

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re <i>Inter Partes</i> Reexamination of)	
U.S. Patent No. 7,490,151)	Control No.: 95/001,714/001,697
Edmund Colby Munger et al.)	Group Art Unit: 3992
Issued: February 10, 2009)	Examiner: Michael J. Yigdall
For: ESTABLISHMENT OF A SECURE)	Confirmation No.: 3428, 2161
COMMUNICATION LINK BASED)	
ON A DOMAIN NAME SERVICE)	
(DNS) REQUEST)	

COMMENTS BY THIRD PARTY REQUESTER PURSUANT TO 37 C.F.R. § 1.947

Mail Stop **Inter Partes Reexam**
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

On July 20, 2012, Patent Owner filed an overlength response (“Response”) to the April 20, 2012 Office action (“Office Action”) and a petition under 37 C.F.R. § 1.183 seeking waiver of the page limit for that response. On September 25, 2012, the Office granted Patent Owner’s petition, which set the date for a response by the Requestor for 30 days from the date of decision, which fell on Thursday, October 25, 2012. Third Party Requester believes that no fee is due in connection with the present response. However, any fee required for entry or consideration of this paper may be debited from Deposit Account No. 18-1260.

- A table of contents is provided at pages ii to iv. Requester submits the table of contents is not counted against the page limits applicable to this response. Should the Office determine otherwise, the Office is requested to disregard the table of contents.
- The response to the Patent Owner Comments begins on page 1.

TABLE OF CONTENTS

I.	Introduction	1
II.	Response to Patent Owner Contentions on Status of References as Prior Art.....	1
III.	The Rejections Of the Claims Were Proper And Should Be Maintained	3
	A. Response to Patent Owner’s Arguments Regarding the Rejection of Claims 1-16 Under 35 U.S.C. § 102(b) Based on <i>Aventail Connect v3.01</i> (Issue No. 1).....	4
	1. Independent Claim 1 (Issue No. 1).....	4
	a. <i>Aventail</i> Describes “Determining Whether the Intercepted DNS Request Corresponds to the Secure Server.”	4
	2. Independent Claims 7 and 13 (Issue No. 1).....	9
	3. Dependent Claims 2, 8, and 14 (Issue No. 1).....	9
	4. Dependent Claims 3, 9, and 15 (Issue No. 1).....	11
	5. Dependent Claims 4, 10, and 16 (Issue No. 1).....	12
	6. Dependent Claims 5 and 11 (Issue No. 1).....	12
	7. Dependent Claims 6 and 12 (Issue No. 1).....	13
	B. Response to Patent Owner’s Arguments Regarding the Rejection of Claims 1-16 Based on <i>Aventail AutoSOCKS Administrator’s Guide</i> (Issue No. 2).....	14
	C. Response to Patent Owner’s Arguments Regarding the Rejection of Claims 1-4, 6-8, 10, 12, 13 and 18 Based on <i>Beser in View of Kent</i> (Issue 4).....	14
	8. Independent Claim 1	14
	b. <i>Beser</i> and <i>Kent</i> Disclose a DNS Proxy Module that Intercepts DNS Requests Sent by a Client.....	16
	d. <i>Beser in View of Kent</i> , Renders Obvious Automatically Initiating an Encrypted Channel Between the Client and the Secure Server When the Request Corresponds to a Secure Server.....	19
	1. Independent Claim 7.....	21
	2. Independent Claim 13.....	21
	3. Dependent Claims 2, 8, and 14.....	22
	4. Dependent Claims 4, 10, and 16.....	23
	5. Dependent Claims 5 and 11.....	24
	6. Dependent Claims 6 and 12.....	25
	D. Response to Patent Owner’s Arguments Regarding the Rejection of Claims 1-16 Under 35 U.S.C. §102(a) Based on <i>BinGO</i> (Issue 3).....	25
	1. <i>BinGO</i> Expressly Incorporates <i>BinGO EFR</i>	25
	2. Independent Claim 1	26
	3. Independent Claims 7 and 13	33
	4. Dependent Claims 2, 8, and 14.....	34
	5. Dependent Claims 4, 10, and 16.....	35
	6. Dependent Claims 5 and 11.....	35
	7. Dependent Claims 6 and 12.....	36
	E. There are No Secondary Considerations Linked to the Claims	36
	F. Conclusions	37

I. Introduction

For reasons set forth in detail below, Requestor urges the Examiner to maintain the rejections of claims 1-16 set forth in the Office Action.

II. Response to Patent Owner Contentions on Status of References as Prior Art.

On pages 4-6 of the Response, Patent Owner asserts there is no evidence that the *Aventail*, *BinGO*, and *Kent* references are prior art under 35 U.S.C. § 102(a) or (b). The Patent Owner's claims border on the frivolous – each contested reference is unquestionably a printed publication, and only by studied ignorance can Patent Owner assert otherwise. Initially, Patent Owner misstates Requestor's burden to provide affirmative evidence with the Request proving the cited publications were publicly disseminated. In reality, all that is required is that Requester represent that the reference was published. In fact, 37 C.F.R. § 11.18 (the regulation patent owner cites) states precisely this – it provides that the submission of a paper by a party is a certification that “[t]o the best of the party's knowledge, information and belief, formed after an inquiry reasonable under the circumstances... [t]he allegations and other factual contentions have evidentiary support or, if specifically so identified, are likely to have evidentiary support after a reasonable opportunity for further investigation or discovery.” 37 CFR 11.18(b)(2)(iii). Thus, no authority supports Patent Owner's contention that Requestor was required to include affirmative evidence of dissemination of these printed publications.

Regardless, each of *Aventail*, *BinGO*, and *Kent* was publicly disseminated prior to February 15, 2000.¹ A reference is publicly accessible if it was “disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art exercising reasonable diligence can locate it.” *Kyocera Wireless Corp. v. Int'l Trade Comm'n*, 545 F.3d 1340, 1350 (2008) (internal quotations omitted).

The *Aventail* publications² were publicly distributed with deployments of *Aventail* products no later than August 9, 1999. Submitted with the Request were three separate declarations, each of which established that the *Aventail* publications were available no later than August 8, 1999. Patent Owner contends that there is no corroborative evidence of dissemination,

¹ Patent Owner did not contest Requester's assertions that the effective filing date of the '151 patent is no earlier than February 15, 2000, as set forth on page 9 of the Request.

² Patent Owner did not differentiate its challenges to the *Aventail* publications, but simply contests all three together. Requester accordingly responds in the same manner.

but that statement ignores the fact that the declarations corroborate each other. Indeed, there is a remarkable degree of consistency between the statements of Mssrs. Hopen, Fratto, and Chester, which conclusively establish the circumstances of the public distribution of the *Aventail* documents well before the effective filing date of the '151 patent.

Patent Owner next asserts that *BinGO* was not publicly distributed.³ Patent Owner is incorrect – *BinGO* was published and distributed publicly no later than March 30, 1999. The *BinGO* documents bear markings indicating they were published well before the filing date of the '135 patent. *Bingo UG*, for example, bears a March 1999 copyright date, while *Bingo EFR* was published one month earlier. Patent Owner contests these dates, asserting they are “merely evidence of creation, not of publication or dissemination” and that “Without more, this unsupported assertion of the alleged copyright date of the document as the publication date does not meet the ‘publication’ standard required for a document to be relied upon as prior art.” Response at 7-8. The “more” that Patent Owner seeks is readily available on the Internet. As documented by the Internet Archive (aka, “the Wayback Machine”), the company that published *BinGO*, in fact, distributed the *BinGO* documents on the Internet. See <http://web.archive.org/web/19990417093944/http://www.bintec.de/eftp/bingo.html>. Exhibit A provides an affidavit from the Office Manager of the Internet Archive, who testified that the numbers evidenced in the “Bingo” URL indicate that both *Bingo UG* and *BinGO EFR* were publicly available on the Internet no later than April 17, 1999. Furthermore, the archived webpage itself indicates that it was “last modified on Tuesday, March 30, 1999” – consistent with the copyright date on the *Bingo UG* publications. Section 2128 of the M.P.E.P states that “[a]n electronic publication, including an on-line database or Internet publication, is considered to be a ‘printed publication’ within the meaning of 35 U.S.C. 102(a) and (b) provided the publication was accessible to persons concerned with the art to which the document relates.” Thus, the evidence conclusively establishes that *BinGO* was publicly distributed no later than March 30, 1999.

Next, Patent Owner challenges the status of several Request for Comment (RFC) publications cited in the Request, claiming that “the record is devoid of evidence that any of

³ *BinGO* consists of the *BinGO User Guide* (“*Bingo UG*”) and the *BinGO Extended Feature Release* (“*BinGO EFR*”), which is expressly incorporated by reference in the *BinGO UG*.

these references are ... printed publications as of' each publication date listed on each RFC. This is a frivolous challenge. As anyone working in the field of network communications would know, RFC documents are published and disseminated to the relevant public by the Internet Engineering Task Force (IETF) pursuant to a transparent and well-known process. Under these well-known procedures, RFCs are self-authenticating printed publications – each contains verifiable information documenting the date of its public distribution. Specifically: (i) each number assigned to an RFC is unique and is not “re-used” if the subject matter in an RFC is revised or updated, (ii) the date each RFC is distributed to the public is listed the front page of the RFC, (iii) RFCs are distributed to the public over the Internet, via numerous protocols, (iv) each RFC is announced via an email distribution list on the date it is released to the public, and (v) RFCs are maintained in numerous archives publicly accessible via the Internet. *Id.* at ¶18-22. Indeed, Patent Owner cites several RFCs as publications in the '151 disclosure.⁴ Given this, it is remarkable that Patent Owner can even suggest that RFCs are not publicly disseminated. The evidence, thus, establishes that *Aventail*, *BinGO*, and *Kent* are each printed publications applicable as prior art to the '151 patent claims.

III. The Rejections Of the Claims Were Proper And Should Be Maintained

Claims are given “their broadest reasonable interpretation, consistent with the specification, in reexamination proceedings.” *In re Trans Texas Holding Corp.*, 498 F.3d 1290, 1298 (Fed. Cir. 2007). In determining that meaning “it is improper to ‘confine the claims to th[e] embodiments’ found in the specification.” *Id.* at 1299 (quoting *Phillips v. AWH Corp.*, 415 F.3d 1303, 1323 (Fed. Cir. 2005) (*en banc*)). While “the specification [should be used] to interpret the meaning of a claim,” the PTO cannot “import[] limitations from the specification into the claim.” *Id.* “A patentee may act as its own lexicographer and assign to a term a unique definition that is different from its ordinary and customary meaning; however, a patentee must *clearly* express that intent in the written description.” *Helmsderfer v. Bobrick Washroom Equip., Inc.*, 527 F.3d 1379, 1381 (Fed. Cir. 2008) (emphasis added). No such express definitions of key claim terms is provided in the '151 patent. Thus, these terms must be given their broadest reasonable interpretation in these reexamination proceedings.

⁴ See, e.g., '151 Patent at 3.

A. Response to Patent Owner's Arguments Regarding the Rejection of Claims 1-16 Under 35 U.S.C. § 102(b) Based on *Aventail Connect v3.01* (Issue No. 1)

1. Independent Claim 1 (Issue No. 1)

As explained in the Request, *Aventail v.3.01* ("*Aventail*") describes a system which intercepts DNS requests sent by a client, and if that request specifies a secure destination, automatically authenticates the client and establishes an encrypted channel between the client and a secure destination. *See, e.g.*, Request at 21-26. Consequently, the Office properly found that *Aventail* describes a system that anticipates claim 1. OA at 6-7. In response, Patent Owner asserts *Aventail* does not teach a system that: (1) "disclose[s] 'determining whether the intercepted DNS request corresponds to a secure server'; or (2) "disclose[s] 'when the intercepted DNS request corresponds to the secure server, automatically initiating an encrypted channel between the client and the secure server.'" Response at 7. Each assertion is incorrect.

a. *Aventail* Describes "Determining Whether the Intercepted DNS Request Corresponds to the Secure Server."

The Examiner correctly found that *Aventail* discloses a system that "determin[es] whether the intercepted DNS request corresponds to a secure server." In response, Patent Owner asserts that "whether or not a hostname is flagged by creating a false DNS entry does not indicate whether the alleged DNS request corresponds to a secure server, as false DNS entries may result even if a redirection rules is not matched." Response at 8. Patent Owner seems to believe that the *capacity* of the *Aventail* systems to be configured to not only handle secure and insecure destinations at the client, but in one implementation, to route all DNS requests for resolution at a remote server, somehow suggests *Aventail* does not automatically establish authenticated and secure connections when it determines that a DNS request specifies a secure destination. Patent Owner ignores two critical points. First, in the implementation Patent Owner does not discuss, *Aventail* plainly shows that the *Aventail Connect* client will, if it determines a request matches a redirection rule because it is specifies a secure destination, automatically establish a VPN between the client computer and the secure destination. Second, Patent Owner fails to point out where all DNS requests are proxied for resolution to a remote server, that server still will evaluate the DNS request, and if it specifies a secure destination, will establish a VPN between

the client computer and the secure destination. Patent Owner's focus on the mechanics of how the Aventail systems process DNS requests, thus, is a red herring.

Patent Owner next asserts that the Request "fail[s] to explain why matching a hostname to a redirection rule to 're-direct a request' is the same as determining whether a DNS request corresponds to a secure server." Response at 8. Yet, the Request explained that the specification of the '151 patent discloses that the claimed "determin[ation]" of whether a DNS request corresponds to a secure server may be "by reference to an internal table." Request at 22 (citing '151 patent at col.37, ll.60-66). As demonstrated above, the "determin[ation]" in *Aventail* occurs in virtually the same way – comparing the destination to entries in a lookup table. Moreover, Patent Owner's assertion presumes the claims restrict how this determination is to be made – but the plain language used in the claims imposes no such restrictions.

Patent Owner also contends that the Request does not show that any particular component "corresponds to a secure server." Response at 8. Patent Owner is incorrect – *Aventail* expressly teaches that when Aventail Connect "receives a connection request, it determines whether or not the connection needs to be redirected [to an Aventail ExtraNet Server and/or encrypted (in SSL)]." Request at 25 (citing *Aventail Connect v3.01* at 10). The Request also explains that the Aventail ExtraNet Server would "automatically establish an encrypted tunnel to the secure destination computer (i.e, a secure server), provided the client successfully authenticated with the Extranet Server." Request at 24. The Aventail Extranet Server is a "secure server" within the broadest reasonable construction of the claim 1.

b. *Aventail* Describes "When the Intercepted DNS Request Corresponds to the Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server."

The Examiner correctly found that *Aventail* discloses a system that "automatically initiat[es] an encrypted channel between the client and the secure server . . . when the intercepted DNS Request corresponds to the secure server." In response, Patent Owner contends that "proxying a connection into a private network based on a 'security policy' or server 'configuration'" does not "include[] automatically initiating an encrypted channel when an intercepted DNS request corresponds to the secure server." Response at 9. Patent Owner is again incorrect.

As explained in the Request, the *Aventail* system worked by automatically authenticating and encrypting communications between a client computer running Aventail Connect and a secure private network resource via the Aventail Extranet Server. Request at 25-26; Fratto ¶¶124-31. In particular, Aventail Connect worked with applications that communicate via TCP/IP—such as Web browsers—and was implemented using the existing WinSock functionality in client computers running Windows. Fratto ¶57. Thus, Aventail Connect necessarily acted on DNS requests containing, for example, either hostnames or IP addresses, Fratto ¶94 (“[Aventail Connect] executes a Domain Name System (DNS) lookup to convert the hostname into an Internet Protocol (IP) address.”), and evaluated such requests to determine if the request was seeking access to a destination that required authentication and encryption, such as a secure website, or access to a non-secure destination, such as a public website on the Internet. Fratto ¶94.

Patent Owner asserts that *Aventail* shows that the “alleged TCP handshake is results from the ‘routable IP address,’ not that it is related to the false DNS entry or the alleged DNS request....” Patent Owner is plainly incorrect. *Aventail* explains that the IP address of the Extranet Server is used as the destination for DNS requests specifying a secure destination – *Aventail* also explains that the fake DNS entry is simply used to enable Aventail Connect to function within OS-based TCP handling procedures. Similarly, *Aventail* shows that the “routable address” of a non-secure destination is provided through a conventional DNS lookup – which happens when the request is passed back to the TCP/IP handling procedures of the client operating system. Request at 25-26.

The Request also explained that “if an encryption module is enabled and selected by the SOCKS server, Aventail Connect encrypts the data on its way to the server ...” Request at 26 (citing *Aventail Connect v.3.01* at 12). In other words, if Aventail Connect determined that a DNS request contained a hostname specifying a secure destination, it would automatically and transparently handle authentication of the user to the private network and automatically encrypt/decrypt the communications between the client computer, the secure server, and the private network resource. Request at 25-26. Specifically, *Aventail* expressly shows that an encrypted channel is automatically established between a client computer running an Aventail client and a secure destination computer after it is determined that the connection request has specified a secure resource (i.e., the destination computer) on a private network. If it does, the

client computer running the Aventail client automatically performs the authentication of the client with the Aventail Extranet Server, which, if successful, results in the automatic establishment of an encrypted channel with the destination specified in the DNS request. The encrypted channel facilitates the transport of encrypted network traffic between the client and secure destination over the Internet, and the Aventail client automatically encrypts outgoing traffic and decrypts incoming traffic from the secure destination. Request at 25-26. By contrast, if the DNS request specifies a non-secure destination, the request is passed to the local operating system to handle DNS resolution and establishment of the connection. Request at 26. These are not, as Patent Owner asserts, “unconnected features and embodiments” of *Aventail* (Response at 9-10) – they are the sequence of events literally and plainly described in *Aventail*.

Indeed, Patent Owner’s remarkable contention that *Aventail* “does not teach any link between the alleged DNS request and the encryption, much less that encryption is automatically initiated when an ‘intercepted DNS request corresponds to a secure server’” is plainly refuted by the literal explanations in *Aventail*. See *Aventail Connect v3.01* at 1 (“Aventail Connect is a proxy client, but when used with SSL it provides the ability to encrypt inbound or outbound information.”); *Id.* at 7 (“Aventail Connect does not require administrators to manually establish an encrypted tunnel; Aventail Connect can establish an encrypted tunnel automatically.”); *Id.* at 42 (“Aventail can establish an encrypted tunnel automatically...”). Indeed, page 12 of *Aventail* explains that “step 3” of the process initiated when Aventail Connect determines that a secure destination is specified in the DNS request is to “transmit and receive data.” In that step, *Aventail* states that “[i]f an encryption module is enabled and selected by the SOCKS server, Aventail Connect encrypts the data on its way to the server on behalf of the application. If data is being returned, Aventail Connect decrypts it so that the application sees cleartext data.” *Id.*

Patent Owner next contends that the Request fails to show “that evaluating a connection request for the presence of a false DNS entry discloses determining that a DNS request corresponds to a secure server.” As noted above, the redirection rules used by Aventail Connect dictate if a destination specifies a secure destination; the false DNS entry is simply a flag used by Aventail Connect to handle a request determined to specify the secure destination.

Next, Patent Owner asserts that *Aventail* “does not disclose that the creation of a false DNS entry automatically initiates a connection, much less an encrypted channel.” Response at 10. Patent Owner again erroneously focuses on the *mechanism* used to implement the processes

described in *Aventail*. As explained in the Request, the Aventail Connect client would determine if a connection request was seeking access to a secure resource or not. If it was, and it contained a domain name, the Aventail Connect client would create a “false” DNS entry would be used to flag that connection request as requiring handling according to the policies enforced by the Aventail ExtraNet Server. Request at 22-25. These policies include, for example, evaluating the requests to determine if the request was seeking access to a destination that required authentication and encryption, such as a secure website, or access to a non-secure destination, such as a public website on the Internet. Request at 25. Obviously, the flag entered by Aventail Connect is simply information – *Aventail* shows that the Aventail Connect client, working with the ExtraNet Server, caused actions based on evaluation of that information.

Patent Owner also asserts that “the Request improperly mixes and matches the various separate embodiments of *Aventail v3.01* by pointing to the inbound access embodiment . . . and then turning to the outbound embodiment.” Response at 10-11. Patent Owner is incorrect, as it wrongly asserts that *Aventail* discloses two distinct embodiments related to outbound and inbound access. In *Aventail*, the characterization of “outbound” and “inbound” access is simply a function of perspective. Indeed, *Aventail* describes an end-to-end system that contemplates outbound requests from a client computer for access to a secure destination—from the perspective of the secure destination, that request and the encrypted channel that follows would, obviously, be described as an inbound connection. The communications are also plainly bi-directional. Moreover, the claims do not employ the terms “inbound” or “outbound” much less restrict the sequence of steps that comprise the claimed “data processing device.”

Patent Owner also criticizes the Request for relying on multiple sections of *Aventail* to demonstrate that the claims are anticipated. In particular, Patent Owner complains that it does not understand how “different embodiments and functionalities . . . separated by over sixty pages, can be combined to disclose” the above claim requirement. Response at 11. Patent Owner’s assertion is frivolous. The various sections and passages of *Aventail* cited in the Request simply provide varying degrees of detail in the description of the features and operation of the *Aventail* systems. The fact that those sections are, like any other technical publication, separated into different sections or found on different pages of the document is irrelevant. Consequently, the Examiner’s determination that claim 1 is anticipated by *Aventail* was proper and should be maintained.

2. Independent Claims 7 and 13 (Issue No. 1)

The Examiner correctly found that *Aventail* describes a system that anticipates claims 7 and 13. In response to the rejection of claim 7, Patent Owner asserts no response distinct from its response to the rejection of claim 1. Response at 11. Because the Examiner's rejection of claim 1 was proper, its rejection of claim 7 based *Aventail* also was proper and should be maintained.

In response to the rejection of claim 13, Patent Owner contends that the Request has "ignore[d]" the difference in claim language between claims 1 and 13. Patent Owner is incorrect. The only distinction identified by Patent Owner is that claim 13 recites "automatically creating a secure channel," while claim 1 recites "automatically initiating an encrypted channel." Response at 11. The Request plainly identified this distinction, explaining that "claim 13 is directed to subject matter similar to that recited in claim 1." Request at 42. Patent Owner identifies no issue of consequence tied to the different phrases. This is logical because there is none – the difference between "creating a secure channel" and "initiating an encrypted channel" is immaterial to the Examiner's determination that *Aventail* describes a system that anticipates claim 13. In fact, as the Examiner recognized, "[i]nitiating an encrypted channel" in claim 1 is simply a narrower limitation than claim 13's "creating a secure channel." See '504 ACP at 33 (explaining that a secure communication link does not require encryption). Because *Aventail* describes this element of claim 1 it necessarily describes a broader form of this element in claim 13. Consequently, the Examiner's rejection of claim 13 based on *Aventail* was proper and should be maintained.

3. Dependent Claims 2, 8, and 14 (Issue No. 1)

The Examiner correctly found that *Aventail* describes a system that anticipates claims 2, 8 and 14. In response to the rejection of the claims, Patent Owner contends that *Aventail* does not disclose the element of "when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client." Response at 12. Patent Owner misunderstands the Request and teachings of *Aventail*. As explained in the Request, a client computer running *Aventail* Connect would have to successfully authenticate before being given access to a secure destination. Request at 27-28. In particular, *Aventail* explains that:

Depending on the security policy and the Aventail ExtraNet Server configuration, Aventail Connect will automatically proxy their allowed application traffic into the private network. In this situation, Aventail Connect will forward traffic destined for the private internal network to the Aventail ExtraNet Server. Then, based on the security policy, the Aventail ExtraNet Server will proxy user traffic into the private network but only those resources allowed.” (emphasis added)

Aventail Connect v.3.01 at 72-73. Patent Owner does not address this passage—which was expressly noted by the Examiner—because it plainly shows the embodiment referenced in these claims.

Patent Owner elects instead to present a convoluted and confused discussion of different aspects of the *Aventail* process. In particular, Patent Owner conflates the various distinct processes that occur when Aventail Connect acts on a request. For example, Patent Owner intermingles the steps taken when a DNS request contains an IP address with those where the DNS request contains a host name. Similarly, Patent Owner confuses the steps taken by Aventail Connect when the client determines a requested destination is secure versus when it is insecure. Patent Owner then presents a mangled conclusion from its incorrect reading of *Aventail*, asserting first that a “routable address” only is obtained when the Aventail Connect client does not create a false DNS entry flag, and then that a “proxy connection” occurs “only after authentication and encryption have already been established.”

Instead of attempting to unravel this hopelessly confused and inaccurate description of *Aventail*, the Examiner need only read *Aventail*, which clearly explains that if the Aventail Connect client determines a request is specifying a secure destination (i.e., either because it contains a secure hostname or because it contains a secure IP address), it sends a message to the ExtraNet Server to commence the authentication process. If that authentication is successful, an encrypted channel is established. *See Aventail Connect v3.01* at 11-12; *see also* Request at 26 (explaining that *Aventail* shows that “[u]ser authentication and encryption on the Aventail ExtraNet Server require all users to use Aventail Connect to authenticate and encrypt their sessions before any connection to the internal private network(s).”) Request at 26. Thus, Aventail plainly anticipates claims 2, 8 and 14, and Patent Owner’s assertion that the Request “pick[s] and choose[es] disparate features from various embodiments” to support the rejection of claims 2, 8, and 14 is simply false.

4. Dependent Claims 3, 9, and 15 (Issue No. 1)

The Examiner correctly found that *Aventail* discloses every limitation of dependent claims 3, 9 and 15. Patent Owner disagrees, wrongly asserting that the Request “does not describe how error values corresponding to a ‘connection not allowed by ruleset’ or a ‘connection refused’” amounts to the ‘host unknown error message’” recited by the claims. Response at 13. Once again, Patent Owner only addresses one of the examples set forth in the Request, wholly ignoring the disclosure that was adopted by the Examiner. As explained in the Request (and in the Declaration of Michael E. Fratto), “an unsuccessful authentication attempt by a client computer running *Aventail* Connect v3.01” will result in the return of a DNS error by the server, a feature that is inherent in “the SOCKS v5 protocol used by” *Aventail*. Request at 28. Moreover, as Mr. Fratto explained, “all of the *Aventail* VPN solutions are implemented in TCP/IP communications. As such, these solutions would inherently know how to handle errors returned according to the relevant DNS and TCP/IP communication protocols.” Fratto at ¶139. Mr. Fratto also explained that if a DNS request is unsuccessful, the address record returned in the response will not contain the resolved IP address, but instead will contain an RCODE. Fratto at ¶140. A common RCODE that would be returned when a DNS request is unsuccessful is RCODE 3, which specifies the error “host not found.” Request at 28-29. So, as explained in the Request, when a DNS request is unsuccessful, the address record returned in the response will not contain the resolved IP address, but instead will inherently contain an RCODE. Request at 28-29; Fratto ¶136-140.

Patent Owner also contends that “it is also not appropriate to rely solely on interpretation or ‘common knowledge’ in the art without evidentiary support in the record as the principal evidence upon which a rejection is based.” Patent Owner further states that “the way a particular component may handle a failure to authenticate is a subject matter of a highly technical field that requires a significant skill in the art.” Patent Owner’s contentions are moot – the Request included the expert opinion of Michael Fratto who unquestionably has “significant skill” in the precise field of the claims. The Examiner’s adoption of Mr. Fratto’s opinions, which were based on relevant technical authorities, was entirely appropriate. Consequently, the Examiner’s rejection of claims 3, 9 and 15 as anticipated by *Aventail* was also proper and should be maintained.

5. Dependent Claims 4, 10, and 16 (Issue No. 1)

The Examiner correctly found that *Aventail* describes a system that anticipates claims 4, 10 and 16. In response to the rejection of these claims, Patent Owner asserts no response distinct from its response to the rejection of claims 1-3, 7-9, and 13-15. Response at 14. Because those rejections were proper, the Examiner's rejection of claims 4, 10 and 16 was proper and should be maintained.

6. Dependent Claims 5 and 11 (Issue No. 1)

The Examiner correctly found that *Aventail* discloses every limitation of dependent claims 5 and 11. In response, Patent Owner asserts that the Request does not show "establishing an [IP] address hopping scheme between the client and the secure server" because the disclosed proxy schemes in *Aventail* "are implemented merely to satisfy the 'need to traverse multiple firewalls.'" Response at 15. The Patent Owner therefore concludes that "providing a mechanism for traversing multiple firewalls does not contribute in any meaningful way towards securing data transmitted over a public network, much less establishing a VPN." Response at 15.

Patent Owner's response is unpersuasive for two different reasons. First, the claims do not impose restrictions on the nature of IP address hopping schemes as the Patent Owner contends (i.e., that they must contribute in a "meaningful way towards securing data"). Second, the IP hopping schemes described in *Aventail* are not simply methods for "traversing multiple firewalls" as Patent Owner argues – they are schemes for routing IP packets between a client and server. Because this is all that the claims require in their broadest reasonable construction, these schemes in *Aventail* meet this requirement of the claims.

Patent Owner also repeats its frivolous argument that the Request "mix[es] and match[es] various unrelated features" from the *Aventail* disclosure. Here Patent Owner asserts that the disclosure of the "MultiProxy scheme or the Proxy Chaining scheme" is "some forty or so pages" from other disclosures relied upon by the Request. As explained above, the IP address hopping schemes disclosed in *Aventail* are features of the *Aventail* system, not "another embodiment" as Patent Owner contends. That descriptions of these features are found on different pages of the *Aventail* publication is entirely irrelevant. Further, the fact that these features may be optionally implemented within the *Aventail* system does not mean *Aventail* does not disclose a system that comprises those features. Finally, Patent Owner's reliance on

NetMoneyIN, Inc. v. Verisign, Inc., 545 F.3d 1359 (Fed. Cir. 2008) is misplaced. Unlike the situation presented in *NetMoneyIN*, no third party interpretation of the teachings of *Aventail* is necessary – *Aventail* expressly discloses the features required by the claims via its description of the MultiProxy or Proxy Chaining schemes, which meet the requirement of “establishing an [IP] address hopping scheme between the client and the secure server.” Request at 30-31, 40-41. Accordingly, the Examiner’s rejection of these claims was proper and should be maintained.

7. Dependent Claims 6 and 12 (Issue No. 1)

The Examiner correctly found that *Aventail* discloses every limitation of dependent claims 6 and 12. In response, Patent Owner presents an obviously incorrect portrayal of the claim requirements, the *Aventail* procedures, and the Request. Specifically, at page 16, Patent Owner incorrectly asserts that “[a]s an initial matter, [] the Office Action and the Request are alleging that a SOCKS server or an Aventail Extranet is a ‘secure server’ with respect to claims 1 and 7.” First, the claims do not delineate what the claimed “secure server” may comprise. They also do not restrict which of several servers in a path of communications may be the “secure server.” Thus, the secure destination computers and/or Aventail Extranet Server of *Aventail* may be the claimed “secure server” of the claims.

More directly, *Aventail* plainly shows that “the true IP address of the secure destination computer would not be sent to the client computer.” Request at 31. *Aventail* also shows that the communications between the client computer, the Aventail Extranet Server, and the secure destination computer are ordinarily encrypted. Request at 31 (citing *Aventail Connect v3.01* at 72-73) (“Depending on the security policy and the Aventail ExtraNet Server configuration, *Aventail Connect* will automatically proxy their allowed application traffic into the private network. In this situation, *Aventail Connect* will forward traffic destined for the private internal network to the Aventail ExtraNet Server. Then, based on the security policy, the Aventail ExtraNet Server will proxy mobile user traffic into the private network but only to those resources allowed.”) *Aventail* thus shows that the encrypted traffic would not be sent directly between the client and the secure destination computers, but instead would be routed through the Aventail Extranet Server. Consequently, the true IP address of the secure destination computer would not be sent to the client computer when that client computer was communicating through an encrypted channel to the secure destination; rather, the client computer running *Aventail Connect* would send its traffic destined for the secure destination computer to the Aventail

Extranet Server, which would then route that traffic to the secure destination computer.

Accordingly, the Examiner's rejection of this claim was proper and should be maintained.

