
Confirmation Number:	3154
Title of Invention:	AGILE NETWORK PROTOCOL FOR SECURE COMMUNICATIONS USING SECURE DOMAIN NAMES
First Named Inventor/Applicant Name:	Victor Larson
Customer Number:	23630
Filer:	Joseph Edwin Palys./connie sisk
Filer Authorized By:	Joseph Edwin Palys.
Attorney Docket Number:	
Receipt Date:	03-DEC-2012
Filing Date:	18-NOV-2003
Time Stamp:	15:51:19
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	Patent_POA_504.pdf	55965 da46b2cbb41d37e3f69699fb630e0ade8f3 14809	no	2

Warnings:	
Information:	

Total Files Size (in bytes):	55965

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

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
10/714,849	11/18/2003	Victor Larson	11798.0007; 11798.0011

CONFIRMATION NO. 3154

22852
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER
LLP
901 NEW YORK AVENUE, NW
WASHINGTON, DC 20001-4413

POA ACCEPTANCE LETTER

OC000000058100471

Date Mailed: 12/14/2012

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 12/03/2012.

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.

/zabraha/

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
10/714,849	11/18/2003	Victor Larson	

23630
McDermott Will & Emery
The McDermott Building
500 North Capitol Street, N.W.
Washington, DC 20001

CONFIRMATION NO. 3154
POWER OF ATTORNEY NOTICE

OC000000058100437

Date Mailed: 12/14/2012

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 12/03/2012.

• 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).

/zabraha/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

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. 6:13-cv-00351	DATE FILED 4/22/2013	U.S. DISTRICT COURT Eastern District of Texas Tyler Division
PLAINTIFF VirnetX Inc. and Science Applications International Corporation		DEFENDANT 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 7,187,274		
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