B. Response to Patent Owner's Arguments Regarding the Rejection of Claims 1-16 Based on *Aventail AutoSOCKS Administrator's Guide* (Issue No. 2)

The Patent Owner does not contest any of the evidence or explanations in the Request that are specific to *AutoSOCKS/Administrator's Guide*, but instead incorporates and relies on its positions regarding *Aventail v3.01/Administrator's Guide*. Because the rejections of claims 1 to 16 based on *Aventail v3.01* were proper, the Examiner's rejection of these claims as anticipated by *AutoSOCKS/Administrator's Guide* also was proper and should be maintained.

C. Response to Patent Owner's Arguments Regarding the Rejection of Claims 1-4, 6-8, 10, 12, 13 and 18 Based on *Beser in View of Kent* (Issue 4).

As the Request explained, *Beser* describes systems and processes in which an IP tunnel is securely and transparently established between two network devices with the aid of a third-party trusted network device on a public network. A description of *Beser* is provided at pages 118 to 122 of the Request. Patent Owner agrees that *Beser* discloses a system for initiating a tunneling connection. Patent Owner asserts, however, that both *Beser* and the '151 claims should be read in an unrealistically narrow manner, which is contrary to how a person of ordinary skill would read them. Patent Owner's reasons are unpersuasive, and should be rejected.

8. Independent Claim 1

The Examiner correctly found that *Beser*, in view of *Kent*, would have rendered obvious claim 1. Patent Owner disagrees, arguing that (1) *Beser* would not be read in conjunction with *Kent*; (2) *Beser* and *Kent* do not suggest a domain name server proxy module that intercepts DNS requests sent by a client; (3) *Beser* and *Kent* do not suggest determining whether an intercepted DNS request corresponds to a secure server; (4) *Beser* in view of *Kent* does not make obvious forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer when the intercepted request does not correspond to a secure server; and (5) *Beser* in view of *Kent* does not make obvious automatically initiating an encrypted channel between the client and the secure server when the request corresponds to a secure server. Response at 36-43. None of these allegations is correct based on what is actually taught by *Beser* and *Kent*, and what the claims read in their broadest reasonable construction actually require.

a. A Person of Ordinary Skill in the Art Would Be Motivated to Combine the Teachings of *Beser* with That of *Kent*

Patent Owner's principle challenge to *Beser* is its belief that *Beser* "teaches away" from the use of encryption in IP tunneling applications, claiming that *Beser* "explains that encryption is 'infeasible' and/or 'inappropriate' in VoIP applications." Response at 36. According to Patent Owner, "*Beser*'s disclosed system and method for initiating a tunneling association is intended as an alternative to encryption to address the drawbacks that arise from the teachings of *Kent* (e.g., high computing power), not to encourage use of encryption." Request at 37. This leads Patent Owner to assert that *Beser* would not have been combined with *Kent*, which describes the IPsec protocol and how to implement it in a variety of network designs. Patent Owner's characterization of *Beser* and *Kent* is grossly inaccurate and ultimately irrelevant.

First, *Beser* never suggests that use of encryption in IP tunneling schemes is "undesirable." Instead, *Beser* consistently and repeatedly points out using encryption in IP tunneling schemes (of which its system is one) is conventional and ordinarily should be used. *Beser* at col.1, ll.54-56 ("Of course, the sender may encrypt the information inside the IP packets before transmission, e.g. with IP Security ('IPsec')."") *Beser* also explains that a decision to not use encryption will be driven by practical considerations, such as (i) the volume of data being transmitted (i.e., certain high data volume VOIP and multimedia settings), (ii) the capacity of a particular hardware setup to handle the volume of data and (iii) cost considerations. Importantly, *Beser* indicates that these practical concerns do not always arise for these two applications – nothing in *Beser* suggests they are even relevant to other situations. *Beser* read accurately actually makes it clear that, other than in rare situations, encryption should be used in IP tunneling applications. *Beser* at col.1, ll.54-66. And, critically, since none of the '151 claims limit use of the claimed systems to settings requiring "high volume" data transfers, the cautions in *Beser* relating to "high data volume" applications of IP tunneling systems are entirely irrelevant to the '151 claims. Thus, *Beser* does not "teach away" from using encryption in IP tunneling systems, and it certainly does not do so for systems not required to transmit high volumes of data (i.e., those other than VOIP and multimedia).

Second, the Patent Owner simply ignores the disclosure in *Beser* that indicates that encryption is, in fact, used in the *Beser* DNS systems. Specifically, *Beser* teaches that queries involving the unique identifier [e.g., a domain name] may be encrypted. *Beser* at col.11, ll.22-25 ("The IP 58 packets may require encryption or authentication to ensure that the unique identifier cannot be read on the public network 12.").

Beser thus teaches that encryption can be used in IP tunneling systems in various ways to support secure communication links (e.g., use of IPSec-compliant systems, use during establishment of the secure communication link). In fact, *Beser* specifically identifies *Kent* (i.e., the RFC describing the IPSec protocol) as the appropriate way to integrate encryption in IP tunnels. Thus, contrary to Patent Owner's assertions, a person would have read *Beser* with *Kent*, and from them would not "understand that *Beser*'s tunneling technique as intended as an alternative to encryption." Response at 37. Patent Owner's strained reading of *Beser* and *Kent* is incorrect and should be disregarded.

b. *Beser* and *Kent* Disclose a DNS Proxy Module that Intercepts DNS Requests Sent by a Client

Patent Owner next asserts that the combination of *Beser* and *Kent* does not disclose or suggest a domain name server proxy module that intercepts DNS requests sent by a client. Response at 38. Here, Patent Owner makes three arguments.

First, Patent Owner asserts that *Beser* "discloses a request to initiate a VoIP association" but not a DNS request as required in the claims. Response at 38. It concedes that the "unique identifier" of *Beser* "may be a domain name," but then concludes that "merely including a domain name in the request to initiate a VoIP association does not transform it into a request for an IP address." Response at 38. As demonstrated in the Request, *Beser* describes a process for initiating an IP tunnel between two devices by sending a unique identifier (e.g., a domain name) to a trusted-third-party network device (e.g., domain name server). Request at 121-22 (citing *Beser* at col.10, ll.37-42, and col.11, ll.32-36). The DNS server in this embodiment returns an IP address in response to the DNS resolution request – which is the function of a DNS server. The IP address that is returned is then used by the trusted third party network device to establish the secure IP tunnel. Similarly, *Beser* shows the unique identifier being used to establish a VoIP (i.e., a "Voice Over Internet Protocol") connection that inherently uses IP addresses. Request at 128. Similarly, a request by domain name for multimedia or WebTV content necessarily would include a request for an IP address corresponding to that domain name. How Patent Owner concludes *Beser* does not show a process that comprises a "request for an IP address" is a mystery – obviously, it does.

Second, Patent Owner argues that nothing in *Beser* "discloses, teaches, or suggests that the trusted-third-party network device . . . may function as a DNS proxy module that intercepts

DNS requests sent by a client.” Response at 38. Here, Patent Owner attempts to read limitations and requirements into its claims. As is well known in the art, a “proxy” device is merely a device or application that acts as an intermediary between two other devices. *Beser* explains that the trusted-third-party network device can be a domain name server and the server may be distributed over several devices in several locations. *Beser* at col.11, ll.32-36 (“In one exemplary preferred embodiment, the trusted-third-party network device 30 is a ... domain name server . . . and may distributed over several physical locations”); *see* Request at 121-22 (citing *Beser* at col.11, ll.32-36). At least in the situation where the domain name server is distributed over multiple devices, the trusted-third-party device would act as a proxy—i.e., an intermediary—between the first network device and the domain name server.

Finally, Patent Owner asserts that *Beser* does not disclose “intercepting” a DNS request because nothing in *Beser* indicates that the process of negotiating an IP tunnel is “transparent” to the user. Response at 38-39. Again, Patent Owner is wrong. *Beser* explains that the trusted-third-party network device makes the “association of the public IP address for the second network device 16 with the unique identifier.” *Beser* at col 11, ll.30-32. After the trusted-third-party device is informed of a request, if the unique identifier (e.g., domain name) is associated with a secure destination, the trusted-third-party network device negotiates an IP tunnel with the destination. *Beser* at col.11, ll.9-10, 59-62. This process occurs with no further action by the user. *See id.* at col.9, ll.29-35, col.12, ll.6-19 (“negotiation may occur through the trusted-third-party network device 30 to further ensure the anonymity of the telephony devices (24, 26).”). Thus, when a request for a secure server is made, the trusted-third-party network device automatically initiates a secure connection before returning the IP address, thereby intercepting the DNS request. The Examiner’s conclusions regarding this claim element were, thus, correct.

Next, Patent Owner contests that *Beser* and *Kent* show the step of “determining whether the intercepted DNS request corresponds to a secure server.” Here, Patent Owner asserts that *Beser*’s description of processes using an “edge router” do not show how this “determining” step “corresponds to this alleged secure server.” Response at 39 (quoting the Office Action at 28-29). In particular, Patent Owner asserts that comparing a request against a table of subscribers “simply does not disclose that this list of numbers has any purpose related to security.” Response at 40.

Patent Owner's assertions rest on its incorrect assumptions about what the claims actually require. First, Patent Owner criticizes *Beser* as not showing that the edge routers or network devices attached to edge routers are "secure servers" because they do not communicate through an authenticated and encrypted channel. Yet, the claims impose no requirements as to how encryption is used in the process of establishing an IP tunnel – and *Beser* plainly shows encryption being used in establishment of IP tunnels. Next, Patent Owner asserts the steps in *Beser* where the unique identifier is compared to a list of authorized destinations are not done for "purposes related to security." Here, Patent Owner not only reads limitations into its claims, but also reads *Beser* in an illogical manner. Specifically, *Beser* shows that the "lookup" step is done to determine if the destination is an authorized destination – a reason plainly linked to security. Indeed, one premise of the VoIP embodiment being discussed in *Beser* is that all of the devices associated with one of the local networks are known to the "trusted third party network device" and are authorized destinations (e.g., other phones on a private network to which a connection is being established).

And because the claims impose no "security purpose" requirements, Patent Owner's assertions are ultimately irrelevant – all that matters is that the numbers included on the list are associated with a secure destination. In short, nothing in the claim term "determining" precludes the activity described in *Beser*—referencing an internal table or list—from meeting this claim requirement. Consequently, the Examiner's finding that this claim element was obvious based on *Beser* in view of *Kent* was proper and should be maintained.

c. *Beser* in View of *Kent* Renders Obvious Forwarding the DNS Request to a DNS Function That Returns an IP Address of a Nonsecure Computer When the Intercepted Request Does Not Correspond to a Secure Server

Patent Owner asserts that the combination of *Beser* and *Kent* does not disclose or suggest "when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer." Response at 41. Specifically, Patent owner asserts that the trusted-third-party network device takes no "forwarding" action. *Id.*

As explained in the Request, *Beser* describes a process where a unique identifier (e.g., a domain name) is used to establish an IP tunnel, and that a trusted-third-party network device can be a domain name server. Request at 164-65 (citing *Beser* at col.10, ll.37-41, and col.11, ll.32-

36 (“In one exemplary preferred embodiment, the trusted-third-party network device 30 is a ... domain name server”). Domain name servers function by evaluating domain names, and returning IP addresses associated with the domain. In the model described in *Beser*, a trusted-third-party network device will receive and then evaluate a request, compare it to a database of entries, and take additional actions to establish the IP tunnel based on the results of that evaluation. *See, e.g., id.* at col.11, ll.45-59. But nothing in *Beser* limits the inherent functionality of a domain name server that must be present in the trusted third party network device. Request at 130.

Moreover, it is well known in the art that domain name servers operate in the application layer of the OSI model. *Beser* explains that the trusted-third-party network device can receive and process requests to initiate a tunneling association on not only the application layer, but also lower layers of the OSI model (e.g., the transport layer). *Beser* at col.8, ll.52-57; *see id.* at Fig. 2. *Beser* further explains that the IP tunnel generally is negotiated on a lower layer of a protocol stack for the network devices. *Beser* at col.9, ll.35-37. Thus, *Beser* describes a system that operates on a lower level of the OSI model than the DNS server. Consequently, if a DNS request did not correspond to a secure server, the trusted-third-party device would forward the request to the DNS module by passing it up to the application layer, where the domain name server operates. The domain name server then would resolve the request, and return the IP address associated with the (non-secure) domain. Consequently, the Examiner’s finding that this claim element would have been obvious based on *Beser* in view of *Kent* was proper and should be maintained.

d. *Beser* in View of *Kent*, Renders Obvious Automatically Initiating an Encrypted Channel Between the Client and the Secure Server When the Request Corresponds to a Secure Server

Patent Owner asserts that the combination of *Beser* and *Kent* does not disclose “automatically initiating an encrypted channel between the client and the secure server.” Response at 42. Patent Owner also bases this assertion on its incorrect belief that *Beser* teaches away from using encryption techniques in tunneling connections. Response at 42-43. Patent Owner misreads the *Beser* and *Kent* and misrepresents what they would have suggested to a person of ordinary skill in the art.

Beser teaches processes where, in response to a request containing a unique identifier specifying the location of a second network device, a trusted-third-party network device will negotiate with first and second network devices to establish an IP tunnel between the first and second network devices. *Beser* further explains that the “negotiation may occur through the trusted-third-party network device 30 to further ensure the anonymity of the telephony devices (24, 26).” *Id.* at col.12, ll.6-19. The private network IP addresses are then used in conjunction with the public IP addresses of the first and second network devices to establish the tunnel between the first and second network devices. *See id.* at col.12, ll.28-37. This process occurs without any further action from or involvement of the user that made the request – it is thus “automatic.”

Beser also explains that IP traffic within a VPN IP tunnel is ordinarily encrypted using the techniques described in *Kent* and provides examples where encryption is used in establishing secure IP tunnels in its systems. *See, e.g., id.* at col.2, ll.36-40 (“It is therefore desirable to establish a tunneling association that hides the identity of the originating and terminating ends of the tunneling association from the other users of a public network. Hiding the identities may prevent a hacker from intercepting all media flow between the ends.”); col.12, ll.13-19 (“In this manner, the identities of the originating 24 and terminating 26 telephony devices are inside the payload fields 84 of the IP 58 packets and may be hidden from hackers on the public network. The negotiation may occur through the trusted-third-party network device 30 to further ensure the anonymity of the telephony devices (24, 26).”) *Beser*, thus, teaches IP tunnels in which at least some IP packets are encrypted – there is no express requirement in the claims that all secure traffic in the IP tunnels be encrypted. *Beser* further explains other than situations where it would be impractical, VPNs and encryption of IP traffic in IP tunnels using the IPsec protocol should be used. *See id.* at col.1, l.54 to col.2, l.18.

Kent describes use of IPsec to establish VPNs including by IP tunneling. *See, e.g., Kent* at 8 (“A tunnel mode SA is essentially an SA applied to an IP tunnel.”) The IPsec protocol calls for encryption of all IP traffic being sent between nodes of the VPN network – the protocol is designed to automatically encrypt traffic being sent between nodes. A person of ordinary skill in the art would have relied on *Kent* to modify the design of *Beser* to incorporate IPsec to encrypt all traffic being sent in IP tunnels between a first and second network device in the IP tunneling procedures being described in *Beser*, rather than to encrypt only the traffic used to

establish the IP tunnel. Consequently, the Examiner's finding that this claim element was obvious in view of *Beser* and *Kent* was proper and should be maintained.

1. Independent Claim 7

The Examiner correctly found that *Beser* in view of *Kent* renders obvious independent claim 7. In response, Patent Owner argues only that this claim recites features similar to those described for claim 1, and accordingly, *Beser* in view of *Kent* "does not disclose or suggest these features of claim 7 for similar reasons as those discussed above with respect to claim 1." Response at 43. Because the Examiner's rejection of independent claim 1 was proper, the rejection of this claim as being obvious based on *Beser* in view of *Kent* was also proper and should be maintained.

2. Independent Claim 13

The Examiner correctly found that *Beser* in view of *Kent* renders obvious independent claim 13. In response, Patent Owner argues only that this claim recites features similar to those described for claim 1, and accordingly, *Beser* in view of *Kent* "does not disclose or suggest these features of claim 13 for similar reasons as those discussed above with respect to claim 1." Response at 44.⁵ Additionally, the linguistic differences between claims 13 and 1 are insignificant. As even Patent Owner observes, "claim 13 recites features similar to those described . . . for claim 1." Response at 43. For example, step (i) of claim 1 specifies "determining whether the intercepted DNS request corresponds to a secure server" while step (i) of claim 13 specifies "determining whether a DNS request sent by a client corresponds to a secure server." As explained in the Request, these elements are anticipated by *Beser* because *Beser* discloses comparing a DNS request against a table of subscribers to determine whether the request corresponds to a secure destination. Request at 145 (citing *Beser* col.11, ll.45-59). There is thus no substantive difference between comparing an "intercepted DNS request" to a table and comparing "a DNS request sent by a client" to the table. Patent Owner surely would agree with

⁵ Patent Owner attempts to rely on an obvious typographical error in the Request to argue for patentability of claim 13. Specifically, Patent Owner criticizes the Request as referring to the language used in claim 1 in support of the proposed rejection of claim 13. In reality, the Request quoted the language of claim 13 in its entirety. Request at 143. It then discussed the disclosures in *Beser*, but, instead of comparing those to the elements of claim 13, a comparison was made to the elements in claim 1. See Request at 145. This obvious error was recognized by the PTO, which correctly rejected claim 13 based on the disclosures in *Beser* and *Kent*.

this statement – it incorporates by reference its arguments pertaining to claim 1 to respond summarily to the rejection of claim 13. Response at 44. Thus, the Examiner’s rejection of independent claim 1 was proper, so the Examiner’s rejection of this claim as obvious by *Beser* in view of *Kent* was also proper and should therefore be maintained.⁶

3. Dependent Claims 2, 8, and 14

The Examiner correctly found that *Beser* in view of *Kent* renders obvious dependent claims 2, 8, and 14. These three claims depend from claims 1, 7, and 13, respectively, and are substantively the same. In response, Patent Owner asserts that *Beser* and *Kent* do not render obvious the additional requirement that “when the client is authorized to access the secure server, sending a request to the secure server” to establish a secure channel between the secure server and the client. Response at 44-45.

To reach this conclusion, Patent Owner first challenges the observation in the Request and the Office Action finding that *Beser* shows that authentication of clients occurs by the inherent operation of the authentication steps referenced in *Beser*. In response, Patent Owner asserts that the Request “merely provides a generalized example of how servers requiring authentication request credentials.” Response at 44. So, despite admitting that *Beser* shows authentication being required before establishment of the tunnel, and that the steps constituting authentication are well known, Patent Owner somehow contends that *Beser* does not show “authentication of a client computer in conjunction with a tunneling association.” Patent Owner’s specious theories should be disregarded – they are plainly incorrect and illogical. For example, the obvious purpose of authenticating a user at the start of the process described in *Beser* is “in conjunction with” establishing an IP tunneling association. For similar reasons, Patent Owner’s incorrect and irrelevant complaint that *Beser* “discloses no reason for requiring the client to be authorized...” should be disregarded. Here, Patent Owner admits the authentication steps required by claims 2, 8 and 14 are actually described in *Beser* – it argues that disclosure should be disregarded because Patent Owner and its Expert cannot deduce from *Beser* why this authentication step is being required. Response at 44-45. The obvious reason is to

⁶ Requester also observes that a claim term in step (iii) of claim 13 lacks an antecedent basis. Step (iii) recites “when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel between the client and the secure server.” There is no antecedent basis for the term “the intercepted DNS request” in claim 13.

prevent unauthorized use of the systems being described in *Beser*. As explained in the Request, *Beser* discloses that “IP 58 packets” sent to the trusted-third-party network device “may require encryption and authentication to ensure the unique identifier cannot be read on the public network.” Request at 132 (quoting *Beser* at col.11, ll.22-24). *Kent* also explains that IPSec provides multiple security services, including, *inter alia*, data origin authentication and access control. *Kent* at § 2.1. Access control, which involves preventing unauthorized use of a resource or data, necessarily involves determining if a requestor is authorized to access a specified resource or data. *Id.* at Appx. A. Thus, alone or when considered in view of *Kent*, the authentication step described in *Beser* teaches the step of “determining whether a client is authorized to access the secure server.”

Patent Owner also argues that *Beser* does not disclose both a DNS request and a “request sent to the secure server to establish the encrypted channel when the client is authorized to access the secure server.” Response at 45. Patent Owner’s analysis overlooks the clear disclosures made in *Beser*. After the trusted-third-party network device receives a request to initiate a tunneling association and has associated the unique identifier with the terminating end of the tunnel, the device sends an IP packet to the second network device to begin negotiating a secure channel with the second network device. *Beser* at col.13, ll.41-48. Thus, *Beser* discloses both a DNS request and a request sent to the secure server to establish the secure channel. Accordingly, the Examiner’s rejection of these claims as rendered obvious by *Beser* in view of *Kent* was also proper and should therefore be maintained.

4. Dependent Claims 4, 10, and 16

The Examiner correctly found that *Beser*, in view of *Kent*, renders obvious claims 4, 10, and 16. Patent Owner disagrees, arguing that *Beser* and *Kent* do not disclose “a web browser into which a user enters a URL resulting in the DNS request.” Response at 46. Patent Owner maintains that, at the time of invention, it would not have been obvious to a person of ordinary skill in the art to use a web browser to initiate the tunneling association described in *Beser*. Response at 46. Patent Owner is incorrect. As described in the Request, *Beser* relied upon and referred to accepted standards and protocols relating to communications over the Internet. In light of this disclosure, a person of ordinary skill in the art would have recognized that a web browser, which was one of the most common methods of initiating communication with a remote server, was an obvious way to access multimedia content on a remote host. Consequently, the

Examiner's rejection of these claims as being obvious based on *Beser* in view of *Kent* was proper and should therefore be maintained.

5. Dependent Claims 5 and 11

The Examiner correctly found that *Beser* in view of *Kent* renders obvious claims 5 and 11. Patent Owner disagrees and asserts that *Beser* and *Kent* do not disclose an "IP address hopping scheme." Response at 47. Patent Owner then asserts that the NAT protocol is not an IP address hopping scheme, and that even if it is, *Beser* teaches away from using the NAT protocol. Response at 47-48. Patent Owner again advances incorrect and implausible readings of its claims and the prior art.

First, in reexamination proceedings claim terms are given their broadest possible construction consistent with the specification. The claims recite only an "IP hopping scheme between the client and secure server." The specification explains that "[t]he algorithm used for IP address-hopping can be any desired algorithm." '151 col.17, ll.41-42. The NAT protocol acts as an interface between a local network and a global network. When a local client requests content from an outside server, NAT works by changing the originating IP address in packets before forwarding the packets to the outside server. See RFC 1631 ("Network Address Translator") ("NAT itself can be seen as providing a kind of privacy mechanism . . . [because] machines on the backbone cannot monitor which hosts are sending and receiving traffic"). Thus, the NAT protocol is an IP address hopping scheme within the meaning of claims 5 and 11.

Second, nothing in *Beser* teaches away from using the NAT protocol. *Beser* notes that NAT is another method that can be used in tunneling, and identifies certain shortcomings in the NAT protocol. *Beser* goes on to say that NAT "may be" inappropriate for the transmission of multimedia or VoIP due to computer power limitations. However, nothing in *Beser* states that NAT should not be used or cannot be used in any tunneling associations. Moreover, the claims are not restricted to the "high volume" applications that prompted these comments in *Beser*. Thus, a person of ordinary skill in the art would not have read *Beser* to teach away from NAT in all applications. Accordingly, the Examiner's rejection of claims 5 and 11 as being obvious based on *Beser* in view of *Kent* was proper and should be maintained.

6. Dependent Claims 6 and 12

The Examiner correctly found that *Beser* in view of *Kent* renders obvious claims 6 and 12. As demonstrated in the Request, *Beser* discloses a method of preventing “the identity of the originating and terminating ends of the tunneling association from the other users of a public network.” *Beser* at col.2, ll.36-39. Patent Owner disagrees and asserts that, although *Beser* describes a method for keeping a client’s identity hidden from other users of the network at large, nothing in *Beser* describes hiding the identity of the originating device from the terminating device. Response at 48-49. Patent Owner’s analysis ignores the teachings of *Beser*. In discussing the negotiation of a tunneling association, *Beser* shows that the neither the second network device nor the device associated with the terminating end of the tunneling association is ever privy to the IP address of the originating device. *Beser* at col.13, ll.41-47; *id.* at col.24, ll.14-53 (showing the information available to the second network device and the terminating end of the tunneling association). At most, those devices know the address of the first network device, but not the originating device. Accordingly, the Examiner’s rejection of claims 6 and 12 as being obvious based on *Beser* in view of *Kent* was proper and should be maintained.

D. Response to Patent Owner’s Arguments Regarding the Rejection of Claims 1-16 Under 35 U.S.C. §102(a) Based on *BinGO* (Issue 3).

1. *BinGO* Expressly Incorporates *BinGO EFR*

Patent Owner presents an initial challenge to the use of *BinGO* as prior art; namely, it asserts that the *BinGO* publications should be limited to teachings found only in the Bingo User Guide (*BinGO UG*). Specifically, Patent Owner contends that *BinGO UG* does not properly incorporate by reference the contents of the Bingo Extended Features Reference (*BinGO EFR*). First, Patent Owner asserts that because the *BinGO UG* refers to the “Extended Features Reference” instead of the “Extended Feature Reference,” “it is not clear that *BinGO EFR* is even the correct document that is referenced in” the *BinGO UG*. Patent Owner’s semantic analysis of the title of the document that is being referred to in the *BinGO UG* is absurd. Both the *BinGO UG* and *BinGO EFR* refer to each other, and specifically describe how to configure and use features in the BiNGO! router. The *BinGO UG* for example does not simply refer generally to the *BinGO EFR*, but points to a particular section found in *BinGO EFR* for instructions on configuring the BinGO! router to implement VPN capabilities. Only by completely ignoring the

substantive contents of the *BinGO EFR* and *BinGO UG* documents could one even suggest that it is unclear that *BinGO EFR* is the document referenced by *BinGO UG*.

Next, Patent Owner takes issue with the version numbers on two different *BinGO EFR* documents that were made publicly available on the Internet. This is another red herring. As shown in Exhibit A (a screenshot of the Internet Archive listing at www.bintec.de), the only “Extended Feature[s] Reference” available on BinTec’s website in April 1999—the month after *BinGO* was published—was “Extended Feature Reference . . . Ver. 1.2.” Further, the “link” associated with the “Extended Feature Reference” on that website is the same as the “link” embedded in the *BinGO* pdf document: <http://www.bintec.de/download/brick/doku/71050a.pdf>. Thus, there is no uncertainty about which version of *BinGO EFR* was being referenced by *BinGO UG* because at the time the *BinGO UG* was published, there was only one version of *BinGO EFR* being publicly disseminated (i.e., version 1.2). Patent Owner’s apparent theory is that because the URL in Requester’s response to a petition in this proceeding presently can be shown to retrieve a different version of *BinGO EFR* (i.e., version 1.5 from 2003), there was some confusion when BinGO UG was published in 1999 about which version of *BinGO EFR* was being referenced in *BinGO UG*. As explained above, the evidence at the time of publication of *BinGO UG* demonstrates this is false.⁷ Thus, as explained in the Request, *BinGO UG* expressly incorporates by reference *BinGO EFR* (vers. 1.2).

2. Independent Claim 1

As explained in the Request, *BinGO* describes a system in which a secure channel is automatically established between a client and a secure server. Patent Owner disagrees, asserting that *BinGO* fails to disclose (1) “[d]etermining whether the intercepted DNS request corresponds to a secure server;” and (2) “[w]hen the intercepted DNS request corresponds to a

⁷ Patent Owner references an URL cited in Requester’s opposition to a Patent Owner petition contesting the status of *BinGO* as prior art. Specifically, Patent Owner contends that using this URL retrieves the *BinGO EFR* version 1.5 document. Yet, Exhibit A shows that the URL (i.e., <http://web.archive.org/web/19990417093944/http://www.bintec.de/download/brick/doku/71050a.pdf>) links to *BinGO EFR* v1.2, not the later “2003” link Patent Owner identifies in the Response. Response at 18. The 2003 link re-directs the request simply because that 1999-era link is no longer available. But, at the time the Internet Archive took a snapshot of this webpage in April of 1999, the 1999 link would have been active, and version 1.2 of the “Extended Feature Reference” would have been the only *BinGO EFR* that was accessible.

secure server, automatically initiating an encrypted channel between the client the secure server.” Each of these assertions is incorrect.

a. *BinGO* Discloses “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

The Examiner correctly found that *BinGO* discloses all the limitations of claim 1. In response, Patent Owner contends that the Request “admits that the BinGO! router does not perform the recited ‘determining’ step at all; rather they allege that a separate DNS server performs the ‘determining’ step.” Response at 22. Patent Owner misunderstands the Request, mischaracterizes *BinGO* and ignores the actual claim requirements.

The Request actually explains that the *BinGO* system could be configured to operate in a variety of ways. One of these would cause the BinGO! router to “function as a DNS proxy server.” Request at 91. In that configuration, “the BinGO! router would use the local DNS server containing the entries of secure destinations to determine if the DNS request specified a secure server.” Request at 91. As explained in both *BinGO UG* and the Request, the local DNS being used in this configuration was a collection of secure names associated with the secure server. See *BinGO UG* 88-89 (“One possibility would be to set up your own Domain Name Server in which all the names of the PCs in your partner’s network and their corresponding IP addresses that you want to reach are listed.”) In this configuration, the BinGO! router is designated as the DNS proxy server, and uses this local DNS server to support its evaluation of DNS requests coming from client computers on the local network. See *BinGO UG* at 87. Thus, contrary to Patent Owner’s assertion, the Request explains precisely how *BinGO* teaches that the BinGO! router uses a local DNS server to determine if an intercepted DNS request corresponds to a secure server.

Under the broadest reasonable construction of the claims, there is no requirement that all of the functionality required to determine if a DNS request specifies a secure server reside in a single computer. In fact, in this case, the specification of the ’151 patent explains that these functions may be distributed among different services and computers. See ’151 Patent at col.38, ll.30-34 (“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.”) Patent Owner’s tortured analysis of the Request, *BinGO* and its own patent claim language—

which reads non-existent limitation into the claim requirement of “determining”—should thus be disregarded.

Patent Owner next contends that *BinGO* does not disclose “determining whether a DNS request corresponds to a secure server.” Response at 22. Initially, Patent Owner again assumes the claims restrict “how” such a determination is made or expressly define the attributes of a “secure server.” Both assumptions are incorrect—the claim language does not limit *how* the “determination” must be made or define the minimum requirements of a “secure server.” Moreover, *BinGO* plainly does disclose “determining whether a DNS request corresponds to a secure server.” As the Request explained, *BinGO* shows the configuration of a BinGO! router to evaluate whether a DNS request specifies a secure remote network destination, and if so, that request would be routed to the secure remote network. See *BinGO UG* at 88-89; Request at 91-94. In that configuration, the DNS request is clearly intercepted, compared to a pre-defined set of secure server destinations, and routed to the secure server destination if the name or IP address matches an entry in that data set.⁸ *Id.*

In another embodiment, *BinGO* explains that the BinGO! router could be configured to determine if a destination is a secure website by checking a VPN (i.e., a secure connection) menu list and optionally setting up a VPN if the request matched a destination on that list. See *BinGO EFR* 73-81. *BinGO* shows how to set up a VPN entry for a particular PPP partner, which explains how to generate menu entries that are checked to determine if a partner is secure. *Id.* For instance, *BinGO* discloses that the BinGO! router may verify the VPN partner by the IP address the VPN partner can be reached at on the Internet.” See *BinGO EFR* at 76. Since *BinGO EFR* expressly applies to the BinGO! router, one of ordinary skill would have understood that *BinGO* possesses this functionality.

⁸ At pages 19-20 of the Response, Patent Owner presents an extended discussion of different ways that a BinGO! router could be configured to communicate with a secure remote network. Patent Owner describes one configuration where a BinGO! router could first connect to an ISP, and through that, reach the secure network. Then, Patent Owner discusses a second configuration where the BinGO! router communicates directly with the secure network. Patent Owner ignores, of course, the explanations in *BinGO* that showing the BinGO! router can be configured to conditionally connect directly to either based on the destination specified in the DNS request. See *BinGO UG* at 90-92 (showing use of default and alternate paths). Patent Owner’s comments are ultimately irrelevant, as the claims impose no restrictions on the path that communications from a client computer must take to reach the secure server destination.

Patent Owner next takes issue with *BinGO's* disclosure of different configurations of the BinGO! router and associated networks, each of which describes all the limitations of claim 1. For example, the Request explains that "*BinGO* also describes a configuration where the BinGO! router has not been configured to have an ISP as a WAN partner." Request at 92. In this configuration, *BinGO* discloses "determining whether a DNS request corresponds to a secure server" because all requests that could not be resolved locally (i.e., computers outside of the LAN) would be routed to a DNS server on a corporate network, where the determination would be made if the request was specifying a secure destination (i.e., a computer on the corporate network) or a non-secure destination (e.g., a public web site on the Internet). Thus, in this second configuration, all DNS requests, for example, would be sent to the WAN Partner for resolution by the BinGO! router. Request at 93-94. In this configuration, the DNS requests are still resolved and the destination specified in the request dictates how the communications are to be handled. The fact that the WAN Partner plays a role in determining if a request is seeking access to a secure server or was specifying a non-secure public website is immaterial to claim 1, which does not restrict "how" that determination must be made or limit a DNS proxy server implemented within a single computer. As noted above, the specification makes clear that a "DNS proxy server" can comprise multiple computers that are linked via a network. The Request thus does not "mix[] and match[]" portions of *BinGO* as Patent Owner contends, but simply describes the alternative configurations expressly described in *BinGO* that satisfies this limitation of claim 1.

b. *BinGO* Discloses "When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server"

The Examiner correctly found that *BinGO* discloses the limitation "when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client the secure server." In response, Patent Owner appears to not contest that the actions taken by the BiNGO! router to establish communications with a server are automatic, but instead simply asserts those automatic actions are not done in response to a determination that a request corresponds to a secure server. Request at 25. In particular, Patent Owner contends that "...nowhere does BinGO explain how or whether this encryption occurs in conjunction with a DNS request, let alone with intercepting a DNS request or determining whether a DNS request

corresponds to a secure server.” *Id.* Instead, Patent Owner contends that “any encryption measures for communicating with a WAN partner are established as part of a manual WAN partner configuration.” *Id.* Patent Owner also criticizes the Request for allegedly failing to identify within *BinGO* disclosures meeting every limitation of the claim. This, of course, is incorrect – the Request precisely identifies the portions of *BinGO* that describe the various elements of claim 1. *See* Request at 94-95.

The first theory Patent Owner advances is that this element cannot be met because Patent Owner does not believe *BinGO* describes a system that determines if an intercepted DNS request corresponds to a secure server. As explained above, Patent Owner is incorrect based on the clear explanations in *BinGO* that intercepted DNS requests are evaluated to determine if they are requesting access to a secure server (e.g., on a remote corporate network) or a non-secure server (e.g., a public website on the Internet). *BinGO* also plainly shows that the automatic initiation of the VPN occurs in response to the determination that the request is seeking access to a secure server. This, again, is part of the inherent functionality of the *BinGO* systems.

Patent Owner next criticizes the Request and the *BinGO* disclosure by asserting that the encryption measures employed by a BinGO! router “are established as part of a manual WAN partner configuration.” Of course, once the BinGO! router is configured, it then will automatically establish an encrypted channel with the secure server using the particular encryption parameters and techniques specified in this “manual” configuration in response to the specific DNS requests being intercepted by the BinGO! router. Once configured, the BinGO! router will handle each request without further user involvement, and those DNS requests will automatically trigger routing and encryption, as the claims specify.

This is explained clearly in *BinGO UG*. For example, *BinGO UG* at 90 explains that it uses pre-defined routes stored in the BinGO! router to communicate with pre-defined destination servers, and that the address specified in the particular packet dictates where it will be routed to:

The routes lead to a certain network with a defined “network address and “netmask.” You must **specify the route to every network you want to access.** You could define, for example, the route to **your WAN partner (e. g. head office).** **All packets whose IP addresses belong to the netmask and network address are sent to the partner network.**

BinGO UG further explains:

Not only does BinGO! avail of a default route, your PC also has one: the gateway. **All packets whose destinations are not within the local network are sent by your PC to this gateway.** BinGO! serves as this gateway. **As soon as your router receives such a packet, it forwards it in turn to one of its known routers** (e.g. to the provider or to another partner's network). (emphasis added)

BinGO also explains that each "partner network" is pre-defined, and that many parameters of communications with each partner network are specified in advance as part of the configuration of communications with that partner. For example, *BinGO UG* at 265 shows that one can designate the type of encryption (if any) to be used in communications with a particular WAN partner. This is an attribute of the configuration of a WAN partner for which a default or other route has been specified. *See, e.g., BinGO UG* at 147 (describing configuration parameters for a WAN partner, with encryption as optional and types of encryption being specified); *see also id.* at 149. The pre-configured WAN partner entries stored on the BinGO! router define the parameters of communications between the BinGO! router and the WAN partner, including whether and what type of encryption is to be used in those communications. Thus, when an IP packet destined for a WAN partner is received by the BinGO! router, the router establishes the communications – including with the specified type of encryption – and all of this happens without further user involvement.

When *BinGO* initiates an encrypted channel, it must first determine whether the secure destination can, in fact, communicate in an encrypted channel. If it can, then *BinGO* will set up an encrypted channel in response to that determination. *BinGO* also explains that the BinGO! router may verify the VPN (i.e., the encrypted channel) partner by "the IP address the VPN partner can be reached at on the Internet." *See BinGO EFR* at 76. By using the IP address (obtained from the DNS request) to verify a VPN partner, it is using the IP address to determine whether to set up a VPN, as required by the claim. Accordingly, Patent Owner's contention that "it is impossible to know" *BinGO* automatically initiates an encrypted channel in response to an intercepted DNS request is simply not true. Response at 25.

Patent Owner also criticizes the *BinGO* disclosure for "providing no guidance on what steps occur before the alleged 'automatically initiating an encrypted channel'" occurs. Of course, neither the claims nor the specification of the '151 patent provide such details. More to the point, these operational details are, in fact, precisely explained in *BinGO* – the fact that the Patent Owner and its expert are for some reason unable to comprehend them does not mean they

are not there. Patent Owner also ignores the various passages in *BinGO* that describe possible operating configurations of the BinGO! router and networks built around these routers. Patent Owner's criticism of *BinGO* based on possible configurations of the BinGO! router ultimately are pointless – as explained in the Request, other, typical configurations are described that will cause the BinGO! router to automatically initiate an encrypted channel between a client and a secure server as the claims require. Request at 94-95.

In one of its irrelevant hypotheticals on page 25 of the Response, Patent Owner asserts that “the manual configuration of the corporate network WAN partner might result in all communications being encrypted, whether directed to a computer on the corporate network (i.e., an alleged secure server) or to a computer on the Internet (i.e., an alleged nonsecure server). This description ignores the actual examples in *BinGO*, which do not route all network traffic in this manner. For example, page 17 of *BinGO* at page 17, explains:

Additionally, a significant advantage of your *BinGO!* is the means by which access to networks is achieved. When using a modem/ISDN-card, you must expressly dial your Internet provider in order to send an e-mail, for example. On the other hand, the router decides independently (once configured, that is) if and how a connection to the Internet provider is established. If you submit an external WWW-address with your browser, for example, your *BinGO!* realizes that the requested address lies outside your own LAN, thus automatically establishes a connection with your provider and the Internet. This procedure is particularly economical as your router disconnects you after a predefined time subsequent to a cessation in external data exchange.

The Patent Owner also contends that the passages in *BinGO EFR* identified in the Request “fail to discuss the BinGO! router at all” and are furthermore are only options for “various BRICK routers.” Response at 26. Patent Owner is obviously incorrect. The section cited in the Request to show this claim requirement is entitled “Virtual Private Networking.” *BinGO EFR* at 73. In the *BinGO UG*, at section 7.5.1—entitled “VPN (Virtual Private Network)”—the user guide explains that “you can find detailed information and configuration instructions (with examples) [for the BinGO! router] in Extended Feature Reference.” Clearly, this section would direct a person of ordinary skill in the art to use the “Virtual Private Networking” section of *BinGO EFR* to configure a BinGO! router. For this reason, Patent Owner's argument is both incorrect and irrelevant. Further, Patent Owner ignores that the *BinGO publications* that anticipate each and every claim of the '151 patent—not a hypothetical

BinGO-based system that results from the possible configuration options chosen by Patent Owner.

Next, Patent Owner asserts that the Request has “change[d] course and directly contradicts its ‘determining’ step arguments by asserting that a client would in fact connect to an ISP if the DNS request specified a secure server.” Response at 27. Patent Owner misunderstands the Request and the unambiguous disclosure of *BinGO*. The Request never contends, as Patent Owner claims, “that a client would connect to an ISP *only if the request did not specify a secure destination*, i.e., did not specify a ‘computer[] on a corporate network.’” Response at 27. If it had, Patent Owner certainly would have quoted that text, rather than paraphrasing and removing phrases from the Request. Indeed, the Request explains that the *BinGO* router would “send the request to a secondary DNS server (e.g., one associated with an ISP . . .”), if the local DNS server “[could] not resolve the address (*i.e.*, because the request did not specify a secure destination.”). Request at 92. Certainly Patent Owner would agree that there is a distinction between connecting to a DNS server associated with an ISP and connecting to an ISP directly. In sum, the Request never makes the assertion that is the basis of Patent Owner’s claims, and as a result, Patent Owner’s response should be disregarded.

Finally, Patent Owner repeats its incorrect assertion that the Request “fail[s] to demonstrate that the alleged encrypted channel is ‘automatically’ initiated.” Response at 28-29. For the same reasons discussed above, Patent Owner is incorrect. Consequently, the Examiner’s rejection of claim 1 based on *BinGO* was proper and should be maintained.

3. Independent Claims 7 and 13

The Examiner correctly found that *BinGO* discloses each of the requirements of claims 7 and 13. In response to the rejection of claim 7, Patent Owner presents no response distinct from its response to the rejection of claim 1. Response at 29. Because the rejections of claim 1 were proper, the Examiner’s rejection of claim 7 based *BinGO* also was proper and should be maintained.

In response to the rejection of claim 13, Patent Owner contends that the Request has “ignore[d]” the difference in claim language between claims 1 and 13. Patent Owner is incorrect. First, the only relevant distinction in the claims noted by Patent Owner is claim 13’s recitation of “automatically creating a secure channel,” as compared to claim 1’s recitation of “automatically initiating an encrypted channel.” Response at 29. The Request notes that “claim

13 is directed to subject matter similar to that recited in claim 1,” Request at 109, and the only distinction identified by Patent Owner here—“creating a secure channel” versus “initiating an encrypted channel”—is irrelevant to the Examiner’s finding that *BinGO* describes a system that anticipates claim 13. “Initiating an encrypted channel” is a narrower requirement than claim 13’s “creating a secure channel,” *see* ’504 ACP at 33 (explaining that a secure communication link does not require encryption), so a finding that *BinGO* satisfies that element from claim 1 necessitates a finding that it satisfies the broader element in claim 13. Consequently, the Examiner’s rejection of claim 13 based on *BinGO* was proper and should be maintained.

4. Dependent Claims 2, 8, and 14

The Examiner correctly found that *BinGO* describes a system that anticipates claims 2, 8 and 14. In response, Patent Owner contends that *BinGO* does not disclose the feature of “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.” Response at 30. Specifically, Patent Owner asserts that *BinGO* does not describe “‘sending a request to the secure server to establish an encrypted channel,’ as recited in step(b) of claims 2, 8, and 14.” Patent Owner is incorrect. As explained in the Request, *BinGO* describes the use of conventional authentication techniques. Request at 96 (citing *BinGO UG* at 242 (“PAP, CHAP and MS-CHAP are the common procedures used for authentication of PPP connections. These use a standard procedure to exchange a user ID and a password for checking the identity of the far end.”)) *BinGO* also explains that it provides additional mechanisms to authenticate users, such as call-back functionality in which a remote user accessing a BinGO! router is called back to establish the connection. Request at 96 (citing *BinGO UG* at 40 (“Before every connection, BinGO! and the router at HQ check the incoming data to see if they should take the call. In order to protect the network against unauthorized access, acceptance of the call only takes place after correct authentication. This authentication is based on a common password and two codes that you and your partner use for the connection.”); *BinGO UG* at 175-176 (authentication required for access to corporate network); *BinGO EFR* at 84-85 (“Both the ISP and the VPN Server will typically want to verify the initiating partner during connection establishment. Authentication is performed inband using PAP, CHAP, or MS-CHAP.”)).

Patent Owner ignores these significant and extensive discussions of various embodiments where a client computer is required to secure authorization from a secure server in order to

establish an encrypted channel. Accordingly, the Examiner's rejection of this claim was proper and should be maintained.

5. Dependent Claims 4, 10, and 16

The Examiner correctly found that *BinGO* describes a system that anticipates claims 4, 10 and 16. In response to the rejection of these claims, Patent Owner asserts no response distinct from its response to the rejection of claims 1-3, 7-9 and 13-15. Response at 14. Because the rejection of those claims was proper, the Examiner's rejection of claims 4, 10 and 16 is proper and should be maintained.

6. Dependent Claims 5 and 11

The Examiner correctly found that *BinGO* discloses every limitation of dependent claims 5 and 11. In response, Patent Owner asserts that the Request does not show "establishing an [IP] address hopping scheme between the client and the secure server" because the disclosed NAT protocol schemes in *BinGO*, in part, only "ensure[] that a connection partner uses only a single IP address." Response at 32. Patent Owner ignores the fact that this only one of the "four purposes" of using a NAT protocol scheme in *BinGO*. Request at 97-98. The NAT protocol scheme has other implementations that include, for example, "hiding the internal host address of a LAN by remapping to one or more external addresses." Request at 98.

Patent Owner also does not substantively address *BinGO*'s disclosure of "Open Shortest Path First" protocol. Rather, it only asserts, without any foundation, that this IP address hopping communication scheme has not been shown to "apply" to the BinGO! router. Response at 33. Of course, Patent Owner presents nothing to demonstrate that *BinGO* would be read by a person skilled in the art in a way that suggests these techniques would not apply to BinGO! routers. Instead, Patent Owner simply relies on the putative absence of an explicit statement explaining that OSPF does apply to BinGO! routers. In fact, *BinGO EFR* would have led a person of ordinary skill in the art to conclude that OSPF would apply to BinGO! routers. For example, in the "What's Covered in this Guide" section, *BinGO* states: "Chapter 2 OSPF describes using the Open Shortest Path First interior routing protocol on your BinTec router." *BinGO EFR* at 3. Accordingly, the Examiner's rejection of these claims as anticipated by *BinGO* was also proper and should be maintained.

7. Dependent Claims 6 and 12

The Examiner correctly found that *BinGO* discloses every limitation of dependent claims 6 and 12. To assert that *BinGO* does not describe this step, Patent Owner presents an obviously incorrect portrayal of the claim requirements, the *BinGO* procedures, and the Request. Specifically, at page 34, Patent Owner incorrectly assumes that the features relied upon in the Request to show, for example, an encrypted channel, are only “disclose[d] in conjunction with BRICK routers.” This is simply not true, as explained above in section 2(b). The disclosure in *BinGO EFR* of virtual private networking applies equally to both BinTec routers (BinGO! and BRICK). See, e.g., *BinGO EFR* at 3.

Patent Owner next recycles its incorrect assertion that the Request contends “that a client would connect to an ISP *only if the request did not specify a secure destination*, i.e., did not specify a ‘computer[] on a corporate network.’” Response at 34. As explained above, Patent Owner misunderstands the Request and the unambiguous disclosure of *BinGO*. In reality, the Request demonstrates that the *BinGO* router would “send the request to a secondary DNS server (e.g., one associated with an ISP . . .”), if the local DNS server “[could] not resolve the address (i.e., because the request did not specify a secure destination.”). Request at 92. Patent Owner’s incorrect response should be disregarded.

Patent Owner also contends that the “Request fails to specifically explain how the mere use of encryption either explicitly or inherently would avoid sending a true IP address to the client.” Response at 35. This is simply incorrect. As explained in the Request, the LAN-to-LAN configuration shows that the two BinGO! routers send encrypted communications to each other via the Internet. Request at 99. The encryption and decryption of traffic between the devices (that is destined for the secure computers, for example) in this configuration is handled exclusively by the BinGO! routers. Request at 99. As indicated in this example, individual hosts are not required to support PPP or PPTP, the VPN remains transparent. Thus, the client computers on each LAN (the “hosts”) communicate only with the BinGO! router, and would not receive the IP address of the remote host. Request at 99 (citing *BinGO UG* at 265; *BinGO EFR* at 83-85 (“data encryption/decryption is performed at each end of the tunnel”). Accordingly, the Examiner’s rejection of this claim was proper and should be maintained.

E. There are No Secondary Considerations Linked to the Claims

Patent Owner provides alleged secondary considerations that are little more than unsupported and self-interested statements from its own Chief Technology Officer, Robert Short.

First, Patent Owner contends that there was “long felt need for a system that could establish secure communications, such as an encrypted channel, in a simple and straightforward manner.” Response at 116. However, Patent Owner has not demonstrated that claimed invention, rather than the prior art secure communication systems (e.g., *Aventail*, *Beser*, *BinGO*, *Kent* or combinations thereof) are responsible for addressing these long-felt needs.

Similarly, the Patent Owner contends there is evidence of significant commercial success. Initially, the putative evidence of commercial success is not evidence of commercial success of any product or service. Instead, Patent Owner refers only to licensing revenue -- which is not probative of commercial success of a claimed product or method. In addition, Patent Owner provides no evidence that establishes that whatever commercial success the Patent Owner’s company has experienced—which is apparently limited collecting licensing revenue—is attributable to the features of the claimed invention. Plainly it is not. Consequently, the self-serving, non-objective statements of its employee simply are not evidence of secondary indicia of non-obviousness, much less is probative evidence of the commercial success of the methods or systems that are claimed. Consequently, the Office should disregard these statements and give them no weight in assessing the obviousness of the claimed methods and articles.

F. Conclusions

As is evident from its responses to each of the rejections imposed by the Office, Patent Owner’s arguments are uniformly based on its belief that the patent claims expressly incorporate a large number of limitations and requirements. The basis for that belief is plainly not the claim language. For example, Patent Owner frequently points to its theory of how its invention functions, what it believes is described in the ’151 patent, or, simply, what it wishes its invention to be. Similarly, in criticizing the teachings in the prior art, Patent Owner frequently resorts to putative distinctions between the systems and methods of the claims and those being described in the prior art. Again, however, those criticisms rest on hypothetical claims that do not correspond to the actual claims of the ’151 patent. Requester, thus, urges the Office to maintain the rejections, as they are based on the broadest reasonable construction of the actual claim language used in the claims of the ’151 patent, and not the Patent Owner’s hypothetical claims or concepts.

For all of the reasons set forth above, the Third Party Requester contends that the Patent Owner has not rebutted the Examiner's rejection of the claims on any of Issues 1-6 of Office Action of April 20, 2012. The rejection of all the claims under each of those Issues should, accordingly, be maintained.

Respectfully submitted,

/ Jeffrey P. Kushan /
Reg. No. 43,401
Attorney for Third Party Requester

SIDLEY AUSTIN LLP
1501 K Street, N.W
Washington, D.C. 20005
tel. (202) 736-8000/ fax (202) 736-8711
Date: October 25, 2012

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re patent of Munger et al.	§	Merged Control Nos.: 95/001,714
	§	95/001,697
U.S. Patent No. 7,490,151	§	Group Art Unit: 3992
Filed: September 30, 2002	§	Examiner: Michael J. Yigdall
Issued: February 10, 2009	§	Confirmation No.: 3428
Title: Establishment of a Secure	§	Real Party in Interest:
Communication Link Based On	§	Cisco Systems, Inc.
a Domain Name Service (DNS) Request	§	
	§	

COMMENTS BY THIRD PARTY REQUESTER

PURSUANT TO 37 C.F.R. §1.947

Mail Stop *Inter Partes* Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TABLE OF CONTENTS

I. REPLY TO PATENT OWNER ARGUMENTS.....1

A. Response to Patent Owner’s Argument That Certain References Are Not Prior Art 1

B. The Rejections Under 35 U.S.C. § 102 Based on Kiuchi Were Proper..... 5

1. Kiuchi..... 5

2. Claims 1-4, 6-10 and 12-16 (ISSUE #7)..... 5

3. Claims 5 and 11 under 35 U.S.C. § 103(a) Based on Kiuchi in view of Martin (ISSUE #8)..... 10

4. Claims 1-4, 6-10 and 12-16 under 35 U.S.C. § 103(a) Based on Kiuchi in view of Edwards (ISSUE #14)..... 10

5. Claims 5 and 11 under 35 U.S.C. § 103(a) Based on Kiuchi in view of Edwards and Martin (ISSUE #15) 13

C. The Rejections Based on Wesinger Were Proper 14

1. Claims 1-4, 6-10 and 12-16 based on 35 U.S.C. § 102(e) (ISSUE #9) 14

2. Claims 5 and 11 based on 35 U.S.C. § 102(e) over Wesinger in view of Martin (ISSUE #10)..... 24

3. Claims 1-4, 6-10, and 12-16 based on 35 U.S.C. § 103 over Wesinger in view of Edwards (ISSUE #16)..... 24

D. The Rejections Based on Blum Were Proper (ISSUE #11)..... 28

1. Overview of Blum..... 28

2. Independent Claim 1 28

3. Independent Claims 7 and 13..... 31

E. Claims 1-4, 6-10 and 12-16 under 35 U.S.C. § 103(a) Based on Aziz in view of Edwards (ISSUE #12) 31

1. Aziz..... 31

2. Independent Claim 1 31

3. Independent Claims 7 and 13..... 36

4. Dependent Claims 2, 8 and 14 37

5.	Dependent Claims 3, 9 and 15	37
6.	Dependent Claims 6 and 12	37
7.	Dependent Claims 4, 10 and 16	38
F.	Claims 5 and 11 based on 35 U.S.C. § 103 over Aziz in view of Edwards and Martin (ISSUE #13).....	38
G.	The Rejections Under 35 U.S.C. § 102 Based on Aventail Were Proper (ISSUE #1).....	38
1.	Aventail.....	39
2.	Independent Claim 1	39
3.	Independent Claims 7 and 13.....	43
4.	Dependent Claims 2, 8 and 14	43
5.	Dependent Claims 3, 9 and 15	44
6.	Dependent Claims 4, 10 and 16	45
7.	Dependent Claims 5 and 11	45
8.	Dependent Claims 6 and 12	46
H.	The Rejections Under 35 U.S.C. § 102 Based on AutoSOCKS Were Proper (ISSUE #2).....	47
I.	The Rejections Under 35 U.S.C. § 102 Based on BinGO Were Proper (ISSUE #3)	47
1.	The BinGO User’s Guide Incorporates the BinGO EFR.....	47
2.	The BinGO User’s Guide.....	48
3.	Response to Patent Owner’s Arguments Regarding “Two Alternative Embodiments”.....	48
4.	Independent Claim 1	48
5.	Independent Claims 7 and 13.....	52
6.	Dependent Claims 2, 8 and 14	52
7.	Dependent Claims 3, 9 and 15	53
8.	Dependent Claims 4, 10 and 16	54
9.	Dependent Claims 5 and 11	54
10.	Dependent Claims 6 and 12	55

J.	The Rejections Under 35 U.S.C. § 103 Based on Beser in View of Kent Were Proper (ISSUE #4)	55
1.	Beser in View of Kent.....	55
2.	Beser and Kent Were Properly Combined.....	55
3.	Independent Claim 1	55
4.	Independent Claims 7 and 13.....	57
5.	Dependent Claims 2, 8 and 14	58
6.	Dependent Claims 4, 10 and 16	58
7.	Dependent Claims 5 and 11	58
8.	Dependent Claims 6 and 12	59
K.	Response to Patent Owner’s Argument That Secondary Considerations Demonstrate Non-Obviousness	59
II.	Non-Adopted Rejections (ISSUES #4, #5, #6).....	61
III.	Conclusion	61

LIST OF EXHIBITS

The present comments by third party requester Cisco Systems are accompanied by the following reference materials that, pursuant to 37 CFR 1.943, are excluded from the page limit restrictions.

- Exhibit F¹: Bradner, “The Internet Standards Process – Revision 3,” Request for Comments 2026, Internet Engineering Task Force (Oct. 1996).
- Exhibit G: Burnside & Keromytis, “Accelerating Application-Level Security Protocols,” The 11th IEEE International Conference on Networks, 2003, pp. 313-318.
- Exhibit H: Copy of catalog listing from Boston University Digital Common website, listing the Martin reference with an issue date of February 21, 1998.
- Exhibit I: Copy of Technical Reports Archive listing from Boston University Computer Science Department which includes a link to the Martin paper. The link to the Martin paper was archived at archive.org on January 22, 1998 and retrieved by the Wayback Machine.
- Exhibit J: Boston University Computer Science Department Technical Reports Instructions, available at: <http://www.cs.bu.edu/techreports/INSTRUCTIONS>.
- Exhibit K: U. Möller, “Implementation eines Anonymisierungsverfahrens für WWW-Zugriffe,” Diplomarbeit, Universität Hamburg (July 16, 1999), citing to Martin at page 77.
- Exhibit L: Joint Claim Construction Chart for US 7490151, *VirnetX v. Cisco*, No. 6:10-cv-00417, Docket No. 194 (Dec. 21, 2011) (selected pages).

¹ Exhibits A-E are part of the originally filed Request for Reexamination.

COMMENTS

On July 20, 2012, the Patent Owner filed the Patent Owner's Response to Office Action ("Response") for the Office Action mailed April 20, 2012 ("the April 20th Office Action") in connection with the above-identified *inter partes* reexamination proceeding, which was initiated by the Request for *Inter Partes* Reexamination filed August 16, 2011 ("the Cisco Request") and merged with the Request for *Inter Partes* Reexamination filed July 25, 2011 ("the Apple Request").

It is respectfully requested, for the reasons identified below, that the Examiner:

- (i) maintain his rejection of, and issue an action closing prosecution for, the original claims 1-16, and
- (ii) deem the arguments advanced by the Patent Owner in the Response to be erroneous, improper, and/or unpersuasive.

In the context of this *inter partes* reexamination, the standard provided in MPEP § 2111 for claim interpretation during patent examination is applied.

Requester's comments respond to the Patent Owner's arguments on an issue-by-issue basis, starting with the rejections originally posed in the Cisco Request, and then addressing the rejections originally posed in the Apple Request.

I. REPLY TO PATENT OWNER ARGUMENTS

First, Patent Owner tries to raise a series of procedural issues to avoid the Examiner's rejections by arguing that published and cited references are not printed publications. Patent Owner's argument is merely an attempt to avoid the substantive teachings of the prior art.

Second, Patent Owner raises improper and incorrect arguments that, in many instances, mischaracterize the prior art and improperly introduce new limitations into the claims. These arguments are unpersuasive and fail to overcome the Examiner's rejections.

A. Response to Patent Owner's Argument That Certain References Are Not Prior Art

Patent Owner argues, on pages 4-6, that the Office Action relies on five references "without showing that these references have been published." Patent Owner argues that (i) the Examiner did not indicate that the Patent Office investigated whether these references qualified as printed publications, and (ii) the Examiner improperly relied upon copyright dates to establish that a reference was known or used by others. Patent Owner's argument is without merit.

The Examiner's rejection was proper because these references are printed publications. A reference is a "printed publication" "upon a satisfactory showing that such document has been disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, can locate it." *In re Wyer*, 655 F.2d 221, 210 USPQ 790 (CCPA 1981). "An electronic publication, including an on-line database or Internet publication, is considered to be a 'printed publication' within the meaning of 35 U.S.C. 102(a) and (b) provided the publication was accessible to persons concerned with the art to which the document relates." MPEP §2128.

The Requester provides the following additional reference materials to show that the Examiner's determination that the cited references are printed publications was correct:

1. Aventail Connect v3.01/2.51 Administrator's Guide ("Aventail v3.01") and Aventail AutoSOCKS v2.1 Administration & Users' Guide v2.1 ("AutoSOCKS")

Aventail v3.01 is the administrator's guide to the Aventail Connect software and AutoSOCKS is the administrator's guide for the Aventail AutoSOCKS software. For each of these references, the Examiner was able to rely upon (i) the document itself containing a date of publication (the copyright date), (ii) the declaration of a former Aventail employee stating when copies of Aventail v3.01 and AutoSOCKS were distributed to customers (Apple Request, Exhibit E1, Declaration of Christopher Hopen, ¶¶7-9), (iii) the declaration of the editor of Network World stating when he received copies of Aventail v3.01 and AutoSOCKS (Apple Request, Exhibit E2, Declaration of Michael Fratto, ¶¶6-7), and (iv) the declaration of an employee at IBM stating when he received copies of Aventail v3.01 and AutoSOCKS and when he distributed copies Aventail 3.01 and AutoSOCKS to customers AutoSOCKS (Apple Request, Exhibit E3, Declaration of James Chester, ¶¶11-18).

Yet, despite the evidence from these parties establishing when they distributed and received copies of Aventail v3.01 and AutoSOCKS, Patent Owner argues, on page 5, that the Examiner should have provided *more* evidence of publication. The Examiner is not required to move mountains and find every last piece of evidence to establish publication – all that is required is a "satisfactory showing" that the document was made available. Ample evidence has been provided to satisfactorily show that Aventail v3.01 and AutoSOCKS were each a "printed publication." The Examiner's reliance on Aventail 3.01 and AutoSOCKS is proper.

2. Kent et al., "Security Architecture for the Internet Protocol" ("Kent")

The *Kent* reference was made available to the public in November 1998 in electronic form as an Internet Draft promulgated by the Internet Engineering Task Force (IETF). Patent Owner's argument that *Kent* is not a printed publication is without merit.

First, the *Kent* reference is self-dated as being available as of "November 1998" and unambiguously states on Page 1 that "Distribution of this document is unlimited." Accordingly, the *Kent* reference itself indicates that it was a "printed publication" within the meaning of 35 U.S.C. § 102.

Second, *Kent* is an early Internet Draft by the Internet Engineering Task Force (IETF).² The IETF promulgates Internet Drafts and "requests discussion and suggestions for improvements."³ The IETF's process for how the IETF publishes Internet Drafts is contained in "The Internet Standards Process – Revision 3" ("IETF Process Description"), which is attached as Exhibit F. The IETF Process Description states that Internet Drafts, such as *Kent*, are freely and widely distributed to interested individuals:

2.2 Internet-Drafts

During the development of a specification, draft versions of the document are made available for informal review and comment by placing them in the IETF's "Internet-Drafts" directory, **which is replicated on a number of Internet hosts**. This makes an evolving working document **readily available to a wide audience**, facilitating the process of review and revision.⁴

Accordingly, the *Kent* reference, an Internet Draft, was created by its authors and then made "readily available to a wide audience" by replicating the Internet Draft across "a number of Internet hosts."

Third, Patent Owner's expert, Dr. Keromytis, has repeatedly cited to *Kent* in his own peer-reviewed papers. As just one example, in "Accelerating Application-Level Security Protocols," Dr. Keromytis cited to *Kent* with a specific date of "Nov. 1998" (the same date listed on the first page of *Kent*). Dr. Keromytis' paper even included an annotation that *Kent* was available "[Online]" and even provided a URL link to *Kent*.

² See *Kent* at 1 ("This document is an Internet Draft. Internet Drafts are working documents of the Internet Engineering Task Force (IETF)...").

³ See *Kent* at 1.

⁴ Exhibit F, RFC 2026 at 8 (emphasis added).

Thus, since (i) the *Kent* reference itself provides a publication date of November 1998, (ii) the IETF has a policy of making Internet Drafts (such as the *Kent* reference) readily available to a wide audience by replicating the Internet Draft across a number of Internet hosts, and (iii) Patent Owner's own expert's papers relied upon an online copy of *Kent* and included a URL link to *Kent*, there is ample evidence showing that the *Kent* reference was disseminated and made available to persons interested and skilled in the subject matter in November 1998. Therefore, the *Kent* reference is a printed publication.

3. Martin, D.M., "A Framework for Local Anonymity in the Internet" ("*Martin*")

The *Martin* paper was published on-line by the Boston University Computer Science Department prior to the critical date of the '151 patent. First, the *Martin* paper itself is unambiguously dated on its face, "21st February 1998." Exhibit M provides a copy of the listing for the *Martin* paper as cataloged at Boston University, dated "1998-02-21." Exhibit N provides a copy of website <http://www.cs.bu.edu/techreports> archived by Archive.org and available through the Wayback Machine.⁵ The Wayback Machine establishes that the *Martin* paper was cataloged in the Boston University Technical Reports Archive and available to the public via the Internet even earlier than the February 21, 1998 date.

Second, Exhibit O is a German thesis,⁶ unambiguously dated 1999, that cites the *Martin* paper at page 77. Because this 1999 publication itself was published before the critical date of the '151 patent, and specifically cites to the contents of the *Martin* paper, it too establishes that the *Martin* paper was publicly disseminated prior to the critical date. Thus, since the *Martin* paper was (i) cataloged by Boston University; (ii) publicly available via Boston University's website; and (iii) actually used and cited by persons of ordinary skill in the art prior to the critical date, there is ample evidence showing that the *Martin* paper was disseminated and made available to persons interested and skilled in the subject matter in February 1998. Therefore, the *Martin* paper reference is a printed publication.

⁵ The Board of Patent Appeals and Interferences has recognized retrievals from archive.org as reliable evidence in establishing the date of a printed publication. See, Appeal 2007-0987 in application 09/810,992, dated May 24, 2007.

⁶ U. Möller, "Implementation eines Anonymisierungsverfahrens für WWW-Zugriffe," Diplomarbeit, Universität Hamburg (July 16, 1999).

Therefore, each of the references relied upon by the Examiner to reject claims 1-16 of the '151 patent were disseminated and made available to persons of ordinary skill in the art. The Examiner relied upon printed publications and the Examiner's rejections were proper.

B. The Rejections Under 35 U.S.C. § 102 Based on Kiuchi Were Proper

1. Kiuchi

Claims 1-4, 6-10 and 12-16 were rejected under 35 U.S.C. § 102 over Kiuchi.

2. Claims 1-4, 6-10 and 12-16 (ISSUE #7)

a. Independent Claim 1

(i) Kiuchi Discloses "a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client"

The Examiner correctly determined that Kiuchi discloses "a domain name server (DNS) proxy module that intercepts DNS requests sent by a client," because Kiuchi discloses:

- The client-side proxy is a DNS proxy module that causes a host name to be converted into an IP address: "A client-side proxy asks the C-HTTP name server whether it can communicate with the host specified in a given URL. ... If the connection is permitted, the C-HTTP name server sends the IP address ... of the server-side proxy. ... If the client-side proxy receives an error status, then it performs DNS lookup, behaving like an ordinary HTTP/1.0 proxy." (Kiuchi, pg. 65)
- When a client sends a connection request, the client-side proxy intercepts the request: "When one of these resource names with a connection ID, for example, 'http://server.in.current.connection/sample.html=@=6zaDfldfcZLj8V!i' in Figure (b), is selected and requested by an end-user, the client-side proxy takes off the connection ID and forwards the stripped, the original resource name to the server in its request." (Kiuchi, pg. 65)

Accordingly, Kiuchi teaches a client-side proxy that (i) receives the name of a host in a URL; (ii) forwards the name of the host to the C-HTTP name server; (iii) receives either an IP address for the host name or an error status; and (iv) if the client-side receives an error status, then the client-side proxy performs a DNS lookup, behaving like an ordinary HTTP/1.0 proxy.

Thus, Kiuchi discloses “a domain name server (DNS) proxy module that intercepts DNS requests sent by a client.”

First, on pages 51-52, Patent Owner argues that Kiuchi “expressly teaches that C-HTTP does not involve DNS.” Patent Owner cites to Kiuchi where “in a C-HTTP-based network, instead of DNS, a C-HTTP-based secure, encrypted name and certification service is used.” Patent Owner’s argument is incorrect. Kiuchi expressly teaches that the client-side proxy module intercepts the name of the host and forwards it to the C-HTTP server, but “if the client-side proxy receives an error status, then it performs DNS lookup, behaving like an ordinary HTTP/1.0 proxy.” (Kiuchi, pg. 65). Accordingly, Kiuchi clearly and unequivocally states that the client-side proxy “performs DNS lookup, behaving like an ordinary HTTP/1.0 proxy.” Thus, Kiuchi teaches “a domain name server (DNS) proxy module that intercepts DNS requests sent by a client.”

Second, on page 52, Patent Owner cross-references to an Office Action in control no. 95/001,679. However, the Patent Owner failed to inform the Examiner that the Patent Office, in pending related cases, has repeatedly found that Kiuchi teaches domain name servers and domain name services:

- In control no. 95/001,856, the Patent Office found that Kiuchi teaches “a domain name service configured for connection a to communication network” (Control No. 95/001,856, Office Action mailed March 5, 2012, pg. 17);
- In control no. 95/001,851, the Patent Office found that Kiuchi teaches “a domain name service configured for connection a to communication network” (Control No. 95/001,856, Office Action mailed March 1, 2012, pg. 17); and
- In control no. 95/001,746, the Patent Office found that “Kiuchi’s C-HTTP name server performs domain name services.” (Control No. 95/001,746, Action Closing Prosecution mailed June 18, pg. 21).

Moreover, Patent Owner has provided no definition, claim interpretation, or substantive analysis to differentiate the recited “DNS request” from the “domain name service” and “performs DNS lookup, behaving like an ordinary HTTP/1.0 proxy” teachings of Kiuchi. Accordingly, Patent Owner only makes general allegations that the claims define a patentable invention, without specifically pointing out how the language of the claims patentably

distinguishes them over the references. Such general allegations are not permitted, and do not overcome the Examiner's rejection. (MPEP § 2666).

(ii) Kiuchi Discloses “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

The Examiner correctly determined that Kiuchi discloses “determining whether the intercepted DNS request corresponds to a secure server,” because Kiuchi discloses:

- The client-side proxy sends information in the DNS request to the C-HTTP server: “A client-side proxy asks the C-HTTP name server whether it can communicate with the host specified in a given URL.” (Kiuchi, pg. 65).
- The client-side proxy receives information about whether the client-side proxy is authorized to make the connection specified in the DNS request: “If the name server confirms that the query is legitimate, it examines whether the requested server-side proxy ... is permitted to accept the connection of the client-side proxy.” (Kiuchi, pg. 65).

Accordingly, Kiuchi discloses that (i) the client-side proxy requests that the C-HTTP name server inform the client-side proxy if the DNS request corresponds to a secure server; and (ii) the target server must be part of the closed network and must be authorized to accept the connection from the client-side proxy. Thus, Kiuchi teaches (i) a secure server (e.g., a server that is part of a closed network and that is authorized to accept a connection from the client-side proxy), and (ii) that the client-side proxy determines if the DNS request corresponds to a secure server by sending a request to the C-HTTP name server and receiving a response. Accordingly, Kiuchi discloses “determining whether the intercepted DNS request corresponds to a secure server.”

Patent Owner argues, on page 52-53, that the Examiner is mixing and matching components of Kiuchi. Patent Owner argues that the C-HTTP name server performs the “determining” step and that Kiuchi's client-side proxy does not perform this step. Patent Owner's argument does not take into account the actual language of the claim.

The claim recites “determining whether the intercepted DNS request corresponds to a secure server.” Specifically, the claim does not recite the manner in which the “determining” is performed, and certainly does not recite “determining, by means other than sending a request to another computer, ...” In Kiuchi, the client-side proxy performs its “determining” step by

sending a request to the C-HTTP name server and awaiting a response. Patent Owner's argument incorrectly focuses on the C-HTTP name server (which receives the request), rather than the actual method and process used by the client-side proxy of Kiuchi to perform the "determining."

Accordingly, because Kiuchi's client-side proxy performs the determining by sending a request to the C-HTTP name server and receiving a response, Kiuchi discloses "determining whether the intercepted DNS request corresponds to a secure server" as recited in the claim.

(iii) Kiuchi Discloses "Automatically Initiating an Encrypted Channel Between the Client and the Secure Server"

Patent Owner argues, on pages 53-54, that "Kiuchi is silent on whether any 'automatic initiating' occurs at all." Patent Owner continues, "Kiuchi's C-HTTP system might indeed require user interaction during the connection establishment."

Once more, Patent Owner falls to take into account the full teachings of Kiuchi. Kiuchi is *not* silent about automatic initiating, and for Patent Owner to say that Kiuchi "might indeed require user interaction" is misleading and disingenuous. Kiuchi *expressly teaches* that the secure C-HTTP communication happens without any end-user involvement⁷:

3) Easy manipulation by end-users

End-users do not have to employ security protection procedures. They do not even have to be conscious of using C-HTTP based communications.

(Kiuchi, pg. 68)

Clearly, Kiuchi teaches that the C-HTTP communications do not require the end-users to provide interaction – Kiuchi even teaches that the end-users are unaware that it is occurring. Patent Owner's arguments are incorrect and wholly inaccurate of the teachings of Kiuchi.

b. Independent Claims 7 and 13

First, Patent Owner argues that claim 7 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Kiuchi teaches all of the limitations of claim 7.

⁷ "Automatically initiating an encrypted channel" has been defined by the Patent Owner to mean "initiating the encrypted channel without involvement of a user." *See Ex. L.*

Second, Patent Owner argues that claim 13 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the same reasons set forth above, Kiuchi teaches all of the limitations of claim 13.

c. Dependent Claims 3, 9 and 15

On pages 55-56 of the Response, Patent Owner argues about the patentability of “when the client is not authorized to access the secure server, returning a host unknown error message to the client.”

Kiuchi discloses (i) the client-side proxy receives a host name from the client (Kiuchi, pg. 65); (ii) the client-side proxy asks the C-HTTP name server if it can communicate with the server that corresponds to the host name (Kiuchi, pg. 65); (iii) if the client is not authorized to communicate with the secure server (i.e., the host name did, in fact, to a secure server, but the client was not permitted), then an error status is sent to the client-side proxy (Kiuchi, pg. 65), (iv) when the client-side proxy receives the error status, the client-side proxy performs an ordinary DNS lookup (behaving like an ordinary HTTP/1.0 proxy) – using the host name that corresponded to a secure server (Kiuchi, pg. 65). The ordinary DNS lookup, behaving like an ordinary HTTP/1.0 proxy, would return a host unknown error message⁸, because the original hostname sent is a nonstandard domain name (e.g., it was a host name that the C-HTTP server would recognize).

d. Dependent Claims 6 and 12

Kiuchi discloses “wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client,” because Kiuchi discloses that “it is impossible to know the IP address of a server-side proxy.” (Kiuchi, pg. 68).

On pages 56-57, Patent Owner argues the “fact that C-HTTP includes its own name service and requires certification to resolve an IP address has nothing whatsoever to do with initiating an encrypted channel that involves avoiding sending the true IP address.” Patent Owner’s argument avoids the teachings of Kiuchi.

⁸ Apple Request, Ex. E2 (Declaration of Mr. Fratto), ¶140: “RFC 1035 describes DNS query formats and response codes. ... Code 3 is used to indicate that the requested host name does not exist. This is typically referred to as ‘host not found.’”

Kiuchi describes that “it is impossible to know the IP address of a server-side proxy” in the context of comparing “proxy-proxy vs. end-end” communications. (Kiuchi, pg. 67-68). Kiuchi describes the advantages of proxy-proxy communication via the C-HTTP name server, and one such advantage is: when creating the secure connections, the IP address of the server-side proxy is not known. (Kiuchi, pg. 67-68). Thus, when initiating proxy-to-proxy communications, by making it “impossible to know the IP address,” Kiuchi discloses “wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.”

e. Dependent Claims 2, 4, 8, 10, 14 and 16

Patent Owner makes no additional arguments with respect to these claims, other than to cross-reference back to claims 1, 7, and 13. For the reasons set for above, claims 1, 7 and 13 are anticipated by Kiuchi. Since Patent Owner makes no additional arguments, the Examiner’s rejections of dependent claims 2, 4, 8, 10, 14 and 16 are proper and should remain.

3. Claims 5 and 11 under 35 U.S.C. § 103(a) Based on Kiuchi in view of Martin (ISSUE #8)

Patent Owner makes no additional arguments with respect to these claims, other than to cross-reference back to claims 1 and 7. For the reasons set for above, claims 1 and 7 are anticipated by Kiuchi. Since Patent Owner makes no additional arguments, the Examiner’s rejections of dependent claims 5 and 11 are proper and should remain.

4. Claims 1-4, 6-10 and 12-16 under 35 U.S.C. § 103(a) Based on Kiuchi in view of Edwards (ISSUE #14)

a. Edwards

Claims 1-4, 6-10 and 12-16 are rejected under 35 U.S.C. § 103(a) based on Kiuchi in view of Edwards.

b. Independent Claim 1

(i) Kiuchi in view of Edwards discloses “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”

In the Office Action, the Patent Office rejected claim 1 based on Kiuchi in view of Edwards. Edwards shows that it was well-known in the art to “intercept” requests being sent to a “name service,” such as the name service of Kiuchi.

The Patent Owner argues, at page 60, that Edwards fails to teach receiving “DNS requests.” Patent Owner’s argument does not address the actual rejection. The actual rejection

relies upon Kiuchi teaching a client-side proxy module that receives DNS requests, with Edwards teaching that it was well-known in the art for an intermediary module (such as the client-side proxy of Kiuchi) to “intercept” requests. Patent Owner “cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references.” (MPEP § 2145 (IV).) Here, the Examiner properly relied on Kiuchi—not Edwards—to teach “DNS requests.” The Patent Owner’s argument is without merit.

(ii) Kiuchi in view of Edwards discloses “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

Patent Owner argues, on page 61, that “any proposed modification of Kiuchi to perform the ‘determining’ step at the alleged DNS proxy module—the client-side proxy—would be improper ...Kiuchi’s C-HTTP name server plays a crucial intermediary role.”

However, Patent Owner’s argument does not address the actual rejection. The rejection is not based on any proposed modification of Kiuchi, where functionality would be moved from the C-HTTP name server to the client-side proxy. As discussed above, Kiuchi’s client-side proxy sends a request to the C-HTTP server and waits for a response to the request before proceeding. Thus, by sending the request, the DNS proxy module of Kiuchi performs the step of “determining whether the intercepted DNS request corresponds to a secure server” all by itself, and no modification is necessary.

Further, Patent Owner discusses Edwards as failing to teach a secure server. Again, Patent Owner’s argument does not address the actual rejection, and instead Patent Owner is trying to improperly rebut “nonobviousness by attacking references individually where the rejections are based on combinations of references.” (MPEP § 2145 (IV).) Edwards shows that it was well-known in the art for an intermediary module (such as the proxy server of Kiuchi) to “intercept” requests. The Examiner properly relied on Kiuchi—not Edwards—to teach receiving “secure server.”

(iii) Kiuchi in view of Edwards discloses “Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

First, on page 62, Patent Owner reiterates the argument that C-HTTP connections of Kiuchi are not automatically initiated. As discussed above, this is a blatantly inaccurate statement by the Patent Owner. Kiuchi expressly teaches that the C-HTTP process happens without any end-user involvement. (Kiuchi, pg. 68).

c. Independent Claims 7 and 13

First, Patent Owner argues that claim 7 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Kiuchi teaches all of the limitations of claim 7.

Second, Patent Owner argues that claim 13 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Kiuchi teaches all of the limitations of claim 13.

d. Dependent Claims 3, 9 and 15

On pages 63-64 of the Response, Patent Owner argues about the patentability of “when the client is not authorized to access the secure server, returning a host unknown error message to the client.” As already discussed, Kiuchi discloses (i) the client-side proxy receives a host name from the client (Kiuchi, pg. 65); (ii) the client-side proxy asks the C-HTTP name server if it can communicate with the server that corresponds to the host name (Kiuchi, pg. 65); (iii) if the client is not authorized to communicate with the secure server (i.e., the host name did, in fact, to a secure server, but the client was not permitted), then an error status is sent to the client-side proxy (Kiuchi, pg. 65), (iv) when the client-side proxy receives the error status, the client-side proxy performs an ordinary DNS lookup (behaving like an ordinary HTTP/1.0 proxy) – using the host name that corresponded to a secure server (Kiuchi, pg. 65). The ordinary DNS lookup, behaving like an ordinary HTTP/1.0 proxy, would return a host unknown error message⁹, because the original hostname sent is a nonstandard domain name (e.g., it was a host name that the C-HTTP server would recognize).

e. Dependent Claims 6 and 12

Kiuchi discloses “wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client,” because Kiuchi discloses that “it is impossible to know the IP address of a server-side proxy.” (Kiuchi, pg. 68).

On pages 56-57, Patent Owner argues the “fact that C-HTTP includes its own name service and requires certification to resolve an IP address has nothing whatsoever to do with

⁹ Apple Request, Ex. E2 (Declaration of Mr. Fratto), ¶140: “RFC 1035 describes DNS query formats and response codes. ... Code 3 is used to indicate that the requested host name does not exist. This is typically referred to as ‘host not found.’”

initiating an encrypted channel that involves avoiding sending the true IP address.” Patent Owner’s argument avoids the teachings of Kiuchi.

Kiuchi describes that “it is impossible to know the IP address of a server-side proxy” in the context of comparing “proxy-proxy vs. end-end” communications. (Kiuchi, pg. 67-68). Kiuchi describes the advantages of proxy-proxy communication via the C-HTTP name server, and one such advantage is: when creating the secure connections, the IP address of the server-side proxy is not known. (Kiuchi, pg. 67-68). Thus, when initiating proxy-to-proxy communications, by making it “impossible to know the IP address,” Kiuchi discloses “wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.”

Further, Edwards discusses sending only service interceptors, rather than the object references. Patent Owner argues, on pg. 65, that “Edwards is concerned with avoiding distributing an object of the client to computers outside of the internal network.” Patent Owner misreads the teachings of Edwards. Edwards states “This prevents services in the internal network accidentally subverting security by handing object references to a client in the outside network.” (Edwards, pg. 936). Patent Owner reads this as avoiding sending object references *of* a client to computers outside the network, but that is twisting the words of Edwards. A more straight-forward reading is that Edwards teaches avoiding sending object references (of services) *to* clients in the outside network.

The Patent Owner’s argument is without merit and the Examiner’s rejection should be maintained.

f. Dependent Claims 2, 4, 8, 10, 14 and 16

Patent Owner makes no additional arguments with respect to these claims on pages 65-66, other than to cross-reference back to claims 1, 7, and 13. For the reasons set for above, claims 1, 7 and 13 are anticipated by Kiuchi. Since Patent Owner makes no additional arguments, the Examiner’s rejections of dependent claims 2, 4, 8, 10, 14 and 16 are proper and should remain.

5. Claims 5 and 11 under 35 U.S.C. § 103(a) Based on Kiuchi in view of Edwards and Martin (ISSUE #15)

Patent Owner makes no additional arguments with respect to these claims on page 66, other than to cross-reference back to claims 1 and 7. For the reasons set for above, claims 1 and

7 are anticipated by Kiuchi. Since Patent Owner makes no additional arguments, the Examiner's rejections of dependent claims 5 and 11 are proper and should remain.

C. The Rejections Based on Wesinger Were Proper

1. Claims 1-4, 6-10 and 12-16 based on 35 U.S.C. § 102(e) (ISSUE #9)

a. Wesinger

Claims 1-4, 6-10 and 12-16 were rejected under 35 U.S.C. § 102(e) over Wesinger.

b. Independent Claim 1

(i) Wesinger Discloses “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

The Examiner correctly determined that Wesinger teaches “determining whether the intercepted DNS request corresponds to a secure server” because Wesinger discloses:

- A firewall that handles DNS requests: “Referring more particularly to FIG. 3, there is shown a firewall 305 having ... a DNS/DDNS module 315.” (Wesinger, 10:25-27);
- Requests from a client computer to initiate a connection with a host computer are sent through the DNS of the firewall: “When a client C tries to initiate a connection to host D using the name of D ... The DNS server for D returns the network address of D to a virtual host on the firewall 155. The virtual host returns its network address to the virtual host on the firewall 157 from which it received the lookup request, and so on, until a virtual host on the firewall 105 returns its network address (instead of the network address of D) to the client C.” (Wesinger, 9:16-25);
- The firewall contains a table to determine whether a particular request should be allowed or disallowed: “Of course, the primary function of a firewall is to selectively allow and disallow communications. Hence, in the course of establishing a connection, each virtual host examines a configuration table to determine, based on the particulars of the requested connection – source, destination, protocol, time-of-day, port number, etc. – whether such a connection will be allowed or disallowed.” (Wesinger, 9:53-60)

- The firewall uses the domain name to determine whether a particular request should be allowed or disallowed: See Fig. 7 (requests to connect to the domain name “MJU.SRMC.COM” are allowed if sent from 192.168.0.9/23 and if RULE1 is met and are denied if sent from 192.168.0.* and the TIME parameter is met).

Accordingly, Wesinger discloses (i) receiving DNS requests at a firewall, (ii) that the firewall determines whether to allow or disallow the connection; (iii) that one of the factors for determining whether to allow or disallow the connection is the destination of the requested connection; and (iv) that domain names (e.g., MJU.SRMC.COM) are used to determine whether a particular requested connection should be denied. Therefore, determining whether request should be allowed, based on the particular domain name of the requested connection, discloses teaches “determining whether the intercepted DNS request corresponds to a secure server” as recited in the claim.

(a) The Security Policy of Wesinger Discloses Determining Whether the Intercepted DNS Request Corresponds to a Secure Server

On pages 68-69, Patent Owner argues that Wesinger applies the access rules database 513 “when a connection request is received.” Patent Owner further argues that Wesinger discusses a DNS request and a connection request separately, and therefore, Wesinger does not disclose determining whether the DNS request is requesting access to a secure web site. Patent Owner is incorrect.

Wesinger specifically teaches that the firewall performs the allow or deny determination *as part of the DNS request process*. Fig. 7 is an example portion of the “access rules database” that “govern[s] access to and through the virtual host, i.e., which connections will be allowed and which connections will be denied” (Wesinger, 15:19-28). Notably, the access rules database of Fig. 7 is structured based on the domain name of the different hosts (e.g., “WWW.SRMC.COM”, “WWW.HONOLULU.NET”, “WWW.SANJOSE.NET”, “MJU.SRMC.COM”, etc.). Therefore, Wesinger teaches that the firewall (which includes the DNS server) determines whether a particular connection request should be allowed or denied based on the domain name of the host and the DNS entries.

Wesinger further teaches that “access scrutiny may be applied based on DNS entries” (Wesinger, 15:56-57, emphasis added). Accordingly, not only does Wesinger teach analyzing

the domain name of the requested host for security purposes, Wesinger also specifically teaches that the allow or deny determination are based on the DNS entries themselves.

Therefore, Wesinger teaches that (i) the firewall contains a DNS server, (ii) the DNS requests are sent to the firewall, (iii) the firewall performs the allow or deny determination with respect to the domain names (e.g., “MJU.SRMC.COM”), and (iv) access may be restricted based on DNS entries. Patent Owner’s argument that the DNS request is handled separately from the allow or deny determination disregards the specific teachings of Wesinger. Wesinger discloses “determining whether the intercepted DNS request corresponds to a secure server” as recited in the claim. The Examiner’s rejection was proper.

(b) The security policy of Wesinger discloses determining whether a request corresponds to a secure server

The Examiner correctly determined that Wesinger teaches determining whether the DNS request corresponds to a secure server, because Wesinger discloses:

- A firewall that handles DNS requests: “Referring more particularly to FIG. 3, there is shown a firewall 305 having ... a DNS/DDNS module 315.” (Wesinger, 10:25-27);
- Requests from a client computer to initiate a connection with a host computer are sent through the DNS of the firewall: “When a client C tries to initiate a connection to host D using the name of D ... The DNS server for D returns the network address of D to a virtual host on the firewall 155. The virtual host returns its network address to the virtual host on the firewall 157 from which it received the lookup request, and so on, until a virtual host on the firewall 105 returns its network address (instead of the network address of D) to the client C.” (Wesinger, 9:16-25);
- Referring to a table of web sites to determine if a request to access to a site should be allowed or denied: “An example portion of a master configuration file is shown in Fig. 7 ... Also as part of the configuration file of each virtual host, an access rules database is provided governing access to and through the virtual host, i.e., which connections will be allowed and which connections will be denied.” (Wesinger, 15:19-27).

- The determination of whether a request should be allowed or disallowed can be based on the particular destination (e.g., web site): “Of course, the primary function of a firewall is to selectively allow and disallow communications. Hence, in the course of establishing a connection, each virtual host examines a configuration table to determine, based on the particulars of the requested connection – source, destination, protocol, time-of-day, port number, etc. – whether such a connection will be allowed or disallowed.” (Wesinger, 9:53-60, emphasis added)

Accordingly, Wesinger discloses (i) receiving DNS requests at a firewall, (ii) the firewall determines whether to allow or disallow the connection; and (iii) that one of the factors for determining whether to allow or disallow the connection is the destination of the requested connection. Thus, Wesinger discloses determining whether the DNS request corresponds to a secure server.

On pages 69-73, Patent Owner argues Wesinger fails to disclose requesting access to a secure server. Patent Owner includes numerous references to the declaration of Patent Owner’s expert and citations to the teachings of Wesinger. In particular, Patent Owner focuses on Fig. 7 of Wesinger, which discloses a table to the firewall/DNS that contains a list of websites and whether access is permitted or denied with respect to such websites. Patent Owner argues that the access rules exemplified in Fig. 7 determine whether the remote host requesting a connection is a secure client. Patent Owner’s argument is nonsensical.

For example, Fig. 7 of Wesinger shows requests to connect to the domain name “MJU.SRMC.COM” are (i) allowed if sent from 192.168.0.9/23 and if RULE1 is met and (ii) denied if sent from 192.168.0.* and the TIME parameter is met. In other words, Wesinger teaches, in order to determine if that particular request is allowed or denied, first examine the domain name of the target server (e.g., “MJU.SRMC.COM”) to what security rules apply. The claim limitation in question is “determining whether the intercepted DNS request corresponds to a secure server.” Wesinger specifically shows that firewall determines whether “MJU.SRMC.COM” is a secure server (e.g., certain connections are allowed to reach the server and certain connections are not allowed to reach the server). Therefore, Wesinger discloses determining whether the DNS request corresponds to a secure server.

The Examiner’s rejection was proper.

(ii) **Wesinger Discloses “When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer”**

On page 73, Patent Owner repeats the argument that Wesinger does not determine whether the intercepted DNS request corresponds to a secure server.” As discussed above, Patent Owner is incorrect.

(a) **Reply to Patent Owner’s Argument regarding “Forwarding” and “Determining”**

On page 74, Patent Owner argues that “it is possible for a security policy for a virtual host in Wesinger to exist without having any entries in its Allow and Deny databases.” Thus, Patent Owner concludes, “such a server would be both secure and unsecure at the same time, which is impossible.” Patent Owner’s argument makes no sense.

Wesinger teaches having a security policy. Within that security policy are Allow and Deny. For example, Fig. 7 of Wesinger shows requests to connect to the domain name “MJU.SRMC.COM” are (i) allowed if sent from 192.168.0.9/23 and if RULE1 is met and (ii) denied if sent from 192.168.0.* and the TIME parameter is met. By virtue of being in the policy, **and** having entries for Allow and Deny, the MJU.SRMC.COM server taught in Wesinger is a secure server as recited in the claim.

(b) **Wesinger discloses determining “When the Intercepted DNS Request Does Not Correspond to a Secure Server”**

On page 75, Patent Owner repeats the previous argument that Wesinger does not perform the “determining” limitation with respect to a *DNS request*. As discussed above, Wesinger teaches that (i) the firewall contains a DNS server, (ii) the DNS requests are sent to the firewall, (iii) the firewall performs the allow or deny determination with respect to the domain names (e.g., “MJU.SRMC.COM”), and (iv) access may be restricted based on DNS entries. Patent Owner’s argument that the DNS request is handled separately from the allow or deny determination disregards the specific teachings of Wesinger.

(c) **Wesinger discloses “Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer”**

On pages 75-77, Patent Owner acknowledges that Wesinger “forwards a DNS query” but argues that the forwarding occurs “not when the firewall finds no entries in the Allow and Deny

databases, and certainly not when an intercepted DNS request does not correspond to a secure server.”

Patent Owner’s argument relies upon the notion that “DNS processing as described in Wesinger is independent of the firewall analyzing the access rules database.” This is incorrect.

As shown in Fig. 7 of Wesinger, the DNS processing is specifically tied to the access rules database. Further, Wesinger teaches “access scrutiny may be applied based on DNS entries.” (Wesinger, 15:56-57).

Since Wesinger discloses access scrutiny based on DNS entries and forwarding DNS entries for resolution to enable a connection to be established, the Examiner’s rejection was proper.

(iii) Wesinger Discloses “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

The Examiner correctly determined that Wesinger teaches “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server” because Wesinger discloses:

- A firewall that handles DNS requests: “Referring more particularly to FIG. 3, there is shown a firewall 305 having ... a DNS/DDNS module 315” (Wesinger, 10:25-27).
- The firewall contains a table to determine whether a particular request should be allowed or disallowed: “Of course, the primary function of a firewall is to selectively allow and disallow communications. Hence, in the course of establishing a connection, each virtual host examines a configuration table to determine, based on the particulars of the requested connection – source, destination, protocol, time-of-day, port number, etc. – whether such a connection will be allowed or disallowed” (Wesinger, 9:53-60).
- Once the determination that access is allowed to the target web site based on the DNS entries, encryption processing can be performed: “If the remote host satisfies the required level of access scrutiny insofar as DNS entries are concerned, the INET wrapper gets the Allow and Deny databases for the virtual host ... If all the rules are satisfied, then the connection is allowed. Once the connection has been allowed, the virtual host process invokes code 818 that performs protocol-based connection

processing and, optionally, code 823 that performs channel processing (encryption, decryption, compression, decompression, etc.)” (Wesinger, 16:57 – 17:5).

- The encryption processing creates a virtual private network: “Combining encryption capabilities with programmable transparency as described above allows for the creation of virtual private networks-networks in which two remote machines communicate securely through cyberspace in the same manner as if the machines were on the same local area network.” (Wesinger, 12:23-27).
- The virtual private network is performed automatically: “The DNS tables of each of the firewalls may then be programmed so as to enable such a connection to be established transparently, without the user so much as being aware of any of the firewalls.” (Wesinger, 8:65-9:2).

Accordingly, Wesinger discloses (i) a firewall that receives DNS requests, (ii) that the firewall uses an internal table to determine if a DNS request is requesting access to a secure web site, (iii) if all the access rules are satisfied for the secure web site, performing encryption processing, (iv) the encryption processing enables the creation of virtual private networks; and (v) the creation of the virtual private networks is performed automatically without the user being aware. Thus, Wesinger discloses “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.”

On pages 78-79, Patent Owner makes several arguments in an attempt to distinguish over Wesinger. First, on page 78, Patent Owner argues that the Office Action did not show that the automatic initiation of the encrypted channel is in response to the determining that the DNS request corresponds to a secure server. Patent Owner is incorrect.

Wesinger discloses that encryption is used to create virtual private networks (Wesinger, 12:23-27). Wesinger also discloses that the decision whether to use the encryption (and thus, whether to create a VPN) is based on the configuration file on the firewalls. (Wesinger, 12:13-15). Accordingly, Wesinger clearly discloses that the decision whether to use the encryption to create a VPN *is based on* the access scrutiny associated with the configuration file stored on the firewalls. Therefore, Wesinger discloses that the initiation of the encrypted is in response to the determining that the DNS request is a request that corresponds to a secure server.

Second, on pages 78-79, Patent Owner makes the argument that, in Wesinger, the DNS request and the connection request are separately handled, and therefore Wesinger does not teach

(a) that the “protocol-based connection processing” or (b) that the “encryption” is initiated in response to determining the DNS request corresponds to a secure server. Patent Owner argues that Wesinger’s creation of the encrypted channel is performed “once the connection has been allowed,” and thus is performed based on a “connection request” (as opposed to a “DNS request”). Based on this, the Patent Owner argues that Wesinger’s creation of the encrypted channel is not based on a DNS request. Patent Owner is incorrect.

As previously discussed, Wesinger teaches that the firewall performs the allow or deny determination as part of the DNS request process, because the firewalls of Wesinger include both the DNS server and the access rules database. (Wesinger, 10:25-27). Further, Wesinger discloses that “access scrutiny may be applied based on DNS entries.” (Wesinger, 15:56-57). And, once all the access rules are satisfied, a connection is allowed and an encrypted VPN is created. (Wesinger, 16:57 – 17:5). Therefore, Wesinger discloses that the encrypted channel is created based on the DNS request.

Accordingly, Wesinger discloses (i) a firewall that receives DNS requests, (ii) that the firewall uses an internal table to determine if a DNS request is requesting access to a secure web site, (iii) if all the access rules are satisfied for the secure web site, performing encryption processing, (iv) the encryption processing enables the creation of virtual private networks; and (v) the creation of the virtual private networks is performed automatically without the user being aware. Thus, Wesinger discloses “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.”

The Examiner’s rejection was proper.

c. Independent Claims 7 and 13

First, Patent Owner argues on page 80 that claim 7 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Wesinger teaches all of the limitations of claim 7.

Second, Patent Owner argues that claim 13 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Wesinger teaches all of the limitations of claim 13.

d. Dependent Claims 2, 8 and 14

Patent Owner argues on pages 80-81 that claims 2, 8 and 14 recite the features of claims 1, 7, and 13, respectively and should therefore be allowable. For the same reasons set forth above, Wesinger teaches all of the limitations of claims 1, 7 and 13, and the Examiner’s rejection should be maintained.

(i) Reply to Patent Owner’s Argument That The Examiner Was Inconsistent

Patent Owner’s argument is frivolous. Wesinger describes (i) access scrutiny to determine if a particular server is secure and (ii) access scrutiny to determine if the client is authorized to access the server.

A portion of Fig. 7 of Wesinger is reproduced below:

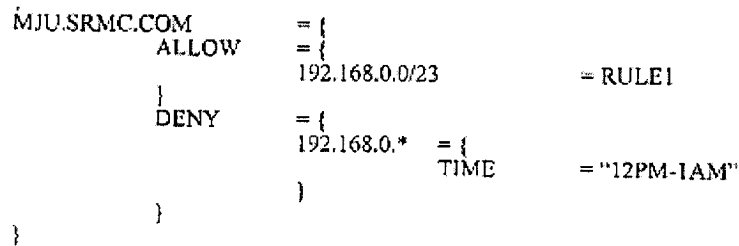


FIG. 7

As Fig. 7 shows, requests to connect to the domain name “MJU.SRMC.COM” are allowed if sent from 192.168.0.9/23 and if RULE1 is met and are denied if sent from 192.168.0.* and the TIME parameter is met. (Wesinger, 15:19 –16:2). Thus, Wesinger describes access scrutiny that (i) determines if the server is secure (e.g., there is a DENY parameter), and (ii) determines if a particular client is authorized to access the server (e.g., DENY any client from 192.168.0.* between 12pm and 1am).

(ii) Wesinger Discloses “Sending a Request to the Secure Server to Establish an Encrypted Channel When the Client is Authorized to Access the Secure Server”

First, on page 82, Patent Owner argues that nothing in Wesinger indicates that the issuing a request from the first computer to a second computer for a connection depends upon the client being authorized to access the secure server. Patent Owner’s argument is not credible.

Wesinger specifically teaches that the connection requests (which go from the first computer to the second computer) depend on authorization and access level. Each firewall perform access scrutiny on the request, which, as discussed above, includes (i) determining if the

server is secure and (ii) determining if a particular client is authorized to access the server. (*See* Wesinger, Fig. 7 and 15:19 – 16:2). The Examiner’s rejection was proper.

Second, on pages 82-83, Patent Owner incorrectly draws the conclusion that Wesinger teaches a connection request that is distinct from a DNS request. This purported distinction argued by the Patent Owner is incorrect. Wesinger teaches that the firewall (which includes the DNS server) determines whether a particular connection request should be allowed or denied based on the domain name of the host and the DNS entries. (*See*, Wesinger, Fig. 7 and 15:56-57). Accordingly, Wesinger teaches the connection request of Wesinger is a DNS request as recited in the claim.

e. Dependent Claims 3, 9 and 15

On pages 83-84 of the Response, Patent Owner argues about the patentability of “when the client is not authorized to access the secure server, returning a host unknown error message to the client.”

Accordingly, even though Wesinger teaches DNS handling, graphical user interfaces, and denying connections, plus the entirety of Wesinger is about facilitating communications and delivering messages, Patent Owner nevertheless contends that Wesinger does not teach returning an error.

Explicit disclosures are not required and the prior art is not to be considered in a vacuum but, “together with the knowledge of one of ordinary skill in the pertinent art ... at the time the ... patent was filed.” *In re Paulson*, 30 F.3d 1475, 1480. *See also, In re Baxter Travenol Labs.*, 952 F.2d 388. It was well-known to one of ordinary skill in the art to return errors when undesired operation occurs.¹⁰ Accordingly, because it was known in the art to return errors, a person of ordinary skill in the art, using the technical knowledge available at the time of the invention, would have known that Wesinger teaches returning an error.

Second, Patent Owner repeats the argument that the request of Wesinger is not a DNS request. As already discussed, Wesinger teaches that the firewall (which includes the DNS server) determines whether a particular connection request should be allowed or denied based on the domain name of the host and DNS entries. (*See*, Wesinger, Fig. 7 and 15:56-57). Accordingly, Wesinger teaches the request of Wesinger is a DNS request, and further, that the

¹⁰ *See* RFC 1035 at 27.

system of Wesinger determines if the DNS request is not requesting access to a secure target web site.

The Examiner's rejection was proper.

f. Dependent Claims 6 and 12

On page 84, Patent Owner repeats the argument that the request of Wesinger initiates encryption "upon processing a connection request and allowing the connection, not in relation to DNS." Again, this purported distinction between a connection request and DNS request argued by the Patent Owner is incorrect. Wesinger teaches that the firewall performs its actions (which includes returning the address of a virtual host rather than the secure server) based on the domain name of the host and the DNS entries. (See, Wesinger, Fig. 7 and 15:56-57).

Accordingly, Wesinger teaches the request of Wesinger is a DNS request, and further, that the system of Wesinger "avoids sending a true IP address of the secure server to the client" as recited in the claims. The Examiner's rejection was proper.

g. Dependent Claims 4, 10 and 16

Patent Owner makes no additional arguments with respect to these claims on pages 84-85, other than to cross-reference back to claims 1-3, 7-9, and 13-15. For the reasons set for above, claims 1-3, 7-9 and 13-15 are anticipated by Wesinger. Since Patent Owner makes not additional arguments, the Examiner's rejections of dependent claims 4, 10 and 16 are proper and should remain.

2. Claims 5 and 11 based on 35 U.S.C. § 102(e) over Wesinger in view of Martin (ISSUE #10)

Patent Owner makes no additional arguments with respect to these claims on page 85, other than to cross-reference back to claims 1 and 7. For the reasons set for above, claims 1 and 7 are anticipated by Wesinger. Since Patent Owner makes no additional arguments, the Examiner's rejections of dependent claims 5 and 11 are proper and should remain.

3. Claims 1-4, 6-10, and 12-16 based on 35 U.S.C. § 103 over Wesinger in view of Edwards (ISSUE #16)

(i) Independent Claim 1

(a) Wesinger in view of Edwards discloses "Determining Whether the Intercepted DNS Request Corresponds to a Secure Server"

First, Patent Owner reiterates on pages 85-87 the argument that Wesinger does not disclose determining whether the intercepted DNS request corresponds to a secure server. As discussed above, Wesinger does, in fact, anticipate such determining as recited in the claim.

Second, Patent Owner discusses Edwards as failing to teach a secure server. Again, Patent Owner's argument does not address the actual rejection. The actual rejection relies upon Wesinger teaching determining whether a DNS request corresponds to a secure server, with Edwards teaching that it was well-known in the art for an intermediary module (such as the firewall/DNS server of Wesinger) to "intercept" requests. Patent Owner "cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references." (MPEP § 2145 (IV)). In the present rejection, the Examiner properly relied on Wesinger—not Edwards—to teach determining whether a DNS request corresponds to a secure server.

(b) Wesinger in view of Edwards discloses "When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer"

First, on page 87, Patent Owner reiterates the argument that Wesinger does not disclose determining whether the intercepted DNS request corresponds to a secure server. As discussed above, Wesinger does, in fact, anticipate such determining as recited in the claim.

Second, on pages 87-88, Patent Owner discusses Edwards as receiving a name service request, not intercepting a DNS request. Patent Owner's argument does not address the actual rejection. The actual rejection relies upon Wesinger teaching determining whether the DNS request corresponds to a secure server, with Edwards teaching that it was well-known in the art for an intermediary module (such as the firewall/DNS server of Wesinger) to "intercept" requests. Patent Owner "cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references." (MPEP § 2145 (IV)). In the present rejection, the Examiner properly relied on Wesinger—not Edwards—to teach receiving a "DNS request."

(c) Wesinger in view of Edwards discloses "When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server"

First, on page 88, Patent Owner reiterates the argument that Wesinger does not disclose when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer. As discussed above, Wesinger does, in fact, anticipate such determining as recited in the claim.

Second, on pages 89-90, Patent Owner discusses Edwards as performing an authorization check, but not doing so based on a DNS request. Patent Owner's argument does not address the actual rejection. The actual rejection relies upon Wesinger teaching performing an authorization check based on a DNS Request, with Edwards teaching that it was well-known in the art for an intermediary module (such as the firewall/DNS server of Wesinger) to "intercept" requests. Patent Owner "cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references." (MPEP § 2145 (IV)). In the present rejection, the Examiner properly relied on Wesinger—not Edwards—to teach "when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server."

(ii) Independent Claims 7 and 13

First, Patent Owner argues on page 90 that claim 7 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Wesinger in combination with Edwards teaches all of the limitations of claim 7.

Second, Patent Owner argues that claim 13 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Wesinger in combination with Edwards teaches all of the limitations of claim 13.

(iii) Dependent Claims 2, 8 and 14

Patent Owner argues on page 91 that claims 2, 8 and 14 recite the features of claims 1, 7, and 13, respectively and should therefore be allowable. For the same reasons set forth above, Wesinger in combination with Edwards teaches all of the limitations of claims 1, 7 and 13, and the Examiner's rejection should be maintained.

(iv) Dependent Claims 3, 9 and 15

On pages 91-92 of the Response, Patent Owner argues about the patentability of "when the client is not authorized to access the secure server, returning a host unknown error message

to the client.” For the reasons set forth above, Wesinger in combination with Edwards teaches all of the limitations of claims 1, 7 and 13, and the Examiner’s rejection should be maintained.

On pages 91-92, Patent Owner discusses that the “object not found” error message of Edwards is not a “host unknown error” message as recited in the claim. Patent Owner’s argument does not address the actual rejection. The actual rejection relies upon Wesinger teaching performing an authorization check based on a DNS Request and returning error messages, with Edwards teaching that it was well-known in the art for an intermediary module (such as the firewall/DNS server of Wesinger) to “intercept” requests. Patent Owner “cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references.” (MPEP § 2145 (IV)). In the present rejection, the Examiner properly relied on the combination of Wesinger and Edwards—not Edwards alone—to teach “when the client is not authorized to access the secure server, returning a host unknown error message to the client.”

(v) Dependent Claims 6 and 12

On page 93, Patent Owner repeats the argument that Wesinger does not disclose “avoids sending a true IP address of the secure server to the client.” As discussed above, Wesinger returns the address of a virtual host, rather than the address of the secure server. Accordingly, Wesinger teaches that the request of Wesinger is a DNS request, and further, that the system of Wesinger “avoids sending a true IP address of the secure server to the client” as recited in the claims.

On pages 93-94, Patent Owner discusses that Edwards does not make up for the deficiencies in Wesinger. Patent Owner’s argument does not address the actual rejection. The actual rejection relies upon Wesinger teaching avoiding sending a true IP address of the secure server to the client, with Edwards teaching that it was well-known in the art for an intermediary module (such as the firewall/DNS server of Wesinger) to “intercept” requests. Patent Owner “cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references.” (MPEP § 2145 (IV)). In the present rejection, the Examiner properly relied on the combination of Wesinger and Edwards—not Edwards alone—to teach “avoids sending a true IP address of the secure server to the client” as recited in the claims.

(vi) Dependent Claims 4, 10 and 16

Patent Owner makes no additional arguments with respect to these claims on page 94, other than to cross-reference back to claims 1-3, 7-9, and 13-15. For the reasons set for above, claims 1-3, 7-9 and 13-15 are rendered obvious by Wesinger in view of Edwards. Since Patent Owner makes not additional arguments, the Examiner's rejections of dependent claims 4, 10 and 16 are proper and should remain.

(vii) Claims 5 and 11 based on 35 U.S.C. § 103(a) over Wesinger in view of Edwards and Martin (ISSUE #17)

Patent Owner makes no additional arguments with respect to these claims on pages 94-95, other than to cross-reference back to claims 1 and 7. For the reasons set for above, claims 1 and 7 are rendered obvious by Wesinger in view of Edwards. Since Patent Owner makes no additional arguments, the Examiner's rejections of dependent claims 5 and 11 are proper and should remain.

D. The Rejections Based on Blum Were Proper (ISSUE #11)

- 1. Overview of Blum**
- 2. Independent Claim 1**
 - a. Blum Discloses "Determining Whether the Intercepted DNS Request Corresponds to a Secure Server"**

The Examiner correctly determined that Blum discloses "determining whether the intercepted DNS request corresponds to a secure server," because Blum discloses:

- Intercepting DNS requests: "The NSP stub 430 is capable of intercepting DNS requests." (Blum, 6:50-51)
- Determining whether the DNS request corresponds to a computer system directly coupled to the LAN or not coupled to the LAN: "The client system 300 includes layered service providers (LSPs) 335 which operate to intercept communications requests from the TCP/IP-compatible client application 325 which are directed to computer systems other than those directly coupled to the LAN 310." (Blum, 5:23-27)
- Determining whether the DNS request corresponds to a remote server: "After determining that the communications request is directed to a remote server, i.e. a server not on the same LAN as the client application..." (Blum, 3:42-44).

- Determining if there is a filter associated with the connection request: “the transparent proxy application 355 checks to see if there is a protocol filter 520 associated with the native protocol of the connection request or with a port indicated in the connection request.” (Blum, 9:33-35)
- Connections to the remote server have special handling: “If the connection request is directed to a remote server not on the LAN 310, the API tunneling LSP 425 directs the connection request to the well-known private port on which the transparent proxy 355 listens for connection requests.” (Blum, 9:19-23).

Accordingly, Blum discloses that (i) DNS requests are intercepted, (ii) DNS requests are analyzed to determine if (x) the request is for a computer system directly coupled to the LAN, or (y) the request is for a remote server (e.g., a server that requires special handling for communications being sent to it); and (iii) determining if communications to the server should be filtered. Thus, Blum discloses “determining whether the intercepted DNS request corresponds to a secure server” as recited in the claims.

On pages 96-98, Patent Owner argues that Blum “only determines whether the server is remote or local, and then proceeds with processing the requests.” Patent Owner continues, “because Blum never addresses the topic of security in connection with the remote servers, they cannot be the recited secure servers.” However, Patent Owner’s arguments improperly introduce new limitations into the claims, because the claims do not recite any particular type of security required to deem that a particular computer is a “secure server.”

Blum teaches that it determines whether a computer is *not* on the same LAN as the client and thus requires special port-handling. Further, Blum teaches that certain communications should be filtered using a “protocol filter 520”. Thus, Blum discloses “determining whether the intercepted DNS request corresponds to a secure server” as recited in the claims.

b. Blum Discloses “When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer”

On pages 98-99, Patent Owner repeats the argument the Blum does not disclose a secure server.

As discussed above, Blum teaches determining whether a computer is *not* on the same LAN as the client and thus requires special port-handling Further, Blum teaches that certain

communications should be filtered using a “protocol filter 520”. Accordingly, Blum teaches determining when a DNS request does not correspond to a secure server.

c. Blum Discloses “When the Intercepted DNS Request Corresponds to the Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

The Examiner correctly determined that Blum discloses “when the intercepted DNS request corresponds to the secure server, automatically initiating an encrypted channel between the client and the secure server,” because Blum discloses:

- Receiving communications according to the native protocol of the client: “A layered service provider intercepts a communications request from a client application in the native protocol of the communications request.” (Blum, 2: 26-28).
- Automatically converting the native protocol into the secure sockets layer protocol: “After determining that the communications request is directed to a remote server, ... the LSP of the invention packages the communications request To package in this context means to add information to the communications request required to forward the request to the proxy server through the various software layers.” (Blum, 3:42-59). “Commonly used protocols include Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Telnet, and *Secure Sockets Layer (SSL)*, for example.” (Blum, 1:46-48, emphasis added).

Accordingly, Blum (i) receiving requests in a native protocol, and (ii) converting the native protocol into another protocol (and one such other protocol is the secure sockets layer (SSL)). Thus, Blum discloses “when the intercepted DNS request corresponds to the secure server, automatically initiating an encrypted channel between the client and the secure server” as recited in the claims.

First, Patent Owner argues on pages 99-100 that Blum “mentions the existence of Secure Socket Layer (SSL) in a list of common protocols in the related art, but does not disclose that SSL is used for remote socket connections in the disclosed system.” Further, Patent Owner argues that “Blum makes multiple references to establishing socket connections, but never discloses that these connections are encrypted.” Patent Owner is incorrect.

Explicit disclosures are not required and the prior art is not to be considered in a vacuum but, “together with the knowledge of one of ordinary skill in the pertinent art...at the time

the...patent was filed.” *In re Paulson*, 30 F.3d 1475, 1480 (Fed. Cir. 1994). *See also, In re Baxter Travenol Labs*, 952 F.2d 388, 391 (Fed. Cir. 1991). In the present instance, Blum discloses (i) facilitating communications using internet protocols, (ii) forwarding communications to private ports, (iii) providing tunneling services (e.g., “[t]he connections request is then intercepted by the API tunneling LSP”, Blum, 9:14-15) and (iv) that a common protocol for communications is the secure sockets layer (SSL). Accordingly, Blum discloses establishing a secure (e.g., encrypted) channel between the client and the secure server.

Second, Patent Owner argues on page 100 that the Office Action improperly points “to two different connections as allegedly disclosing ‘automatically initiating an encrypted channel.’” As discussed above in connection with Aventail, Patent Owner is attempting to re-write the claims.

The claims simply recite automatically “*initiating*.” Blum shows that, when the intercepted DNS request corresponds to a server that is not on the LAN and that requires special handling, it automatically performs the port-handling and protocol filtering (for example, switching to the Secure Socket Layer). The process for encryption begins with (e.g., is initiated by) the intercepting the DNS request and determining that it corresponds to a secure server. The remainder of the encryption process completes itself, resulting in the encryption between the client application and the destination server.

3. Independent Claims 7 and 13

First, Patent Owner argues on page 101 that claim 7 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Blum teaches all of the limitations of claim 7.

Second, Patent Owner argues that claim 13 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Blum teaches all of the limitations of claim 13.

E. Claims 1-4, 6-10 and 12-16 under 35 U.S.C. § 103(a) Based on Aziz in view of Edwards (ISSUE #12)

- 1. Aziz**
- 2. Independent Claim 1**

a. The Combination of Aziz and Edwards Discloses a “Data Processing Device ... Storing a Domain Name Server (DNS) Proxy Module” that Performs All of the Recited Features

On pages 103-105, Patent Owner argues that Aziz teaches that the “resolver” can only be located at the authorized client and provides citations to Aziz purporting to support its position. Patent Owner is incorrect and fails to take into account the full teachings of Aziz.

First, Aziz teaches that the “resolver is a program that acts as an intermediary between a name server and an application program on a client.” (Aziz, 6:62-63). “The term resolver 225 will be used herein to refer to the full functionality provided by the invention, regardless of how many components are used to implement such functionality, *or where those components may be located.*” (Aziz, 8:8-11). So, despite Aziz specifically teaching that it does not matter where the components to implement the resolver are located, Patent Owner nevertheless argues that the resolver can only be located on the client. Patent Owner’s argument contradicts the specific teachings of Aziz.

Second, Aziz teaches an embodiment where the outside name server 110 performs certain steps. Aziz teaches that the name server 110 checks if an SX record (indicating secure communications are required for a particular host) exists in the DNS entry for the host (e.g., determining whether the DNS request is requesting access to a secure server). Aziz also teaches that the name server 110 sends the SX record to the client, which uses the SX record to initiate encrypted communications (e.g., initiating the VPN). To the extent that Patent Owner argues that sending the SX record is not initiating the VPN, such an argument improperly introduces new limitations into the claim. The claim does not recite that the DNS server has to *complete every step necessary to invoke and implement a VPN*. The claim merely states that the DNS server has to *initiate* the VPN. Aziz’s teaching of a DNS server that sends an SX record that results in the VPN discloses a DNS server that initiates the VPN.

Accordingly, either (or both) of the resolver (which can be located anywhere) and the name server 110 disclose “data processing device ... storing a domain name server (DNS) proxy module,” as recited in the claim.

b. The Combination of Aziz and Edwards Discloses a “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”

Patent Owner argues, on page 105-106, that “Aziz merely discloses receiving, but not intercepting, DNS requests” and that Edwards does not make up this deficiency because Edwards does not disclose intercepting DNS requests.

First, Patent Owner provides no substantive arguments as to why the broadest reasonable interpretation of “intercepting” does not read on receiving DNS requests as taught by Aziz. Once again, Patent Owner makes only general allegations that the claims define a patentable invention, without specifically pointing out how the language of the claims patentably distinguishes them over the references. Such general allegations are not permitted, and do not overcome the Examiner’s rejection. (MPEP § 2666).

Second, Edwards discloses that it was well-known in the art to “intercept” requests being sent to a “name service,” such as the name service of Aziz. The Patent Owner argues, at page 105, that Edwards fails to teach receiving “DNS requests.” Patent Owner’s argument does not address the actual rejection. The actual rejection relies upon Aziz teaching receiving a DNS request, with Edwards teaching that it was well-known in the art for an intermediary module (such as the resolver of Aziz) to “intercept” requests. Patent Owner “cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references.” (MPEP § 2145 (IV).) Here, the Examiner properly relied on Aziz—not Edwards—to teach receiving “DNS requests.” The Patent Owner’s argument is without merit.

c. The Combination of Aziz and Edwards Discloses “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

The Examiner correctly determined that Aziz teaches “determining whether the intercepted DNS request corresponds to a secure server” because Aziz discloses:

- Assigning an SX record to a computer that requires secure communications: “[T]he registered name server’s database includes an SX record with an owner name that matches the requested host name.” (Aziz, 10:46-48). “The data field of the SX record contains the identifier (e.g., name or address) of a ‘secure exchanger’ associated with the owner of the record.” (Aziz, 6: 27-29). “The data in the SX record is used by a program called a resolver to update information used by a client for secure communications with protected hosts.” (Aziz, 6:57-60).

- Receiving DNS requests at a name server: “Execution starts at step 305 when outside NS [name server] 120 receives a query for the address of a host (the ‘requested host’) in domain 100.” (Aziz, 9:49-51)
- Determining whether the DNS request is requesting access to a server that has been assigned an SX record: “At step 310, outside NS 120 checks if its zone database has an SX record with an owner name that matches the requested host name... If the database does, at step 315, outside NS 120 adds the SX record identifying the secure exchanger for the requested host to the response.” (Aziz, 9:49-56).

Accordingly, Aziz discloses (i) assigning an SX record to a host name, which indicates whether to use secure communications with that host; (ii) receiving DNS requests; and (iii) determining whether an SX record exists for a host. Thus, receiving a DNS request and then determining whether to use secure communications with the host identified in the DNS request (using the corresponding SX record), discloses “determining whether the intercepted DNS request corresponds to a secure server,” as recited in the claim.

First, on pages 106-107, Patent Owner argues the SX records are just “another type of resource record.” Patent Owner contends that “[j]ust because an SX record may be used for secure communication does not mean that all host names with SX records correspond to secure servers and all host names without SX records do not correspond to secure servers.” Patent Owner is essentially arguing that, even though Aziz teaches that certain hosts use the SX record to indicate that such host is a secure server, not all SX records correspond to secure servers. Patent Owner’s argument is irrelevant.

There is nothing in the claim language that indicates that all “DNS requests” have to return (or not return) particular information. The claim language is simply “determining whether the intercepted DNS request *corresponds* to a secure server.” Aziz specifically teaches that, if a particular host desires to have secure communications, then an SX record having certain security information is assigned to that host name. Then, when a DNS request is received for that host name, the domain name server determines if an SX record exists that corresponds to the host name. Since certain SX records contain information that indicates that the corresponding host desires secure communications, then the DNS server, by determining if an SX record exists, is determining whether the corresponding host desires secure communications. Thus, Aziz

discloses “determining whether the intercepted DNS request corresponds to a secure server,” as recited in the claim.

Second, on page 108, Patent Owner argues that Edwards fails to teach “determining whether the intercepted DNS request corresponds to a secure server.” Patent Owner’s argument does not address the actual rejection. The actual rejection relies upon Aziz teaching determining whether a DNS request corresponds to a secure server, with Edwards teaching that it was well-known in the art for an intermediary module (such as the resolver of Aziz) to “intercept” requests. Patent Owner “cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references.” (MPEP § 2145 (IV).) Here, the Examiner properly relied on Aziz—not Edwards. The Patent Owner’s argument is without merit.

d. The Combination of Aziz and Edwards Discloses a “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

The Examiner correctly determined that Aziz teaches “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” because Aziz discloses:

- Creating a tunnel map entry: “Once resolver 225 receives all these records, execution proceeds at step 430, where resolver 225 creates a tunnel map entry, ... which is used by crypto-processor 230 to encrypt messages to inside host 140.” (Aziz, 11:16-20).
- The client computer encrypts communications to the host computer using the tunnel map entry: “[A]fter creating tunnel map entry 500, resolver 225 returns the address of inside host 140 to application 215 at step 435. If execution ends here, application 215 can now communicate securely with inside host 140 because the tunnel map entry 500 provides all the information that crypto-processor 230 needs to encrypt messages to inside host 140.” (Aziz, 11:54-60).
- The encrypted communications are automatically initiated (i.e., without human intervention): “With the increasing number and mobility of clients, it is burdensome or impossible to keep the outbound secure message information up-to-date by relying on human intervention. ... [T]he problem is solved by enabling authorized clients to dynamically update their outbound secure message information.” (Aziz, 5:39-46).

Accordingly, Aziz discloses (i) creating a tunnel map entry that enables encrypted messages; (ii) the client computer encrypts communications to the host computer using the tunnel map entry; and (iii) the outbound encrypted messages from the client to the host computer are automatically initiated (i.e., without human intervention). Therefore, Aziz teaches “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.”

First, on page 109, Patent Owner argues that the tunnel map entry is merely a listing of fields that are stored at the authorized client and that merely storing resource records in a memory does not initiate an encrypted channel. However, Aziz does not “merely” store resource records. To the contrary, Aziz does much more than store the resource records. For example, once the resource record is stored, that resource record provides *all the information that crypto-processor 230 needs to encrypt messages inside host 140*. (Aziz, 11:58-60). Patent Owner attempts to confuse the issue by focusing on the sentence that says “This completes the execution...” and then arguing that Aziz discloses stopping execution of the process. This is completely inaccurate. Aziz specifically states “If execution ends here, application 215 can now communicate securely with inside host 140 *because the tunnel map entry 500 provides all the information that crypto-processor 230 needs to encrypt messages to inside host 140*.” (Aziz, 13:13-16, emphasis added)

Second, on page 110, Patent Owner argues that Aziz does not teach *automatically* initiating an encrypted channel because Aziz “discloses situations when an encrypted channel may not be automatically initiated.” Patent Owner’s argument is not relevant.

Whether or not Aziz establishes an encrypted channel *every* time is not relevant to whether or not the encryption is *automatically* initiated. When encryption is supposed to be implemented, Aziz teaches that such encryption occurs without any human intervention: “it is burdensome or impossible to keep the outbound secure message information up-to-date by relying on human intervention. ... [T]he problem is solved by ...” (Aziz, 5:39-46).

3. Independent Claims 7 and 13

First, Patent Owner argues on pages 110-11 that claim 7 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Aziz in view of Edwards teaches all of the limitations of claim 7.

Second, Patent Owner argues that claim 13 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Aziz in view of Edwards teaches all of the limitations of claim 13.

4. Dependent Claims 2, 8 and 14

On pages 111-112, Patent Owner argues that Aziz in view of Edwards does not teach “sending a request to the secure server to establish an encrypted channel between the secure server and the client.” Patent Owner focuses on a communication to a second name server, but does not take into account the full teachings of Aziz. Aziz discloses that, when processing a connection to initiate an encrypted channel to inside host 140 (e.g., the secure server), and not having a “record for inside host 140,” the “resolver 225 makes additional queries (not shown in FIG. 4c) as necessary.” (See Aziz 11:63 – 12:33).

5. Dependent Claims 3, 9 and 15

On pages 112-113, Patent Owner argues about the patentability of “when the client is not authorized to access the secure server, returning a host unknown error message to the client.”

Patent Owner contends that returning a particular type of error message makes these claims patentable. Patent Owner makes this argument, despite Aziz teaching that errors have occurred (Aziz, 12:17-22) and Edwards teaching returning an ‘object not found’ error. (Edwards at 933).

Accordingly, it would have been obvious to a person of ordinary skill in the art¹¹ to combine the error detection of Aziz with the returning an error message of Edwards. Thus, Aziz in view of Edwards render obvious “when the client is not authorized to access the secure server, returning a host unknown error message to the client.”

6. Dependent Claims 6 and 12

On pages 114-115, Patent Owner argues that Aziz in view of Edwards does not disclose “automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.” First, Patent Owner argues that Aziz does not disclose not sending a true IP address. Second, Patent Owner argues that Edwards does

¹¹ Patent Owner contends that such a person has a “master’s degree in computer science or engineering, as well as two years of experience in computer networking.” (Dec. of Dr. Keromytis, para. 4).

not disclose automatically initiating the encrypted channel. Patent Owner's arguments address the teachings of Aziz and Edwards *individually*, and do not address the actual rejection. The actual rejection relies upon Aziz in combination with Edwards. Patent Owner "cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references." (MPEP § 2145 (IV).) The Patent Owner's argument against Aziz is without merit.

Aziz discloses automatically establishing a secure channel between a client and a secure server, while the "network topology is hidden" (Aziz 11:64-12:10). Edwards discloses avoiding sending IP addresses because Edwards send service interceptors rather than direct object references. Accordingly, it would have been obvious to a person of ordinary skill in the art¹² to combine automatically establishing a secure channel on a hidden network topology of Aziz with the indirect object references of Edwards. Thus, Aziz in view of Edwards render obvious "automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client."

7. Dependent Claims 4, 10 and 16

Patent Owner makes no additional arguments with respect to these claims, other than to cross-reference back to claims 1-3, 7-9, and 13-15. For the reasons set for above, claims 1-3, 7-9 and 13-15 are rendered obvious by Aziz in view of Edwards. Since Patent Owner makes not additional arguments, the Examiner's rejections of dependent claims 4, 10 and 16 are proper and should remain.

F. Claims 5 and 11 based on 35 U.S.C. § 103 over Aziz in view of Edwards and Martin (ISSUE #13)

Patent Owner makes no additional arguments with respect to these claims, other than to cross-reference back to claims 1 and 7. For the reasons set for above, claims 1 and 7 are obvious over Aziz in view of Edwards. Since Patent Owner makes no additional arguments, the Examiner's rejections of dependent claims 5 and 11 are proper and should remain.

G. The Rejections Under 35 U.S.C. § 102 Based on Aventail Were Proper (ISSUE #1)

¹² Patent Owner contends that such a person has a "master's degree in computer science or engineering, as well as two years of experience in computer networking." (Dec. of Dr. Keromytis, para. 4).

1. Aventail

The Examiner rejected claims 1-16 under § 102(b) based on Aventail.

2. Independent Claim 1

a. Aventail discloses “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

The Examiner correctly determined that Aventail discloses “determining whether the intercepted DNS request corresponds to a secure server,” because Aventail discloses:

- Redirection rules define whether a destination server is a “secure server” by defining whether traffic is redirected, granted or denied to those servers: “Once servers and destinations are defined, you can specify how you want Aventail Connect to redirect (or deny) access to various hosts and services.”

Proxy Redirection	Specify how to redirect traffic.	
	Redirect via	Redirect all traffic through the extranet server selected from the list.
	Do not redirect	Route traffic directly to the specified destination without being redirected through SOCKS.
	Deny service	Deny access to the specified destination. The network connection is blocked locally instead of at the server level.

(Aventail, pg. 38-40).

- Aventail Connect intercepts DNS lookup requests from an application: “The application does a DNS lookup to convert the hostname to an IP address ... [Then] Aventail does the following: If the hostname matches a local domain string or does not match a redirection rule, Aventail Connect passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack performs the lookup.” (Aventail, pg. 11).
- Aventail Connect examines the hostname contained in the DNS lookup to determine if the hostname corresponds to a redirection rule of a destination computer: “If the destination hostname matches a redirection rule domain name (i.e., the host is part of a domain we are proxying traffic to) then Aventail Connect ...” (Aventail, pg. 11).
- Aventail Connect also determines if the username and password should be sent to the destination server: “Aventail Connect supports the Challenge Handshake

Authentication Protocol (CHAP). This authentication method sends your username and password encrypted across the network to the destination server.” (Aventail, pg. 45).

Accordingly, Aventail discloses that (i) certain destination host computers are secure servers (either the destination computer has a corresponding redirection rule governing access to the server or the destination computer requires that the username and password be sent across the network), (ii) Aventail Connect intercepts the DNS lookup request before the TCP/IP stack performs DNS processing; and (iii) Aventail determines if the intercepted DNS lookup request corresponds to a secure server (e.g., a server that has a corresponding redirection rule governing access to the server or requires that the username and password be sent across the network). Thus, Aventail discloses “determining whether the intercepted DNS request corresponds to a secure server” as recited in the claims.

Patent Owner argues, on pages 7-8, that the redirection rules and false DNS entries of Aventail do not disclose any link between the alleged DNS request and whether a server is secure or not. Patent Owner further argues that the Office Action fails to explain why matching a hostname to a redirection rule is the same as determining whether a DNS request corresponds to a secure server. Patent Owner also argues, on page 8, that Aventail does not show any component that corresponds to a “secure server.” Patent Owner’s arguments are an attempt at misdirection and do not address the actual teachings of Aventail.

In Aventail, a hostname is examined to determine if there is an applicable redirection rule associated with the hostname. (Aventail, pg. 11). The redirection rule is used to control access to the destination server. (Aventail, pg. 38). Further, Aventail teaches determining whether certain destinations require a name and password to be sent to the server across the network. (Aventail, pg. 45). Thus, the servers that have a corresponding redirection rule that governs access to the server and servers that require usernames and passwords be sent across the network disclose the “secure servers” recited in the claims. Therefore, determining whether an intercepted DNS request corresponds to a server that has a redirection rule, and determining whether an intercepted DNS request corresponds to a server that requires a name and password, as taught by Aventail discloses “determining whether the intercepted DNS request corresponds to a secure server” as recited in the claim.

b. Aventail Discloses “When the Intercepted DNS Request Corresponds to the Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

The Examiner correctly determined that Aventail discloses “when the intercepted DNS request corresponds to the secure server, automatically initiating an encrypted channel between the client and the secure server,” because Aventail discloses:

- Receiving a DNS request: “The application does a DNS lookup to convert the hostname to an IP address ...” (Aventail, pg. 11)
- Determining if there is a redirection rule that corresponds to the DNS request and whether an authentication module applies for a particular secure server: “If the destination hostname matches a redirection rule domain name...” (Aventail, pg. 11). “Aventail Connect supports the Challenge Handshake Authentication Protocol (CHAP). This authentication method sends your username and password encrypted across the network to the destination server.” (Aventail, pg. 45).
- Automatically initiating an encrypted channel: “If the destination hostname matches a redirection rule domain name ... then Aventail Connect creates a false DNS entry (HOSTENT) that it can recognize during the connection request.” (Aventail, pg. 11-12).
- Once it has been initiated, establishing an encrypted channel between the client and the secure server once the process has been initiated: “Aventail Connects checks the connection request. If the request contains a false DNS entry (from step 1), it will be proxied.” (Aventail, pg. 12) “It then sends the proxy request to the extranet (SOCKS) server” (Aventail, pg. 12). “If an encryption module is enabled and selected by the SOCKS server, Aventail Connect encrypts the data on its way to the server on behalf of the application.” (Aventail, pg. 12). “For this example, the Aventail ExtraNet Server encrypts all sessions with SSL.” (Aventail, pg. 73).

Accordingly, Aventail discloses that (i) the application sends a DNS request, (ii) Aventail Connect determines if the hostname for the destination has a redirection rule (e.g., the destination server is a secure server); (iii) when the DNS request corresponds to a destination with a redirection rule, Aventail Connect initiates the encryption process by creating a false DNS entry;

and (iv) when there is a false DNS entry, Aventail Connect and SOCKS server complete the encryption process by encrypting the data. Thus, Aventail discloses “when the intercepted DNS request corresponds to the secure server, automatically initiating an encrypted channel between the client and the secure server” as recited in the claims.

First, on page 10, Patent Owner repeats the argument that Aventail does not teach whether a particular destination server is determined to be secure or not. As discussed above, access to a particular destination server is granted or denied based on the existence of a redirection rule for such server. Accordingly, determining whether a redirection rule (which grants or denies access to a destination server) corresponds to a particular DNS request discloses that Aventail determines when the DNS request corresponds to a secure server.

Second, on pages 9-11, Patent Owner argues that Aventail does not teach any link between the DNS request and the encryption. Patent Owner argues that Aventail “does not disclose that whether a completed connection is subsequently encrypted has anything to do with a DNS request, let alone automatically initiating an encrypted channel.” Patent Owner is incorrect.

The claims simply recite automatically “*initiating*” an encrypted channel when the DNS request corresponds to the secure server. Aventail teaches a basic sequence of events that result in the establishment of an encrypted channel: (i) the application sends a DNS request, (ii) Aventail intercepts the DNS request; (iii) if the DNS request corresponds to a redirection rule, then create a false DNS entry; (iv) if a false DNS entry was created, then redirect the application’s communications to the proxy server; and (v) the proxy server encrypts all sessions with SSL. (Aventail, pg. 11-12).

Aventail shows that, when the intercepted DNS request corresponds to a secure server (e.g., the destination server has a redirection rule), Aventail Connect automatically creates a false DNS entry: “If the destination hostname matches a redirection rule domain name ... then Aventail Connect creates a false DNS entry (HOSTENT).” (Aventail, pg. 11-12)). Then, because the false DNS entry has been created, the remainder of the encryption process establishes an encryption channel between the client application and the destination server (“If the request contains a false DNS entry (from step 1), it will be proxied.” (Aventail, pg. 12)). Accordingly, because the process to establish the encryption channel only proceeds if the false

DNS entry has been created, the act of creating the false DNS entry initiates the process to establish the encrypted channel.

Therefore, because Aventail Connect (i) automatically creates the false DNS entry when the DNS request corresponds to a secure and (ii) the false DNS entry initiates the process to establish the encryption channel server, Aventail discloses “when the intercepted DNS request corresponds to the secure server, automatically initiating an encrypted channel between the client and the secure server” as recited in the claims.

3. Independent Claims 7 and 13

First, Patent Owner argues on page 11 that claim 7 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Aventail teaches all of the limitations of claim 7.

Second, Patent Owner argues on pages 12-13 that the Examiner did not contemplate all of the language of claim 13, by Patent Owner only cites to the Apple Request. It is disingenuous to say that the Examiner did not take into account all of the claim 13, but only cite to the Apple Request as proof. Contrary to Patent Owner’s assertions, the Examiner fully addressed all of the limitations of claim 13 and did not quote the language of claim 1. Page 15 of the Office Action sets forth the full language of claim 13, and provides citations to the portions of Aventail used to reject the language of claim 13.

4. Dependent Claims 2, 8 and 14

The recited element for claims 2, 8 and 14 is “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.”

Patent Owner’s argues on pages 12-13 that the Examiner “cannot use the cited TCP handshake, which corresponds to a routable IP address, to satisfy one element of a claim, while at the same time point to the separate ‘false DNS entry’ or alleged DNS request to satisfy another element of that claim.” Patent Owner’s argument is unpersuasive. The TCP handshake was used to satisfy one element, and the false DNS entry was used to satisfy another element. Patent Owner has not shown why using two portions of the teaching to show two claim limitations is improper. Patent Owners argument is merely an attempt to distract from the basic teachings of Aventail.

Moreover, it is clear that Aventail discloses “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.” First, Aventail discloses that authentication check occurs (e.g., “determining if the client is authorized”). For example, pages 42-52 of Aventail discuss multiple ways in which the SOCKS proxy server and the destination server interact to manage authentication and to determine that the client is authorized to access the destination server: “This authentication method sends your username and password encrypted across the network the destination server.” Aventail, pg. 45). Second, Aventail discloses that as part of the overall encrypted process initiated by the creation of the DNS entry, a connection request is sent to the destination server: the username is sent “along with your connection request.” (Aventail, pg. 44). Accordingly, Aventail discloses “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.”

5. Dependent Claims 3, 9 and 15

On pages 13-14 of the Response, Patent Owner argues about the patentability of “when the client is not authorized to access the secure server, returning a host unknown error message to the client.”

Patent Owner contends that returning *a particular type of* error message makes these claims patentable. Patent Owner makes this argument, despite:

- Aventail teaching using the standard TCP handshake to connect to a remote host: “The application requests a connection to the remote host. This causes the underlying stack to being the TCP handshake.” (Aventail, pg. 12).
- A declaration from a person of ordinary skill in the art discussing error messages are implemented in TCP communications (Apple Request, Ex. E2, Declaration of Mr. Fratto, ¶140): “RFC 1035 describes DNS query formats and response codes. ... Code 3 is used to indicate that the requested host name does not exist. This is typically referred to as ‘host not found.’”

The Examiner is permitted to use multiple references as the basis for the § 102 rejection, when the extra references are used to prove that the primary reference contains an enabled disclosure or to explain the meaning of a term used in the primary reference. MPEP 2131.01(I) and (II). In the present instance, Aventail discloses using the TCP handshake to handle

connection requests. The Declaration of Mr. Fratto and RFC 1035 are “evidence of what was in the public’s possession” and explain the meaning of terms and phrases used in Aventail. *See*, MPEP 2131.01(I) and (II).¹³

Patent Owner’s focus on inherency (covered under MPEP 2131.01 (III)) is misplaced.

6. Dependent Claims 4, 10 and 16

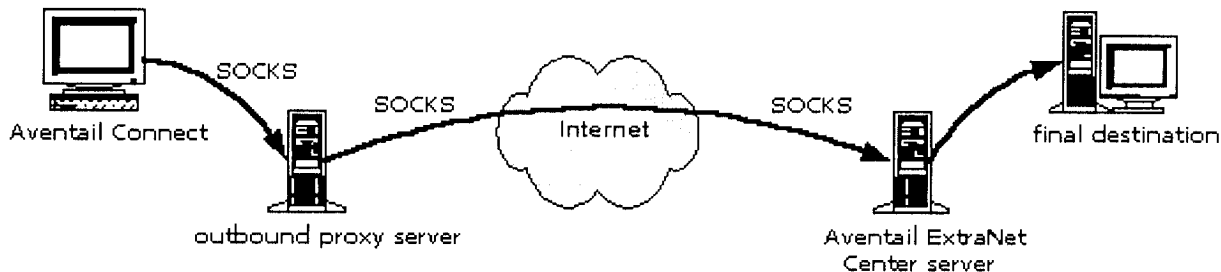
Patent Owner makes no additional arguments with respect to these claims on page 14, other than to cross-reference back to claims 1-3, 7-9, and 13-15. For the reasons set for above, claims 1-3, 7-9 and 13-15 are anticipated by Aventail. Since Patent Owner makes no additional arguments, the Examiner’s rejections of dependent claims 4, 10 and 16 are proper and should remain.

7. Dependent Claims 5 and 11

On page 15, Patent Owner argues that Aventail does not disclose the limitation “establishing an IP address hopping scheme between the client and the secure server” in claims 5 and 11. Patent Owner argues that the multiple firewall traversal of Aventail does not disclose this limitation – but Patent Owner merely makes conclusory statements without any pointing out any substantive distinction between an “IP address hopping scheme” and the multiple firewall traversal taught in Aventail.

Contrary to Patent Owner’s empty assertions, Aventail discloses an IP address hopping scheme and the steps for implementing the scheme. (Aventail, pp. 59-60). Aventail also provides an exemplary diagram, showing communications hopping from the client to the outbound proxy server, and then hopping to the Aventail ExtraNet Center server, before hopping to the final destination:

¹³ There is nothing that prevents the Examiner from issuing a rejection under § 103 that it would be obvious to “return a host unknown error message to the client.” Modifying the words of an error message of Aventail to include the “host not found” error message of RFC 1035 would be an obvious combination under *KSR International v. Teleflex*, 82 USPQ2d 1385, particularly in view of the declaration of Mr. Fratto. *See also* MPEP 2141(III).



Aventail, pg. 60.

Patent Owner also argues that Aventail does not disclose this limitation because Aventail's description of this feature appears "forty or so pages later" in the document. Patent Owner appears to be arguing that Aventail's disclosure is *too* lengthy, comprehensive, and full of detail. Rather than discuss the merits and teaching of the art, Patent Owner would rather discuss the number of pages. The number of pages does not take away from the teachings of Aventail – the Examiner's rejection was proper.

8. Dependent Claims 6 and 12

On pages 15-17, Patent Owner states "The rejection appears to be based on an unsubstantiated presumption, put forth by the Request, that the system in Aventail v3.01 operates in this manner." However, Patent Owner does not provide any argument or substance to support this statement. As such, the Patent Owner makes only a "general allegation that the claims define a patentable invention, without specifically pointing out how the language of the claims patentably distinguishes them over the references." (MPEP § 2666, emphasis added)

On the other hand, the Examiner has specifically shown how Aventail "avoids sending a true IP address of the secure server to the client" as recited in claims 6 and 12:

- "For security reasons the Aventail ExtraNet Server is configured such that operating system routing is disabled. Therefore, no direct network connections between the public LAN and the private LAN can be created without being securely proxied through the Aventail ExtraNet Server." (Aventail, pg. 72)
- "Aventail Connect will forward traffic destined for the private internal network to the Aventail ExtraNet Server." (Aventail, pg. 73)
- "Then, based on the security policy, the Aventail ExtraNet server will proxy mobile user traffic into the private network." (Aventail, pg. 73)

Accordingly, rather than sending the address of the destination server to the client; Aventail proxies and forwards traffic. The Examiner's rejection was proper.

H. The Rejections Under 35 U.S.C. § 102 Based on AutoSOCKS Were Proper (ISSUE #2)

On page 17, Patent Owner argues AutoSOCKS is substantially similar to Aventail, and accordingly, Patent Owner incorporates by reference the arguments made with respect to Aventail. Requester likewise incorporates by reference the response to Patent Owner's arguments.

I. The Rejections Under 35 U.S.C. § 102 Based on BinGO Were Proper (ISSUE #3)

1. The BinGO User's Guide Incorporates the BinGO EFR

Patent Owner argues on pages 17-18 that the BinGO User's Guide does not incorporate the BinGO EFR because in two places, the BinGO User's Guide refers to the "Extended Features Reference." Patent Owner concludes, then, that the BinGO User's Guide must not be referring to the BinGO Extended Feature Reference. Thus, according to the Patent Owner, a person with a "master's degree in computer science or engineering, as well as two years of experience in computer networking"¹⁴ would not understand the documentation because one item says "Extended Features Reference," while the other item says "Extended Feature Reference." Evidently, Patent Owner thinks little of the skills of a person of ordinary skill in the art.

Patent Owner also attempts to argue about "Version 1.2" versus "Version 1.5." However, this is another argument without any substance. The BinGO User's Guide is dated March of 1999.¹⁵ The BinGO EFR (version 1.2) cited by the Examiner was the version in existence in February of 1999. The Examiner used the User's Guide (March 1999) which incorporates BinGO EFR (February 1999). The fact that *other, later* versions exist (e.g., the version 1.5 referred to Patent Owner) is not relevant.

¹⁴ This is the definition of a person of ordinary skill suggested by Patent Owner's expert. (Response, Decl. of Dr. Keromytis, ¶ 4)

¹⁵ This date is evident by following the link supplied in the Apple Request (See Exhibit X7, cover page).

Since, the BinGO User's guide specifically incorporates the Extended Features Reference, and the Examiner's rejection under § 102 was proper.¹⁶

2. The BinGO User's Guide

Claims 1-16 were rejected under § 102(a).

3. Response to Patent Owner's Arguments Regarding "Two Alternative Embodiments"

Patent Owner argues on pages 19-21 about the embodiments discussed in the BinGO User's Guide. However, while Patent Owner cites court cases, Patent Owner provides no argument as why to case law or the embodiments are relevant *with respect to the claims* of the '151 patent. Instead, Patent Owner makes only general allegations that the claims define a patentable invention, without specifically pointing out how the language of the claims patentably distinguishes them over the references. Such general allegations are not permitted, and do not overcome the Examiner's rejection. (MPEP § 2666).

4. Independent Claim 1

a. BinGO Discloses "Determining Whether the Intercepted DNS Request Corresponds to a Secure Server"

The Examiner correctly determined that BinGO discloses "determining whether the intercepted DNS request corresponds to a secure server," because BinGO discloses:

- Intercepting name requests from being sent to a DNS server: "Unlike 'names' such as www.bintec.de, computer names are not known on the Internet. They are only used within a corporate network (domain, working group). The DNS server of the provider is thus usually unable to translate the name. ... In order that such unintentional and useless connections are not established, you must prevent such requests about computers in your partner's network taking place in the first place. ... You still want to have the name *BossPC* translated into its IP address."
(BinGO, pg. 88)

¹⁶ There is nothing that prevents the Examiner from issuing a rejection under § 103 that it would be obvious to combine the BinGO User's Guide and the BinGo Extended Features Reference, given the express teaching in the BinGO User's guide to combine the two references. See *KSR International v. Teleflex*, 82 USPQ2d 1385; MPEP 2141(III).

- Using the BinGO router to handle hostname to IP address translation: “The following options are available for name resolution: ... BinGO! as a DNS proxy server. ... [and] is then also used as an intermediary for DNS requests.” (BinGO, pg. 87)
- The BinGO router determines whether the DNS request is for names known on the Internet or are used only within a corporate network: “Unlike ‘names’ such as www.bintec.de, computer names are not known on the Internet. They are only used within a corporate network (domain, working group).” (BinGO, pg. 88)

Accordingly, BinGO discloses (i) intercepting host name requests being sent to a DNS server; and (ii) determining if the host name request is for the Internet, or if the host name request is for a corporate network (e.g., a secure server). Thus, BinGO, by intercepting host name requests and determining if the host name request is for a (private, non-Internet) corporate network, discloses “determining whether the intercepted DNS request corresponds to a secure server” as recited in the claims.

Patent Owner first argues, on pages 21-22, that BinGO “does not show that the alleged DNS proxy module – the BinGO! router” performs this limitation. This is incorrect. BinGO discloses that the BinGO! router can be a DNS proxy server that receives DNS requests. (BinGO, pg. 87). Further, BinGO shows that the BinGO! router determines whether the DNS requests are for the Internet or for the corporate network. (BinGO, pg. 88). By determining if the DNS requests are for the corporate network, the BinGO! router performs determining whether the intercepted DNS request corresponds to a secure server” as recited in the claims.

Next, Patent Owner argues, on pages 22-23, that BinGO “does not determine whether a DNS request corresponds to a secure server.” Notably, Patent Owner provides no discussion of what is or what is not a secure server. Instead, Patent Owner makes only a general allegation that the claims define a patentable invention, without specifically pointing out how the language of the claims patentably distinguishes them over the references. Such general allegations are not permitted, and do not overcome the Examiner’s rejection. (MPEP § 2666).

On the other hand, the Examiner has properly shown that BinGO specifically teaches that the BinGO! router makes a determination as to whether a DNS request corresponds to a corporate network (e.g., a secure server) versus whether the DNS request corresponds to a site on the Internet (e.g., an unsecure server). (BinGO, pg. 88). Accordingly, BinGO discloses

“determining whether the intercepted DNS request corresponds to a secure server” under the broadest reasonable interpretation. The Examiner’s rejection was proper.

b. BinGO Discloses “When the Intercepted DNS Request Corresponds to the Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

The Examiner correctly determined that BinGO discloses “when the intercepted DNS request corresponds to the secure server, automatically initiating an encrypted channel between the client and the secure server,” because BinGO discloses:

- Establishing a connection after a DNS request is determined to identify a destination inside a corporate network: “You still want to have the name *BossPC* translated into its IP address.” (BinGO, pg. 88). “Unlike ‘names’ such as www.bintec.de, computer names are not known on the Internet. They are only used within a corporate network (domain, working group). (BinGO, pg. 88). “By connecting to your company’s head office from your home or branch office, you can conveniently access any information you may need from the headquarters.” (BinGO, pp. 15-16).
- The BinGO! router uses virtual private networking: “The Setup Tool is a menu-driven tool for the configuration and administration of BinGO!” (BinGO, pg. 105)

VPN	Here the necessary settings for Virtual Private Networking (VPN) are made. It only appears if you have entered the relevant valid license. To use the function, you need a VPN server from Security Dynamics. The license can be optionally acquired. You will find more detailed explanations and instructions in Extended Features Reference.

- The VPN connections of the BinGO! router are encrypted: “Since these VPN connections are encrypted (user data portion) network administrators can be assured that the use of the underlying public data network does not compromise data integrity.” (EFR, pg. 82)

Accordingly, BinGO discloses (i) intercepting host name requests being sent to a DNS server; (ii) determining if the host name request is for the Internet, or if the host name request is for a corporate network (e.g., a secure server); and (iii) when the host name request corresponds to a destination in the corporate network, creating an encrypted VPN connection. Thus, BinGO

discloses “when the intercepted DNS request corresponds to the secure server, automatically initiating an encrypted channel between the client and the secure server” as recited in the claims.

First, Patent Owner alleges on page 25 that the Apple Request does not address the correct claim language. Patent Owner’s argument is without merit. The Office Action issued by the Examiner properly addresses *each and every* limitation of claim by incorporating by reference Exhibit C3 of the Apple Request. Patent Owner’s attempt to focus on a sentence within the Apple Request is misleading when the Office Action itself addresses every limitation.

Second, Patent Owner argues on page 25 that the encryption feature of BinGO “may come into effect based on entirely different criteria than those recited in claim 1.” Patent Owner then discusses different examples of when the encryption feature of BinGO might be initiated. However, it is not relevant that Patent Owner’s expert has the ability to hypothesize potential scenarios. What *is* relevant is that BinGO teaches (i) when the DNS request is for the corporate network, (ii) an encrypted VPN is automatically initiated.

Third, Patent Owner argues on page 26 that the BinGO EFR describes features applicable to the BRICK product, and therefore, do not describe the features applicable to the BinGO! router. Patent Owner’s argument is incorrect – BinGO EFR describes features applicable to “BIANCA/BRICK **and** BinGO! routers” (EFR, pg. 2, emphasis added).

Fourth, Patent Owner argues on page 26-28 that, because there is a connection to an ISP, the Office Action is contradicting its position with respect to a secure server. This is not correct. There is a distinction between routing of the DNS request (and the determination of whether the request is for a corporate network, e.g., a secure server), as discussed above, and the physical network over which the DNS requests occur, which is what is shown in the figures highlighted by Patent Owner. BinGO EFR highlights this distinction: “The same client then initiates a second, logical connection, to the VPN Server.” (EFR, pg. 83). Once the VPN is established, “the ISP is unaware of its participation in the VPN.” (EFR, pg. 84).

Fifth, Patent Owner argues on page 28, that the BinGo! router does not “automatically initiate.” Patent Owner alleges that BinGO has a “manual” authentication process, citing to BinGO’s description of how an administrator would login to BinGO. Patent Owner focuses on a small portion of BinGO, when the rest of its disclosure shows that Patent Owner is wrong. For example, “a significant advantage of your BinGO! is the means by which access to networks is achieved. When using a modem/ISDN-card, you must expressly dial your Internet provider ...

On the other hand ... BinGO! realizes that the requested address lies outside your own LAN, thus **automatically** establishes a connection with your provider and the Internet. ... The same principle is applicable for conveniently accessing data from your home office.” (BinGO, pg. 17, emphasis added). Accordingly, the essential purpose of BinGO’s operation is that its connectivity features (DNS routing, authentication, and encryption) happen automatically.

5. Independent Claims 7 and 13

First, Patent Owner argues on page 29 that claim recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, BinGO teaches all of the limitations of claim 7.

Second, Patent Owner argues that the Examiner did not contemplate all of the language of claim 13, by Patent Owner only cites to the Apple Request. It is disingenuous to say that the Examiner did not take into account all of the claim 13, but only cite to the Apple Request as proof. Contrary to Patent Owner’s assertions, the Examiner fully addressed all of the limitations of claim 13 in the Office Action and did not quote the language of claim 1. Page 28 of the Office Action incorporates by reference Ex. C3 of the Apple Request, which sets forth each and every limitation of claim 13 (see, e.g., pg. 36 of Ex. C3), and provides citations to the portions of BinGO used to reject the language of claim 13.

6. Dependent Claims 2, 8 and 14

The Examiner correctly determined that BinGO discloses “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client,” because BinGO discloses:

- Using authentication procedures to check for authorization at the secure server: “PAP, CHAP and MS-CHAP are the common procedures used for authentication of PPP connections. These use a standard procedure to exchange a user ID and a password for checking the identity of the far end.” (BinGO, pg. 242).
- The VPN link is established by a sending a request: The VPN “is established on demand by software that establishes a link between a client and the server.” (EFR, pg. 82, emphasis added).
- The request for the VPN link occurs when the client is authorized: “the VPN server will typically want to verify the initiating partner during connection

establishment. Authentication is performed inband using PAP, CHAP, or MS-CHAP. (EFR, pg. 84).

Accordingly, BinGO discloses (i) checking, during establishing of the connection, if the client is authorized using authentication protocols; and (ii) establishing a VPN on demand (e.g., a request) between the client and the server.

Patent Owner argues, on page 30, that BinGO does not disclose “any request relating to any encryption.” This is not correct. As discussed, BinGO discloses that the VPN is established “on demand.” (EFR, pg. 82).

Patent Owner then argues, on pages 30-31, that BinGO and BinGO EFR do not inherently disclose “sending a request to the secure server to establish an encrypted channel.” This argument is irrelevant. BinGO EFR expressly discloses that the VPN (which is encrypted) is established “on demand.” (EFR, pg. 82).

7. Dependent Claims 3, 9 and 15

On pages 31-32 of the Response, Patent Owner argues about the patentability of “when the client is not authorized to access the secure server, returning a host unknown error message to the client.”

Patent Owner contends that returning *a particular type of* error message makes these claims patentable. Patent Owner makes this argument, despite:

- BinGO teaching using conventional DNS procedures (BinGO, pg. 358).
- BinGO teaching using well-known and standardized industry protocols governing communications (e.g., PPTP (RFC 1171) and DNS handling and resolution (RFC1034, 1035)).
- A declaration from a person of ordinary skill in the art discussing error messages are implemented in TCP communications (Apple Request Ex. E2, Declaration of Mr. Fratto, ¶140): “RFC 1035 describes DNS query formats and response codes. ... Code 3 is used to indicate that the requested host name does not exist. This is typically referred to as ‘host not found.’”

The Examiner is permitted to use multiple references as the basis for the § 102 rejection, when the extra references are used to prove that the primary reference contains an enabled disclosure or to explain the meaning of a term used in the primary reference. MPEP 2131.01(I) and (II). In the present instance, BinGO discloses uses standardized industry protocols to handle

connection requests. The Declaration of Mr. Fratto and RFC 1035 are “evidence of what was in the public’s possession” and explain the meaning of terms and phrases used in BinGO. *See*, MPEP 2131.01(I) and (II).¹⁷

Patent Owner’s focus on inherency (covered under MPEP 2131.01 (III)) is misplaced.

8. Dependent Claims 4, 10 and 16

Patent Owner makes no additional arguments with respect to these claims on page 32, other than to cross-reference back to claims 1-3, 7-9, and 13-15. For the reasons set for above, claims 1-3, 7-9 and 13-15 are anticipated by Aventail. Since Patent Owner makes not additional arguments, the Examiner’s rejections of dependent claims 4, 10 and 16 are proper and should remain.

9. Dependent Claims 5 and 11

First, on pages 32-33, Patent Owner argues that BinGO does not disclose the limitation “establishing an IP address hopping scheme between the client and the secure server” in claims 5 and 11. Patent Owner argues that NAT protocol and the “Open Shortest Path First” features of BinGO do not disclose this limitation – but Patent Owner merely provides conclusory statements without any pointing out any substantive distinction between an “IP address hopping scheme” and the teachings of BinGO.

For example, BinGO specifically teaches using the “Open Shortest Path First” (OSPF) routing protocol, and describes how the OSPF protocol is used because it has “[n]o hop-count limitations” whereas the RIP protocol has a 15 hop limit. Further, in discussing how the OSPF protocol works, BinGO provides an example on page 18 of EFR, describing the number of hops that a particular packet will travel (“the best route for a packet travelling from A to C is ABEFC ... this route requires 4 hops”).

Second, Patent Owner argues that these teachings of an IP address hopping scheme contained in EFR apply only to the BRICK router and not the BinGO! routers. Again, the EFR

¹⁷ There is nothing that prevents the Examiner from issuing a rejection under § 103 that it would be obvious to “return a host unknown error message to the client.” Modifying the words of an error message of BinGO to include the “host not found” error message of RFC 1035 would be an obvious combination under *KSR International v. Teleflex*, 82 USPQ2d 1385, particularly in view of the declaration of Mr. Fratto. *See also* MPEP 2141(III).

applies to “features available on BIANCA/BRICK and BinGO! routers” (EFR, pg. 2, emphasis added).

10. Dependent Claims 6 and 12

First, on pages 33-34, Patent Owner repeats previous arguments that already have been fully addressed above. Patent Owner provides an additional argument on page 35 that “mere use of encryption does not necessarily require avoiding ‘sending a true IP address of the secure server to the client.’” Once again, Patent Owner focuses on one element, while avoiding the relevant teachings of the reference. Instead, the proper focus (and the one used by Examiner) is how the BinGO! router serves as a *proxy* for the client and handle the routing of communications to and from the corporate network (e.g., the “secure server”). Since the BinGO! router is the proxy serving as the intermediary for the inbound and outbound traffic, this avoids sending the true IP address of the server to the client – only the proxy needs to have the true IP addresses.

J. The Rejections Under 35 U.S.C. § 103 Based on Beser in View of Kent Were Proper (ISSUE #4)

1. Beser in View of Kent

Claims 1, 2, 4-8, 10-14 and 16 were rejected under 35 U.S.C. § 103 over Beser in view of Kent.

2. Beser and Kent Were Properly Combined

On page 36, Patent Owner argues that Beser and Kent cannot be combined because Beser teaches away from using IPsec and other encryption techniques, while Kent proposes IPsec. However, Patent Owner’s interpretation of Beser is incorrect – Beser explains that ordinarily all IP traffic within an IP tunnel will be encrypted, but that in certain high traffic volume implementations, encrypting all IP packets may prove impractical. Beser also points out the importance of assuring the secure and private nature of IP tunnels (Beser, 2:36-40). Accordingly, while Beser discusses the merits of particular implementations, Beser nevertheless discusses combining an IP tunnel with encryption techniques, and further, Beser describes IPsec as a particular encryption technique.

3. Independent Claim 1

a. Beser in view of Kent Render Obvious “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”

The Examiner correctly determined that Beser in view of Kent discloses “a domain name server (DNS) proxy module that intercepts DNS requests sent by a client,” because Beser discloses:

- A third party network device that can be a domain name server: “The trusted-third-party 30 may be a ... a domain name server.” (Beser, 4:9-10).
- The third party network device receives a domain name of a target telephony device: “the request includes a unique identifier for the terminating telephony device... the unique identifier is any of a dial-up number, an electronic mail address, or a domain name.” (Beser, 10:37-41).

Accordingly, Beser discloses a third party network device that (i) can be a domain name server and (ii) receives a domain name for the target device. Therefore, Beser in view of Kent discloses “a domain name server (DNS) proxy module that intercepts DNS requests sent by a client.”

Patent Owner’s arguments on page 38-39 are unpersuasive in view of the teachings of Beser and Kent of a third party network device that receives domain names sent by the client.

b. Beser in view of Kent Render Obvious “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

The Examiner correctly determined that Beser in view of Kent discloses “determining whether the intercepted DNS request corresponds to a secure server,” because Beser discloses evaluating a request, comparing the request to a database of entries, and then taking additional actions to establish the IP tunnel based on the result of that evaluation. (Beser, 11:45-59). Accordingly, the determination of whether a particular server meets a predefined set of criteria as taught in Beser renders obvious “determining whether the intercepted DNS request corresponds to a secure server”

Patent Owner’s arguments on page 39-40 are unpersuasive in view of the teachings of Beser and Kent of a third party network device that determines whether a particular server meets a predefined set of criteria.

c. Beser in view of Kent Render Obvious “When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer”

The Examiner correctly determined that Beser in view of Kent discloses “when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a

DNS function that returns an IP address of a nonsecure computer,” because Beser discloses a DNS request is sent to the trusted-third-party network device that is functioning as a DNS server (Beser, 10:38-41; 11:33-36). If the destination does not cause the trusted-third-party network device to negotiate the establishment of an IP tunnel, the trusted-third-party network device will, by its nature of being a DNS server, simply return the IP address of the (non-secure) domain name.

Patent Owner’s arguments on page 41-42 are unpersuasive in view of the teachings of Beser and Kent of a third party network device that provides DNS functionality.

d. Beser in view of Kent Render Obvious “When the Intercepted DNS Request Corresponds to the Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

The Examiner correctly determined that Beser in view of Kent discloses “when the intercepted DNS request corresponds to the secure server, automatically initiating an encrypted channel between the client and the secure server,” because (i) Beser discloses that the trusted-third-party network device will automatically negotiate with first and second network devices to establish an IP tunnel between the first and second network devices. (Beser, Fig. 4; 11:9-25; 12:6-19) and (ii) Kent discloses encryption and tunneling mechanisms that work automatically (Kent at 13; 29-34).

Patent Owner’s arguments on page 42-43 are unpersuasive in view of the teachings of Beser and Kent of a third party device that automatically negotiates an IP tunnel and that encryption and tunneling mechanisms work automatically.

4. Independent Claims 7 and 13

First, Patent Owner argues on pages 43-44 that claim 7 recites features similar to those described for claim 1, but makes no additional arguments beyond cross-referencing to the arguments of claim 1. For the reasons set forth above, Beser in view of Kent teaches all of the limitations of claim 7.

Second, Patent Owner argues that the Examiner did not contemplate all of the language of claim 13, by Patent Owner only cites to the Apple Request. It is disingenuous to say that the Examiner did not take into account all of the claim 13, but only cite to the Apple Request as proof. Contrary to Patent Owner’s assertions, the Examiner fully addressed all of the limitations of claim 13 and did not quote the language of claim 1. The Office Action incorporates by reference Ex. C4 of the Apple Request, which sets forth each and every limitation of claim 13

(see, e.g., pg. 26 of Ex. C4 of the Apple Request), and provides citations to the portions of Beser and Kent used to reject the language of claim 13.

5. Dependent Claims 2, 8 and 14

The Examiner correctly determined that Beser in view of Kent discloses “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client,” because Beser discloses (i) “the IP 58 packets may require encryption and authentication to ensure that the unique identifier cannot be read on the public network” (Beser 11:22-24); and (ii) the “trusted-third-party network device 30 constructs a second IP 58 packet 194 ... The second IP 58 packet 194 is sent to the second network device 16.” Accordingly, Beser teaches that the packets may be authenticated and that the packets are sent to the secure server as part of establishing a connection.

Patent Owner’s arguments on pages 44-45 are unpersuasive in view of the teachings of Beser and Kent of authenticating packets and send packets to the secure server as part of establishing a connection.

6. Dependent Claims 4, 10 and 16

On pages 45-47, Patent Owner argues about the patentability of “wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.”

Explicit disclosures are not required and the prior art is not to be considered in a vacuum but, “together with the knowledge of one of ordinary skill in the pertinent art ... at the time the ... patent was filed.” *In re Paulson*, 30 F.3d 1475, 1480. *See also, In re Baxter Travenol Labs.*, 952 F.2d 388. It was well-known to one of ordinary skill in the art that a common method of initiating communication with a remote server is a user entering a URL into a web browser.¹⁸

Patent Owner’s arguments are incorrect. Using a web browser to enter URLs was known to a person of ordinary skill in the art at the time of invention.

7. Dependent Claims 5 and 11

The Examiner correctly determined that Beser in view of Kent discloses “establishing an [IP] address hopping scheme between the client and the secure server,” because Beser discloses

¹⁸ A search of the U.S. Patent Office (using the search term “ISD/1/1/1990->10/30/1998 and spec/browser and spec/URL”) located over 160 patents issued prior to the earliest possible priority date of the ‘151 patent that discuss web browsers and URLs for retrieving information.

that the network address translator (NAT) protocol is routinely used in establishing IP tunnels (Beser, 2:18-27).

Patent Owner's arguments (pages 47-48) reciting the potential issues of NAT described by Beser are irrelevant. Whether or not NAT is computationally expensive or causes security problems does not take away from the *actual teaching of Beser* that NAT can be used. Patent Owner has provided no reasoning or justification how or why the address hopping scheme recited in claims 5 and 11 is different or better than the IP address hopping scheme taught by Beser.

8. Dependent Claims 6 and 12

The Examiner correctly determined that Beser in view of Kent discloses "avoids sending a true IP address of the secure server to the client," because Beser discloses that it "hide[s] the source IP address" (Beser, 2: 12-14).

Patent Owner argues on pages 48-49 that the hiding in Beser is different than the claimed invention because the end users do not learn the identity of the terminating end of the tunneling association. Claims 6 and 12 contain no such limitation. Moreover, Beser teaches that one of the problems that Beser solves is that a hacker would still be capable of reading the source address of the packets. To solve this problem, the IP packets "need to be encrypted before the encapsulation in order to hide the source IP address." (Beser, 2: 12-14).

K. Response to Patent Owner's Argument That Secondary Considerations Demonstrate Non-Obviousness

On pages 116-118, Patent Owner argues that secondary considerations rebut any finding of obviousness. To be given substantial weight in determining obviousness or nonobviousness, evidence of secondary considerations must be relevant to the subject matter as claimed, and therefore the Examiner must determine whether there is a nexus between the merits of the claimed invention and the evidence of secondary considerations. MPEP 716.01(b). Further, in the absence of an established nexus with the claimed invention, secondary consideration factors are not entitled to much, if any, weight and generally have no bearing on the legal issue of obviousness. *See In re Vamco Machine & Tool, Inc.*, 752 F.2d 1564, 1577 (Fed. Cir. 1985).

First, Patent Owner has failed to establish any nexus between the '151 patent and the "evidence." Patent Owner points to a declaration by the inventor of the '151 that describes different government funding programs designed to promote science and technology. However,

simply because a government agency funds programs for “Next Generation Internet” and “Dynamic Coalitions” does not establish a nexus between those programs and the *actual claims* of the ‘151 patent. In order for any such evidence to be given weight, if any, the Patent Owner must establish a nexus between the evidence and the claimed invention. Patent Owner has merely listed a number of government-funded programs, with a passing reference to “secure communications.” Patent Owner has not established a nexus between this evidence **and the actual claims of the ‘151 patent.**

Second, Patent Owner argues that the claimed invention has achieved commercial success by noting that several companies have licensed the patent portfolio. However, **a portfolio license does not establish commercial success.** (*Ex parte NTP, Inc., Appeal 2008-004603, slip op. at 132 (BPAI Dec. 22, 2009)*). The Board of Patent Appeals and Interferences has set forth the evidence needed to support the use of a list of licensees as evidence of secondary considerations: (i) testimony from a licensee as to why the licensee took a license; (ii) whether the taking of the license was a business cost-benefit analysis with regarding to defending an infringement suit, as opposed to the actual merits of the invention; (iii) the number of entities who refused to take a license and why; (iv) the terms of the licenses and whether the licenses were favorable to the licensee; (v) market information indicating the number of products that are sold under licenses and the number of products that are not under license; (vi) the structure and operation of the devices made by the licensees to determine if those products embody the reasons as to why the “invention” is advantageous over the prior, if at all; (vii) whether the licensee took the licenses for reasons substantively related to each and every one of the claims of the ‘135 patent; and (viii) a declaration from a representative of any of the licensees attesting to and praising the merits of the claimed invention. (*Ex parte NTP* at 132-134). Patent Owner has not provided any such evidence.

Patent Owner has merely provided a declaration by the inventor that describes government programs and portfolio licenses. Patent Owner has not established any nexus between this “evidence” and the actual claims of the invention. Further, Patent Owner has not provided any of the evidence necessary to establish commercial success.

Accordingly, the evidence of secondary considerations should be afforded no weight. The Examiner’s rejections based on obviousness were proper.

II. Non-Adopted Rejections (ISSUES #4, #5, #6)

The comments by Examiner with respect to Issue #4, Issue #5, and Issue #6 and the non-adopted rejections are noted. Requester reserves any responsive comments for the appeal, if any, in this matter.

III. Conclusion

Therefore, it is requested that claims 1-16 all be finally rejected.

As identified in the attached Certificate of Service and in accordance with MPEP §2266.06 and 37 CFR §§1.248 and 1.903, a copy of the present response, in its entirety, is being served to the address of the attorney/agent of record at the address provided for in 37 CFR 1.33(c). Please direct all correspondence in this matter to the undersigned.

Respectfully submitted,


/David L. McCombs/

David L. McCombs
Registration No. 32,271

Dated: August 17, 2012
HAYNES AND BOONE, LLP
2323 Victory Avenue, Suite 700
Dallas, Texas 75219
Telephone: 214/651-5533
Attorney Docket No.: 43614.99

CERTIFICATE OF SERVICE

I hereby certify that this correspondence, all attachments, and any corresponding filing fee is being transmitted via the Electronic Filing System (EFS) Web with the United States Patent and Trademark Office on August 17, 2012.



Theresa O'Connor



RECEIVED

JUL 20 2012

Reexam

CENTRAL REEXAMINATION UNIT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re <i>Inter Partes</i> Reexaminations of:)	
)	
Edmund Colby Munger et al.)	Control Nos.: 95/001,714; 95/001,697
)	
U.S. Patent No. 7,490,151)	Group Art Unit: 3992
)	
Issued: February 10, 2009)	Examiner: Michael J. Yigdall
)	
For: ESTABLISHMENT OF A SECURE)	Confirmation Nos. 3428, 2161
COMMUNICATION LINK BASED ON A)	
DOMAIN NAME SERVICE (DNS) REQUEST)	

Mail Stop *Inter Partes* Reexam
 Commissioner for Patents
 P.O. Box 1450
 Alexandria, VA 22313-1450

Dear Commissioner:

TRANSMITTAL LETTER

Enclosed please find the following:

1. Patent Owner's Response to Office Action (119 pages);
2. Declaration of Angelos D. Keromytis, Ph.D. (57 pages) with appended *curriculum vitae*;
3. Declaration of Dr. Robert Dunham Short III (6 pages);
4. Appendix - List of Exhibits (1 page);
5. Exhibits Listed on Appendix;
6. Petition Seeking Waiver of 37 C.F.R. § 1.943 for Patent Owner's Response to Office Action of April 20, 2012 (4 pages);
7. Check in the amount of \$400 for the Petition Fee; and
8. Certificate of Service (2 pages).

Please grant any extension of time and charge any additional fees to Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: July 20, 2012

By: /Joseph E. Palys/
Joseph E. Palys
Reg. No. 46,508

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re <i>Inter Partes</i> Reexaminations of:)	
)	
Edmund Colby Munger et al.)	Control Nos.: 95/001,714; 95/001,697
)	
U.S. Patent No. 7,490,151)	Group Art Unit: 3992
)	
Issued: February 10, 2009)	Examiner: Michael J. Yigdall
)	
For: ESTABLISHMENT OF A SECURE)	Confirmation Nos. 3428, 2161
COMMUNICATION LINK BASED ON A)	
DOMAIN NAME SERVICE (DNS) REQUEST)	

Mail Stop *Inter Partes* Reexam
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Commissioner:

**PATENT OWNER'S RESPONSE TO
OFFICE ACTION OF APRIL 20, 2012**

Table of Contents

	Page
I. Introduction.....	1
II. Background.....	2
A. Overview of the '151 Patent	2
B. Applicable Legal Standards for Anticipation	3
C. Applicable Legal Standards for Obviousness.....	3
III. The Rejections Are Improper and Should Be Withdrawn.....	4
A. Certain References Have Not Been Shown to Be Prior Art	4
B. The Rejection of Claims 1-16 Under 35 U.S.C. § 102(b) Based on <i>Aventail v3.01</i> Should Be Withdrawn (Issue 1).....	6
1. Overview of <i>Aventail v3.01</i>	6
2. Independent Claim 1	7
a. <i>Aventail v3.01</i> Fails to Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”	7
b. <i>Aventail v3.01</i> Fails to Disclose “When the Intercepted DNS Request Corresponds to the Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”	9
3. Independent Claims 7 and 13.....	11
4. Dependent Claims 2, 8, and 14	12
5. Dependent Claims 3, 9, and 15	13
6. Dependent Claims 4, 10, and 16	14
7. Dependent Claims 5 and 11	15
8. Dependent Claims 6 and 12	15
C. The Rejection of Claims 1-16 Under 35 U.S.C. § 102(b) Based on <i>AutoSOCKS</i> Should Be Withdrawn (Issue 2).....	17

D.	The Rejection of Claims 1-16 Under 35 U.S.C. § 102(a) Based on <i>BinGO</i> Should Be Withdrawn (Issue 3).....	17
1.	The <i>BinGO</i> User’s Guide Has Not Been Shown to Expressly Incorporate <i>BinGO EFR</i>	17
2.	Overview of <i>BinGO</i>	18
3.	The Office Action and the Apple Request Rely on Two Alternative Embodiments of <i>BinGO</i>	19
4.	Independent Claim 1	21
a.	<i>BinGO</i> Fails to Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”	21
b.	<i>BinGO</i> Fails to Disclose “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”	24
5.	Independent Claims 7 and 13.....	29
6.	Dependent Claims 2, 8, and 14	30
7.	Dependent Claims 3, 9, and 15	31
8.	Dependent Claims 4, 10, and 16	32
9.	Dependent Claims 5 and 11	32
10.	Dependent Claims 6 and 12	33
E.	The Rejection of Claims 1, 2, 4-8, 10-14, and 16 Under 35 U.S.C. § 103(a) Based on <i>Beser</i> in View of <i>Kent</i> Should Be Withdrawn (Issue 4)	35
1.	Overview of <i>Beser</i>	35
2.	<i>Beser</i> Cannot Be Combined with <i>Kent</i>	36
3.	Independent Claim 1	38
a.	The Combination of <i>Beser</i> and <i>Kent</i> Fails to Disclose “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”.....	38
b.	The Combination of <i>Beser</i> and <i>Kent</i> Fails to Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”	39

c.	The Combination of <i>Beser</i> and <i>Kent</i> Fails to Disclose “When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer”	41
d.	The Combination of <i>Beser</i> and <i>Kent</i> Fails to Disclose “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”	42
4.	Independent Claims 7 and 13	43
5.	Dependent Claims 2, 8, and 14	44
6.	Dependent Claims 4, 10, and 16	45
7.	Dependent Claims 5 and 11	47
8.	Dependent Claims 6 and 12	48
F.	The Rejections Based on <i>Kiuchi</i> Should Be Withdrawn	49
1.	Overview of <i>Kiuchi</i>	50
2.	The Rejection of Claims 1-4, 6-10, and 12-16 Under 35 U.S.C. § 102(b) Based on <i>Kiuchi</i> Should Be Withdrawn (Issue 7)	51
a.	Independent Claim 1	51
(i)	<i>Kiuchi</i> Fails to Disclose “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”	51
(ii)	<i>Kiuchi</i> Fails to Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”	52
(iii)	<i>Kiuchi</i> Fails to Disclose “Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”	53
b.	Independent Claims 7 and 13	54
c.	Dependent Claims 3, 9, and 15	55
d.	Dependent Claims 6 and 12	56
e.	Dependent Claims 2, 4, 8, 10, 14, and 16	58

3.	The Rejection of Claims 5 and 11 Under 35 U.S.C. § 103(a) Based on <i>Kiuchi</i> in View of <i>Martin</i> Should Be Withdrawn (Issue 8).....	58
4.	The Rejection of Claims 1-4, 6-10, and 12-16 Under 35 U.S.C. § 103(a) Based on <i>Kiuchi</i> in View of <i>Edwards</i> Should Be Withdrawn (Issue 14).....	59
a.	Overview of <i>Edwards</i>	59
b.	Independent Claim 1	59
(i)	The Combination of <i>Kiuchi</i> and <i>Edwards</i> Fails to Disclose or Suggest “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”	60
(ii)	The Combination of <i>Kiuchi</i> and <i>Edwards</i> Fails to Disclose or Suggest “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”	61
(iii)	The Combination of <i>Kiuchi</i> and <i>Edwards</i> Fails to Disclose or Suggest “Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”	62
c.	Independent Claims 7 and 13.....	62
d.	Dependent Claims 3, 9, and 15	63
e.	Dependent Claims 6 and 12	65
f.	Dependent Claims 2, 4, 8, 10, 14, and 16	65
5.	The Rejection of Claims 5 and 11 Under 35 U.S.C. § 103(a) Based on <i>Kiuchi</i> in View of <i>Edwards</i> and <i>Martin</i> Should Be Withdrawn (Issue 15).....	66
G.	The Rejections Based on <i>Wesinger</i> Should Be Withdrawn.....	66
1.	The Rejection of Claims 1-4, 6-10, and 12-16 Under 35 U.S.C. § 102(e) Based on <i>Wesinger</i> Should Be Withdrawn (Issue 9).....	66
a.	Overview of <i>Wesinger</i>	66
b.	Independent Claim 1	67

(i)	<i>Wesinger</i> Does Not Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”	67
(ii)	<i>Wesinger</i> Fails to Disclose “When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer”	73
(iii)	<i>Wesinger</i> Fails to Disclose or Suggest “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”	78
c.	Independent Claims 7 and 13	80
d.	Dependent Claims 2, 8, and 14	80
(i)	The Request’s Analysis of Claims 2, 8, and 14 Is Inconsistent with Its Analysis of Independent Claims 1, 7, and 13	81
(ii)	<i>Wesinger</i> Does Not Disclose “Sending a Request to the Secure Server to Establish an Encrypted Channel When the Client Is Authorized to Access the Secure Server”	82
e.	Dependent Claims 3, 9, and 15	83
f.	Dependent Claims 6 and 12	84
g.	Remaining Dependent Claims 4, 10, and 16	84
2.	The Rejection of Claims 5 and 11 Under 35 U.S.C. § 103(a) Based on <i>Wesinger</i> in View of <i>Martin</i> Should Be Withdrawn (Issue 9)	85
3.	The Rejection of Claims 1-4, 6-10, and 12-16 Under 35 U.S.C. § 103(a) Based on <i>Wesinger</i> in View of <i>Edwards</i> Should Be Withdrawn (Issue 16)	85
1.	Independent Claim 1	85
a.	The Combination of <i>Wesinger</i> and <i>Edwards</i> Fails to Disclose or Suggest “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”	85

b.	The Combination of <i>Wesinger</i> and <i>Edwards</i> Fails to Disclose or Suggest “When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer”	87
c.	The Combination of <i>Wesinger</i> and <i>Edwards</i> Fails to Disclose or Suggest “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”	88
2.	Independent Claims 7 and 13.....	90
3.	Dependent Claims 2, 8, and 14.....	91
4.	Dependent Claims 3, 9, and 15.....	91
5.	Dependent Claims 6 and 12.....	93
6.	Remaining Dependent Claims 4, 10, and 16.....	94
7.	The Rejection of Claims 5 and 11 Under 35 U.S.C. § 103(a) Based on <i>Wesinger</i> in View of <i>Edwards</i> and <i>Martin</i> Should Be Withdrawn (Issue 17).....	94
H.	The Rejection of Claims 1, 7, and 13 Under 35 U.S.C. § 102(e) Based on <i>Blum</i> Should Be Withdrawn (Issue 11)	95
1.	Overview of <i>Blum</i>	95
2.	Independent Claim 1	96
a.	<i>Blum</i> Fails to Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”	96
b.	<i>Blum</i> Fails to Disclose or Suggest “When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer”	98
c.	<i>Blum</i> Fails to Disclose or Suggest “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”	99
3.	Independent Claims 7 and 13.....	101

I.	The Rejection of Claims 1-4, 6-10, and 12-16 Under 35 U.S.C. § 103(a) Based on <i>Aziz</i> in View of <i>Edwards</i> Should Be Withdrawn (Issue 12)	102
1.	Overview of <i>Aziz</i>	102
2.	Independent Claim 1.....	103
a.	The Combination of <i>Aziz</i> and <i>Edwards</i> Fails to Disclose or Suggest a “Data Processing Device . . . Storing a Domain Name Server (DNS) Proxy Module” that Performs All of the Recited Features.....	103
b.	The Combination of <i>Aziz</i> and <i>Edwards</i> Fails to Disclose or Suggest “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”	105
c.	The Combination of <i>Aziz</i> and <i>Edwards</i> Fails to Disclose or Suggest “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”	106
d.	The Combination of <i>Aziz</i> and <i>Edwards</i> Fails to Disclose or Suggest “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”.....	109
3.	Independent Claims 7 and 13.....	110
4.	Dependent Claims 2, 8, and 14	111
5.	Dependent Claims 3, 9, and 15	112
6.	Dependent Claims 6 and 12	114
7.	Dependent Claims 4, 10, and 16	115
J.	The Rejection of Claims 5 and 11 Under 35 U.S.C. § 103(a) Based on <i>Aziz</i> in View of <i>Edwards</i> and <i>Martin</i> Should Be Withdrawn (Issue 13).....	115
K.	Secondary Considerations Demonstrate Nonobviousness.....	116
IV.	Conclusion	119

I. Introduction

VirnetX Inc. (“VirnetX”), the owner of U.S. Patent No. 7,490,151 (“the ’151 patent”), provides the following remarks in response to the Office Action mailed April 20, 2012, in the above-identified reexamination proceedings. The U.S. Patent and Trademark Office (“Office”) issued this combined Office Action after issuing a Decision mailed March 15, 2012, merging the reexamination proceedings in control nos. 95/001,714 and 95/001,697, granted in response to a Request for Reexamination filed by Apple Inc. on July 25, 2011 (“Apple Request”), and a Request for Reexamination filed by Cisco Systems, Inc. on December 13, 2011 (“Cisco Request”).

The patent at issue in this merged reexamination, the ’151 patent, is part of a family of patents (“Munger patent family”) that stems from U.S. provisional application nos. 60/106,261 (“the ’261 application”), filed on October 30, 1998, and 60/137,704 (“the ’704 application”), filed on June 7, 1999. The ’151 patent is a divisional of U.S. application no. 09/504,783 (now U.S. Patent No. 6,502,135, “the ’135 patent”). The ’135 patent is a continuation-in-part of U.S. application no. 09/429,643 (now U.S. Patent No. 7,010,604, “the ’604 patent”), which claims priority to the ’261 and ’704 applications.

The Munger patent family discloses numerous inventions relating to secure communications. Patents in this family have been subject to several reexamination proceedings and district court actions. For instance, three other patents from the family were asserted in an action against Microsoft Corporation in the Eastern District of Texas.¹ The jury found the asserted claims willfully infringed and not invalid, and awarded VirnetX over one hundred million dollars in damages. (Ex. A-1 at 2.) Microsoft also sought reexamination of two of the patents, but all claims were confirmed during those proceedings. (*See* control nos. 95/001,269 and 95/001,270.) And just recently, the Office denied a request for reexamination of one of the patents in the Munger patent family. (Order in control no. 95/001,792.)

Given that the validity of the patents in the Munger patent family has now been tested multiple times, and for the other reasons set forth below, including that the asserted references do not disclose or suggest the combination of features recited in the claims, Patent Owner requests reconsideration and withdrawal of all the rejections in the Office Action and confirmation of the patentability of all of the claims of the ’151 patent.

¹ One of these patents, U.S. Patent No. 6,839,759, was asserted initially but was dropped from this case before trial.

This Response is supported by a Declaration of Angelos D. Keromytis, Ph.D. (“Keromytis Decl.”) and by a Declaration of Dr. Robert Dunham Short III (“Short Decl.”).

II. Background

A. Overview of the '151 Patent

The '151 patent discloses embodiments relating to automatically initiating encrypted channels and/or automatically creating secure channels between devices connected to a network. (Keromytis Decl. ¶ 15.) For example, one such embodiment may establish encrypted channels between a client and a secure server when a domain name server (DNS) proxy module intercepts a DNS request sent by the client and determines that the DNS request corresponds to a secure server. ('151 patent 37:50-38:21; Keromytis Decl. ¶ 15.)

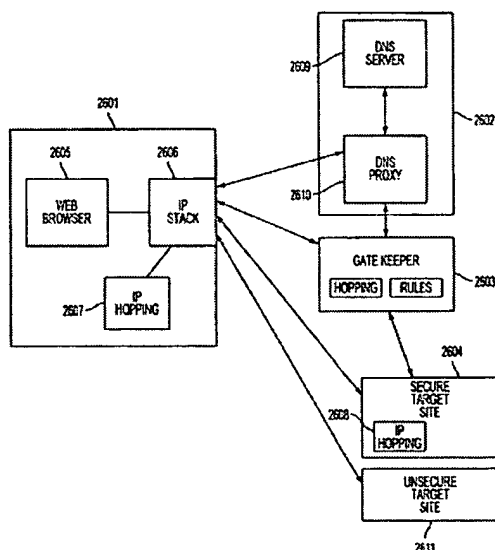


FIG. 26

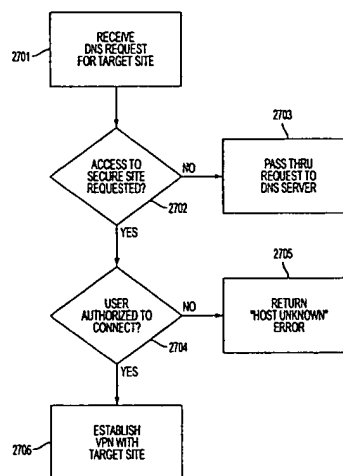


FIG. 27

As shown in Figures 26 and 27 of the '151 patent, reproduced above, a DNS proxy 2610 may intercept a DNS request from client 2601. ('151 patent 37:60-61; Keromytis Decl. ¶16.) The DNS proxy 2610 determines whether the DNS request corresponds to a secure target site 2604, such as a secure server. ('151 patent 37:61-62; Keromytis Decl. ¶ 16.) If the DNS request corresponds to a secure site, the DNS proxy 2610 may, in certain embodiments, determine whether the client 2601 is authorized to access the site. ('151 patent 37:62-66; Keromytis Decl. ¶ 16.) If so, the DNS proxy 2610 may automatically initiate an encrypted channel between the client 2601 and the secure target site 2604. ('151 patent 37:62-38:11; Keromytis Decl. ¶ 16.)

If, on the other hand, the intercepted DNS request does not correspond to a secure target site 2604, DNS proxy server 2610 may forward the request to a conventional DNS server 2609, which may return the IP address of an unsecure target site 2611. ('151 patent 38:36-43; Keromytis Decl.

¶ 17.)

The claims of the '151 patent are directed to some of these embodiments. Claims 1, 7, and 13 are independent claims. Claims 2-6 depend from claim 1, claims 8-12 depend from claim 7, and claims 14-16 depend from claim 13. As explained below, none of the references relied upon by the Office Action, either individually or in combination, discloses or suggests the combination of features recited in these claims.

B. Applicable Legal Standards for Anticipation

To support a rejection under 35 U.S.C. § 102, each and every element of each claim at issue must be found in that single reference. See M.P.E.P. § 2131. “The identical invention must be shown in as complete detail as is contained in the . . . claim.” *Id.* (quoting *Richardson v. Suzuki Motor Co.*, 868 F.2d 1126, 1236 (Fed. Cir. 1989)). Further, “[t]he elements must be arranged as required by the claim” *Id.* (citing *In re Bond*, 910 F.2d 831, 832 (Fed. Cir. 1990)). Thus, “unless a reference discloses within the four corners of the document not only all of the limitations claimed but also all of the limitations arranged or combined in the same way as recited in the claim, it . . . cannot anticipate under 35 U.S.C. § 102.” *Net MoneyIN, Inc. v. VeriSign, Inc.*, 545 F.3d 1359, 1371 (Fed. Cir. 2008). Moreover, “[t]he requirement that the prior art elements themselves be ‘arranged as in the claim’ means that claims cannot be ‘treated . . . as mere catalogs of separate parts, in disregard of the part-to-part relationships set forth in the claims and that give the claims their meaning.’” *Therasense, Inc. v. Becton, Dickinson & Co.*, 593 F.3d 1325, 1332 (Fed. Cir. 2010) (quoting *Lindemann Maschinenfabrik GmbH v. Am. Hoist & Derrick Co.*, 730 F.2d 1452, 1459 (Fed. Cir. 1984)).

C. Applicable Legal Standards for Obviousness

Obviousness is a question of law based on underlying factual inquiries that include, inter alia, determining the scope and content of the prior art and ascertaining the differences between the claimed invention and prior art. See M.P.E.P. § 2141(II). In order to establish a prima facie case of obviousness, the Examiner must “include[] findings of fact concerning the state of the art and the teachings of the references” *Id.* Moreover, “[o]nce the findings of fact are articulated, [the Examiner] must provide an explanation to support an obviousness rejection under 35 U.S.C. [§] 103.” *Id.*

The reasons why the claimed invention would have been obvious must be clearly articulated and cannot be premised on conclusory statements. M.P.E.P. § 2142. In addition, the references relied on must be enabling, *id.* at § 2145, and “[t]he mere fact that references *can* be combined or

modified does not render the resultant combination obvious unless the results would have been predictable to one of ordinary skill in the art” at the time the invention was made, *id.* at § 2143.01(III) (internal citation omitted). “All words in a claim must be considered in judging the patentability of that claim against the prior art.” *Id.* at § 2143.03 (internal citation omitted). Also, “[i]n determining the differences between the prior art and the claims, the question under 35 U.S.C. [§] 103 is not whether the differences *themselves* would have been obvious, but whether the claimed invention *as a whole* would have been obvious.” *Id.* at § 2141.02(I) (internal citations omitted).

III. The Rejections Are Improper and Should Be Withdrawn

A. Certain References Have Not Been Shown to Be Prior Art

As a preliminary matter, the Requests and the Office Action rely on the following five references without showing that these references have been published:

1. Aventail Connect v3.01/2.51 Administrator’s Guide (“*Aventail v3.01*”) (submitted by Apple as Exhibit X2);
2. Aventail AutoSOCKS v2.1 Administrator’s Guide (“*AutoSOCKS*”) (submitted by Apple as Exhibit X3);
3. S. Kent, “Security Architecture for IP,” RFC 2401 (“*Kent*”) (submitted by Apple as Exhibit X6);
4. BinGO! User’s Guide (“*BinGO*”) (submitted by Apple as Exhibit X7); and
5. D.M. Martin, “A Framework for Local Anonymity in the Internet” (“*Martin*”) (submitted by Cisco as Exhibit D-6).

Because the Office and the Request have not shown that any of the above-listed references are printed publications, the rejections of the claims in view of these references (specifically, the rejections corresponding to Issues 1-4, 8, 10, 13, 15, and 17) are improper and should be withdrawn.² (*See* OA at 6-32.)

Since this reexamination was initiated before the America Invents Act’s reexamination provisions took effect, reexamination of the ’151 patent is limited to situations where a substantial new question of patentability has been shown “based on patents or printed publications.” M.P.E.P. § 2247. The statutory phrase “printed publication” has been interpreted to mean that the alleged prior art reference must have been sufficiently accessible to the public interested in the art. *In re Cronyn*,

² VirnetX filed petitions on November 7, 2011, and November 9, 2011, raising this issue regarding *BinGO* and *Martin*, respectively. Cisco and Apple filed petitions in opposition. The Office denied VirnetX’s petitions and dismissed Cisco’s and Apple’s petitions as being moot. (*See* Decisions mailed Dec. 1, 2011.)

890 F.2d 1158, 1160 (Fed. Cir. 1990) (quoting *Constant v. Advanced Micro-Devices, Inc.*, 848 F.2d 1560, 1568 (Fed. Cir. 1988)).

The party asserting the prior art bears the burden of establishing a date of publication. See *Carella v. Starlight Archery*, 804 F.2d 135, 139 (Fed. Cir. 1986) (finding that a mailer did not qualify as prior art because there was no evidence as to when the mailer was received by any of the addressees); see also M.P.E.P. §§ 716.01(c), 2128; *In re Wyer*, 655 F.2d 221, 227 (C.C.P.A. 1981) (“[T]he one who wishes to characterize the information, in whatever form it may be, as a ‘printed publication’ . . . should produce sufficient proof of its dissemination or that it has otherwise been available and accessible to persons concerned with the art to which the document relates and thus most likely to avail themselves of its contents.”) (emphasis added). Here, the Office and the Requesters bear the burden of establishing a prima facie case of unpatentability. This includes, among other things, demonstrating that the references relied upon are proper prior art. See *In re Hall*, 781 F.2d 897 (Fed. Cir. 1986).

But Cisco, Apple, and the Office have not shown that the *Kent*, *BinGO*, or *Martin* references were publicly available or that they are printed publications. Cisco’s Request baldly asserts, without any evidence, that *Martin* is a printed publication. (See Cisco Req. at 15-16.) Apple’s Request also asserts, without any evidence, that *Kent* and *BinGO* are printed publications. (See Apple Req. at 11-12.) But *Martin*, submitted by Cisco as Exhibit D-6, and *Kent* and *BinGO*, submitted by Apple as Exhibits X6 and X7, contain no indication whatsoever that they were published or even publicly available before the effective filing date of the ’151 patent. These assertions by the Requesters, therefore, are nothing more than attorney argument and are not evidence that those references are printed publications.

Apple also has not shown that *Aventail v3.01* or *AutoSOCKS* were publicly available or that they are printed publications. Apple submitted uncorroborated declarations of Hopen, Fratto, and Chester (“the Declarants”) to support its allegation that *Aventail v3.01* and *AutoSOCKS* are prior art, but the Declarants fail to provide any evidence to corroborate that these documents were disseminated and publicly available before the effective filing date of the ’151 patent. For example, Mr. Chester states that one or more of *Aventail v3.01* and *AutoSOCKS* were distributed with deployments of Aventail products to more than 65,000 people. But if that many copies were distributed, why has Apple not offered any documentation of such distribution? Also, Mr. Hopen has testified that (1) although Aventail, Inc. had email, he does not have any email evidencing distribution of *Aventail v3.01* and *AutoSOCKS*; (2) he does not have evidence that *Aventail v3.01* and

AutoSOCKS were available for download on the Internet in the relevant time period; and (3) he does not have evidence that *Aventail v3.01* and *AutoSOCKS* were published in a journal. (Ex. A-4, Apr. 11, 2012, Hopen Dep. Tr. 55:1-7, 119:11-23, 189:1-191:6.) Thus, despite the number of alleged distributions of *Aventail v3.01* and *AutoSOCKS*, it is surprising that Mr. Hopen does not have any corroborative evidence that they were actually distributed and publicly accessible. Accordingly, Patent Owner respectfully submits that Apple has not satisfied its burden of showing that *Aventail v3.01* and *AutoSOCKS* are prior art.

The Office has not remedied Cisco's and Apple's shortcomings. Neither the Office's Orders nor the merged Office Action indicate that the Office investigated whether *Aventail v3.01*, *AutoSOCKS*, *Kent*, *BinGO*, and *Martin* were publicly available or qualified as printed publications. Instead, the Office Action adopts portions of the proposed rejections without making any initial determination or even providing any indication as to whether *Aventail v3.01*, *AutoSOCKS*, *Kent*, *BinGO*, and *Martin* were publicly available or were printed publications. (*See generally* OA.)

In view of the above, Cisco, Apple, and the Office have failed to demonstrate that any of *Aventail v3.01*, *AutoSOCKS*, *Kent*, *BinGO*, and *Martin* qualifies as a prior art reference. Thus, the rejections of the claims based on these references, corresponding to Issues 1-4, 8, 10, 13, 15, and 17, should be withdrawn. (*See id.* at 6-32.)

B. The Rejection of Claims 1-16 Under 35 U.S.C. § 102(b) Based on *Aventail v3.01* Should Be Withdrawn (Issue 1)

The Office Action rejects claims 1-16 under § 102(b) based on the *Aventail Connect v3.01/2.51 Administrator's Guide* (Apple Req. Ex. X2) ("*Aventail v3.01*"). (*Id.* at 6.) For the reasons discussed below, this rejection should be withdrawn and the claims should be confirmed.

1. Overview of *Aventail v3.01*

Aventail v3.01 is an administrator's guide for configuring *Aventail Connect*, a client component of the *Aventail ExtraNet Center*, an extranet solution. (*Aventail v3.01* 3, 7.) *Aventail Connect* works in connection with extranet servers running the SOCKS protocol, including the *Aventail ExtraNet Server*, the SOCKS 5 server component of the *Aventail ExtraNet Server*. (*Id.* at 7.) *Aventail v3.01* discloses two primary embodiments:

- (1) *Aventail Connect* may be used to provide secure inbound access, i.e., allowing an organization to provide its mobile employees and partners secure access to the organization's private network, extranet, or LAN from remote locations over the Internet. (*E.g., id.* at 5, 7, 72.)

- (2) Aventail Connect may also be used as a simple proxy client for managed outbound access, e.g., from a corporate network to the Internet, through a SOCKS-compliant server. (*E.g., id.* at 5, 7, 59-61.)

In the first embodiment, Aventail Connect accesses the private network through the Aventail ExtraNet Server. (*Id.* at 72.) The Aventail ExtraNet Server restricts inbound access by allowing only authorized client computers running Aventail Connect to send or receive data to a computer on the private network, and provides an encrypted connection between the Aventail ExtraNet Server and the external client computer. (*See, e.g., id.* at 63.)

In the second embodiment, Aventail Connect may be configured to route certain traffic from a client computer running Aventail Connect to a SOCKS-compliant proxy server to traverse a firewall (*id.* at 6-7), or in some cases, to traverse multiple firewalls using successive proxy servers (*id.* at 59-64). Routing is accomplished, in part, by an administrator first defining which of several possible SOCKS proxy servers Aventail Connect should use when routing connections. (*Id.* at 33-35, figure depicting that a user may choose SOCKS v4, SOCKS v5, or HTTP proxy.) The administrator may then define destinations (e.g., hostnames) and create redirection rules. (*See id.* at 35-37.) A redirection rule defines, for a particular destination, what type of traffic (i.e., TCP and/or UDP) will be allowed to be routed to that destination, and which proxy server will be used to route that traffic. (*Id.* at 38-40.)

2. Independent Claim 1

Independent claim 1 is directed to a data processing device storing a domain name server (DNS) proxy module. *Aventail v3.01* fails to disclose the combination of features recited in this claim for at least the reasons discussed below.

a. ***Aventail v3.01* Fails to Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”**

Independent claim 1 recites, among other things, “a domain name server (DNS) proxy module that . . . performs the step[] of . . . determining whether the intercepted DNS request corresponds to a secure server.” *Aventail v3.01* does not disclose this feature.

The Apple Request and the Office Action assert that *Aventail v3.01* discloses determining whether an intercepted DNS request corresponds to a secure server because *Aventail v3.01* describes evaluating a hostname by way of matching a redirection rule: if the redirection rule is matched, the hostname is flagged by creating a false DNS entry, and the false DNS entry is used “during the

process of establishing a connection to *re-direct* the request to the Aventail Extranet Server.” (Apple Req. at 24, citing *Aventail v3.01* 12, emphasis added.) This is incorrect. (Keromytis Decl. ¶ 22.)

Whether or not a hostname is flagged by creating a false DNS entry does not indicate whether the alleged DNS request corresponds to a secure server, as false DNS entries may result even if a redirection rule is not matched. (See *Aventail v3.01* 12, step 1; Keromytis Decl. ¶ 23.) For example, a false DNS entry may be created as a result of selecting a DNS proxy option, i.e., to proxy all DNS lookups that cannot be looked up directly, whether for secure destinations or not. (See *Aventail v3.01* 12, step 1, bullet point 3; Keromytis Decl. ¶ 23.) Thus, a person of ordinary skill in the art would have understood that the redirection rules and false DNS entries of *Aventail v3.01* do not disclose any link between the alleged DNS request and whether a server is secure or not. (Keromytis Decl. ¶ 23.)

Furthermore, the Office Action and the Request fail to explain why matching a hostname to a redirection rule to “re-direct a request” is the same as determining whether a DNS request corresponds to a secure server, as recited in claim 1. (See OA at 6-7; Apple Req. at 24.) The Office Action cites a portion of *Aventail v3.01* discussing forwarding a hostname to the SOCKS server, but does not explain how merely forwarding that hostname to the SOCKS server to perform hostname resolution actually discloses “determining whether the intercepted DNS request corresponds to a secure server.” (See OA at 7.)

Moreover, not a single portion of *Aventail v3.01* has been cited to show that any particular component of *Aventail v3.01* corresponds to a “secure server,” or to even identify the alleged “secure server.” The only server identified in the Request is the Extranet Server, as the Request alleges that an encrypted tunnel is established “provided the client successfully authenticated to the Extranet Server.” (Apple Req. at 24.) Thus, the Request has not demonstrated, or even properly alleged, that *Aventail v3.01* discloses the elements of claim 1. Accordingly, the rejection of claim 1 over *Aventail v3.01* is deficient and should be withdrawn.

For these reasons, the § 102(b) rejection of claim 1 based on *Aventail v3.01* should be withdrawn, and the claim confirmed.

b. *Aventail v3.01 Fails to Disclose “When the Intercepted DNS Request Corresponds to the Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”*

Independent claim 1 recites, among other things, “when the intercepted DNS request corresponds to the secure server, automatically initiating an encrypted channel between the client and the secure server.” *Aventail v3.01* does not disclose this feature.

The Office Action and the Apple Request assert that *Aventail v3.01* discloses the “automatically initiating” step of claim 1 because if a connection request matched a redirection rule, a connection would be established between the client running Aventail Connect and a proxy server, the user would be authenticated, and, upon successful authentication, a secure connection would be established between the client computer and the destination computer. (Apple Req. at 25-26.) This is incorrect.

In particular, the Request does not explain how the cited portions of *Aventail v3.01* regarding proxying a connection into a private network based on a “security policy” or server “configuration” includes automatically initiating an encrypted channel when an intercepted DNS request corresponds to the secure server. (*Id.* at 25, citing *Aventail v3.01* 72-73.) The Request asserts that *Aventail v3.01* discloses that “Aventail Connect will call Winsock . . . to begin the TCP handshake with the server designated in the configuration file.” (*Id.* at 26, citing *Aventail v3.01* 12.) However, the cited portion of *Aventail v3.01* discloses that the alleged TCP handshake results from a “routable IP address,” not that it is related to the false DNS entry or the alleged DNS request relied upon by the Request to support its anticipation argument for the “determining” step of claim 1. (*See Aventail v3.01* 12, step 2.a, compare bullet points.) Thus, the alleged TCP handshake fails to evidence any authentication or encryption based on the false DNS entry or the alleged DNS request that the Request and the Office Action highlight as disclosing claim 1’s “determining” step. Therefore, the Request and the Office Action have improperly mixed and matched various unconnected features and embodiments of *Aventail v3.01* to try to meet the claim language, and, therefore, the § 102(b) rejection of claim 1 should be withdrawn. *Net MoneyIN*, 545 F.3d at 1371.

The Request further contends that the alleged encrypted channel is automatically established because “the Aventail ExtraNet Server require[s] all users to use Aventail Connect to authenticate and encrypt their sessions before any connection to the internal private network(s).” (Apple Req. at 43, citing *Aventail v3.01* 73.) But requiring Aventail users to authenticate and encrypt their sessions before connecting does not show any relationship between a DNS request and the encryption in

Aventail v3.01. (Keromytis Decl. ¶ 25.) Indeed, *Aventail v3.01* does not teach any link between the alleged DNS request and the encryption, much less that encryption is automatically initiated when an “intercepted DNS request corresponds to a secure server,” as recited by claim 1. (*Id.* ¶25.)

The Office Action, meanwhile, points to page 12 of *Aventail v3.01*, which states that “if the request contains a false DNS entry (from step 1), it will be proxied . . . ,” in an attempt to show the claimed feature of “when the intercepted DNS request corresponds to a secure server.” (OA at 7.) But, as previously explained, the Request and the Office Action cannot properly contend that evaluating a connection request for the presence of a false DNS entry discloses determining that a DNS request corresponds to a secure server. (Keromytis Decl. ¶ 26.) This is because a false DNS entry may be created regardless of whether a destination is allegedly determined to be secure or not. (*Id.*) For example, a false DNS entry will be created as a result of selecting a DNS proxy option, i.e., to proxy all DNS lookups that cannot be looked up directly, whether for secure destinations or not. (*See Aventail v3.01* 12, step 1, bullet point 3; Keromytis Decl. ¶ 26.)

Furthermore, *Aventail v3.01* does not disclose that the creation of a false DNS entry automatically initiates a connection, much less an encrypted channel. (Keromytis Decl. ¶ 27.) *Aventail v3.01* teaches that, in step 2.a, Aventail Connect checks an already existing connection request to determine whether the request contains a false DNS entry. (*Aventail v3.01* 12.) *Aventail v3.01* does not disclose that whether a completed connection is subsequently encrypted has anything to do with a DNS request, let alone automatically initiating an encrypted channel “when the intercepted DNS request corresponds to the secure server,” as recited in claim 1. *Aventail v3.01* explains that encryption is initiated, if at all, “[w]hen the connection is completed” to the SOCKS server. (*Id.*, “step 2.b”, emphasis added.) Thus, *Aventail v3.01* does not teach any link between a *DNS request* and the encryption, much less automatically initiating an encrypted channel between the client and the secure server “when the intercepted DNS request corresponds to a secure server,” as recited in claim 1. (Keromytis Decl. ¶ 27.)

Finally, the Request improperly mixes and matches the various separate embodiments of *Aventail v3.01* by pointing to the inbound access embodiment of *Aventail v3.01* on pages 72-73, which describe that when confronted with inbound traffic, an “Aventail ExtraNet Server will proxy mobile user traffic . . . to those resources allowed,” and then turning to the outbound embodiment of *Aventail v3.01* on pages 7 and 10, which is directed to providing outbound access through an extranet (SOCKS) server, routing and redirecting traffic, and potentially encrypting connections. (Apple Req. at 25.) But the Request does not describe how these two portions of *Aventail v3.01*, describing

different embodiments and functionalities and separated by over sixty pages, can be combined to disclose automatically initiating an encrypted channel between the client and the secure server “when the intercepted DNS request corresponds to the secure server,” as recited in claim 1. Thus, the Request’s anticipation analysis is improper as these mixed and matched embodiments would not be “arranged or combined in the same way as recited in the claim,” as required by *Net MoneyIN*, 545 F.3d at 1371.

For at least these reasons, a person of ordinary skill would not have understood *Aventail v3.01* to disclose an encrypted channel between the client and the secure server or automatically initiating such a channel when the intercepted DNS request corresponds to the secure server. (Keromytis Decl. ¶ 27.) Accordingly, Patent Owner requests that the rejection of claim 1 be withdrawn, and its patentability confirmed.

3. Independent Claims 7 and 13

Independent claim 7 recites features similar to those described above for claim 1. For example, claim 7’s recited feature of “determining whether the intercepted DNS request corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Also, claim 7’s recited feature of “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” is similar to the “automatically initiating” feature of claim 1, also discussed above. Accordingly, *Aventail v3.01* does not disclose these features of claim 7 for similar reasons as those discussed above with respect to claim 1.

Independent claim 13 also recites features similar to those described above for claim 1. For example, claim 13’s recited feature of “determining whether a DNS request sent by a client corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Independent claim 13 also recites features that differ from the features recited in claim 1, and these features are not even addressed by the Apple Request or the Office Action. For example, claim 13 recites “when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel” The Apple Request, however, ignores this difference in claim language and instead quotes another portion of independent claim 1 when purporting to reject claim 13. (*See, e.g.*, Apple Req. at 42-46.) By ignoring the language of claim 13 and instead analyzing a feature of claim 1, the rejection of claim 13 in view of *Aventail v3.01* is improper for failing to consider all of the words in the claim. M.P.E.P. § 2131; *see also id.* at § 2143.04 (“*All words* in a claim must be considered in judging the patentability of that claim against the prior art.”) (emphasis added) (internal

citations omitted). Moreover, to the extent the Requester and the Office later assert that the features recited in claim 13 are similar to the features recited in claim 1, Patent Owner asserts that *Aventail v3.01* does not disclose these features of claim 13 for similar reasons as those discussed above with respect to claim 1.

Thus, for these reasons, Patent Owner requests that the rejection of claims 7 and 13 under § 102(b) be withdrawn, and their patentability confirmed.

4. Dependent Claims 2, 8, and 14

Claims 2, 8, and 14 depend from independent claims 1, 7, and 13, respectively, and include all of their features. Thus, *Aventail v3.01* does not anticipate these claims, and the rejection of these claims should be withdrawn for at least the reasons discussed above in connection with independent claims 1, 7, and 13. Claims 2, 8, and 14 also distinguish over *Aventail v3.01* for additional reasons. For example, claims 2, 8, and 14 recite “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.” *Aventail v3.01* does not disclose this feature.

The Office Action and the Apple Request allege that *Aventail v3.01* teaches this feature because (1) “‘Aventail Connect will . . . begin the TCP handshake with the server designated in the configuration file,’ and then, when that connection is completed would authenticate the user”; and (2) a subsequent proxy connection corresponds to “sending a request to the secure server to establish an encrypted channel between the secure server and the client.” (*See, e.g.*, OA at 8; Apple Req. at 27-28, discussing claim 2.) This is incorrect.

As discussed above, the disclosure in *Aventail v3.01* regarding the initiation of a “TCP handshake with the server designated in the configuration file” is directed to a separate feature of *Aventail v3.01* than that pointed to by the Request to satisfy the determining step of independent claim 1. Specifically, the cited portion of *Aventail v3.01* discloses that the cited TCP handshake results from finding a “routable IP address,” not from the previously cited “false DNS entry” or alleged DNS request. (*See Aventail v3.01* 12, step 2.a, “If the request contains a routable IP address”) Accordingly, the Request cannot use the cited TCP handshake, which corresponds to a routable IP address, to satisfy one element of a claim, while at the same time pointing to the separate “false DNS entry” or alleged DNS request to satisfy another element of that claim under 35 U.S.C. § 102. *Net MoneyIN*, 545 F.3d at 1371.

Furthermore, the Request, having alleged that a proxy connection is made through a SOCKS server only *after* authentication and encryption have already been established, cannot now argue that

a proxy connection establishes the (already established) encryption. For instance, the Request alleges with respect to claim 1 that “if authentication is successful, a secure connection would be established.” (Apple Req. at 26.) But the Request now alleges with respect to claim 2 that a proxy request, made after the connection and authentication with the SOCKS server is completed, is a request to establish an encrypted channel. As explained previously, the Request cannot pick and choose disparate features from various embodiments of *Aventail v3.01* to support a rejection under 35 U.S.C. § 102. *Net MoneyIN*, 545 F.3d at 1371.

For at least these reasons, Patent Owner requests that the § 102(b) rejection of claims 2, 8, and 14 be withdrawn, and their patentability confirmed.

5. Dependent Claims 3, 9, and 15

Claims 3, 9, and 15 depend from one or more of claims 1, 2, 7, 8, 13, and 14, and include all of their features. Thus, *Aventail v3.01* does not anticipate claims 3, 9, and 15, and the rejection of these claims should be withdrawn for at least the reasons discussed above with respect to those claims. Claims 3, 9, and 15 also distinguish over *Aventail v3.01* for additional reasons. For example, claims 3, 9, and 15 recite “when the client is not authorized to access the secure server, returning a host unknown error message to the client.” *Aventail v3.01* does not disclose at least these features.

The Office Action, adopting the proposed rejection of the Apple Request, states that “[r]eturning a host unknown error message when the client application is not authorized to access the secure server is inherent” to TCP/IP and SOCKS v5 protocols. (OA at 9.) This is incorrect. (Keromytis Decl. ¶ 28.)

The Request alleges, by way of declaration, that this feature is disclosed by *Aventail v3.01* because the SOCKS v5 protocol informs a SOCKS client whether authentication was successful, and if not, an error value is returned to the *Aventail* client. (Apple Req. at 29, citing Ex. E2, Fratto, at 136-42.) The Request gives two examples for error values: “X’02’ connection not allowed by ruleset” or “X’05’ Connection refused”. (*Id.*) The Request then makes the unsubstantiated allegation that both “*Aventail Connect v3.01* and/or the *Aventail Extranet Server*” are a “DNS server,” and that a DNS server must provide standard responses, including a “host not found” error. (*Id.*)

But anticipation requires showing not only all of the limitations claimed, “but also all of the limitations arranged or combined in the same way as recited in the claim.” *Net MoneyIN*, 545 F.3d at 1371. The Request does not describe how error values corresponding to a “connection not allowed by ruleset” or a “connection refused” amounts to a “host unknown error message,” as recited by

claim 3. Moreover, nothing in *Aventail v3.01* discloses that the Aventail Connect client module or Aventail Extranet Server may function as a “DNS Server,” as proposed by the Request. (Keromytis Decl. ¶ 29.) The Request simply weaves this allegation into its proposed rejection without pointing to any support within the reference. For these reasons, the rejection is deficient and should be withdrawn.

Additionally, the Request itself alleges that multiple different types of error messages may be returned when SOCKS authentication fails, none of which are the claimed host unknown error message. (Apple Req. at 29.) Indeed, there may be many ways to return an error, and *Aventail v3.01* does not disclose that the Aventail Connect client module or the Aventail ExtraNet Server necessarily uses any one of them, or returns any error message at all. (Keromytis Decl. ¶ 29.) Accordingly, the missing descriptive matter is not necessarily present in the cited portions of *Aventail v3.01*, and therefore cannot support a rejection based on inherency. *In re Robertson*, 169 F.3d 743, 745 (Fed. Cir. 1999).

It is also not appropriate to rely solely on interpretation or “common knowledge” in the art without evidentiary support in the record as the principal evidence upon which a rejection is based. M.P.E.P. §§ 2144.03, 2112 (“In relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art.”). The way a particular component may handle a failure to authenticate is a subject matter of a highly technical field that requires a significant skill in the art. Thus, Patent Owner respectfully submits that the Office Action may not properly make the unsupported assertion that the subject matter of claims 3, 9, and 15 would be inherent or anticipated.

Accordingly, for these reasons, Patent Owner requests that the § 102(b) rejection of claims 3, 9, and 15 be withdrawn, and their patentability confirmed.

6. Dependent Claims 4, 10, and 16

Claims 4, 10, and 16 depend from one or more of claims 1-3, 7-9, and 13-15, and include all of their features. Thus, *Aventail v3.01* does not anticipate these claims, and the rejection of these claims under § 102(b) should be withdrawn and the claims should be confirmed for at least the reasons discussed above in connection with claims 1-3, 7-9, and 13-15.

7. Dependent Claims 5 and 11

Claims 5 and 11 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, *Aventail v3.01* does not anticipate these claims, and the rejection of these claims should be withdrawn for at least the reasons discussed above in connection with independent claims 1 and 7. Claims 5 and 11 also distinguish over *Aventail v3.01* for additional reasons. For example, claims 5 and 11 recite that the feature of automatically initiating the encrypted channel between the client and the secure server “comprises establishing an [IP] address hopping scheme between the client and the secure server.”

The Office Action and the Apple Request assert that *Aventail v3.01* discloses the features of claims 5 and 11 because *Aventail v3.01* discloses a MultiProxy scheme and a Proxy Chaining scheme. (OA at 9; Apple Req. at 29-30.) This is incorrect. (Keromytis Decl. ¶ 30.)

The proxy schemes disclosed by *Aventail v3.01* are implemented merely to satisfy the “need to traverse multiple firewalls.” (*Aventail v3.01* 59; Keromytis Decl. ¶ 31.) Providing a mechanism for traversing multiple firewalls does not contribute in any meaningful way towards establishing an IP address hopping scheme between the client and the secure server. (Keromytis Decl. ¶ 31.) Indeed, nothing in *Aventail v3.01* discloses that traffic would be routed in either proxy scheme any differently than normal Internet traffic, much less that the alleged automatic initiating of the alleged encrypted channel would involve *establishing* either proxy scheme. (*Id.*)

Moreover, what the previous rejection of claims 1 and 7 points to in *Aventail v3.01* as *initiating* the alleged encrypted channel between the client and alleged secure server does not involve the MultiProxy scheme or the Proxy Chaining scheme, which are described some forty or so pages later in the reference with respect to another embodiment. Dependent claims 5 and 11 depend from claims 1 and 7, respectively, and pertain to the step of automatically initiating the encrypted channel recited in those claims. Therefore, by mixing and matching various unrelated features of *Aventail v3.01*, the rejection of claims 5 and 11 does not meet all of the elements of those claims, nor the requirements set forth by *Net MoneyIN*, 545 F.3d at 1371.

Thus, for these reasons, Patent Owner requests that the rejection of claims 5 and 11 under § 102(b) be withdrawn, and their patentability confirmed.

8. Dependent Claims 6 and 12

Claims 6 and 12 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, *Aventail v3.01* does not anticipate these claims, and the rejection of these claims should be withdrawn for at least the reasons discussed above in connection with independent claims

1 and 7. Claims 6 and 12 also distinguish over *Aventail v3.01* for additional reasons. For example, claims 6 and 12 recite that the feature of automatically initiating the encrypted channel between the client and the secure server “avoids sending a true IP address of the secure server to the client.” *Aventail v3.01* does not disclose this feature.

The Apple Request alleges that *Aventail v3.01* anticipates these claims because “the true IP address of the secure destination computer would not be sent to the client computer . . . [;] rather, the client . . . would send its traffic destined for the secure destination computer to the Aventail Extranet Server, which would then route that traffic to the secure destination computer.” (See, e.g., Apple Req. at 31-32, discussing claim 6.) Similarly, the Office Action cites portions of *Aventail v3.01* disclosing that the Aventail Extranet Server prevents a direct connection between two different LANs. (OA at 10.)

As an initial matter, if the Office Action and the Request are alleging that a SOCKS server or an Aventail Extranet Server is a “secure server” with respect to claims 1 or 7, then the Office Action and the Request cannot now allege that some other computer corresponds to the “secure server” and still meet the requirements for anticipating claims 6 and 12 set forth by *Net MoneyIN*, 545 F.3d at 1371.

Furthermore, not a single portion of *Aventail v3.01* has been cited as disclosing avoiding sending a true IP address of a secure server to a client. The rejection appears to be based on an unsubstantiated presumption, put forth by the Request, that the system described by *Aventail v3.01* operates in this manner. (See Apple Req. at 31-32.) But a “claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference.” *Verdegaal Bros. v. Union Oil Co. of Cal.*, 814 F.2d 628, 631 (Fed. Cir. 1987). Since *Aventail v3.01* has not disclosed the claimed feature, and that feature has not been alleged to be inherent, a rejection of claims 6 and 12 cannot be maintained. *In re Rijckaert*, 9 F.3d 1531, 1533, 28 U.S.P.Q.2d 1955, 1957 (Fed. Cir. 1993) (“[W]hen the PTO asserts that there is an explicit or implicit teaching or suggestion in the prior art, it must indicate *where* such a teaching or suggestion appears in the prior art.”) (emphasis added). See also *Ex parte Schricker*, 56 U.S.P.Q.2d 1723, 1725 (B.P.A.I. 2000) (“[W]hen an examiner relies on inherency, it is incumbent on the examiner to point to the ‘page and line’ of the prior art which justifies an inherency theory.”).

Even if inherency were a basis for rejection, *Aventail v3.01* does not “make clear that the missing descriptive matter is *necessarily* present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill.” *In re Robertson*, 169 F.3d at 745 (emphasis

added); *see also* M.P.E.P. § 2112. There is simply no reason why routing traffic through an intermediate server (e.g., a proxy server) would *necessarily* prevent the proxy server from providing the client with the true IP address of the proxy server or the destination. Indeed, traffic could be routed through an intermediate server while still providing the client with the true IP address of the same. (Keromytis Decl. ¶ 32.)

Thus, the rejection of claims 6 and 12 under § 102(b) should be withdrawn, and their patentability confirmed.

For at least the foregoing reasons, Patent Owner respectfully requests that the rejection of claims 1-16 under § 102(b) based on *Aventail v3.01* be withdrawn, and the patentability of these claims be confirmed.

C. The Rejection of Claims 1-16 Under 35 U.S.C. § 102(b) Based on *AutoSOCKS* Should Be Withdrawn (Issue 2)

Claims 1-16 are rejected under 35 U.S.C. § 102(b) as allegedly being anticipated by the *Aventail AutoSOCKS v2.1 Administrator's Guide* (Apple Req. Ex. X3) ("*AutoSOCKS*"). Patent Owner submits that the claims are patentable over *AutoSOCKS* and respectfully traverses the rejection.

AutoSOCKS is substantially similar to *Aventail v3.01*, discussed above. The allegations made and the art cited by the Office Action and the Apple Request in support of the rejections based on *AutoSOCKS* are also substantially similar to the allegations made in view of *Aventail v3.01*. Therefore, Patent Owner incorporates by reference the arguments made in support of the patentability of claims 1-16 over *Aventail v3.01*, and repeats and reaffirms those arguments in support of the patentability of claims 1-16 over *AutoSOCKS*.

Accordingly, Patent Owner respectfully requests that the rejection of claims 1-16 under § 102(b) based on *AutoSOCKS* be withdrawn, and the patentability of these claims be confirmed.

D. The Rejection of Claims 1-16 Under 35 U.S.C. § 102(a) Based on *BinGO* Should Be Withdrawn (Issue 3)

The Office Action rejects claims 1-16 under 35 U.S.C. § 102(a) based on the *BinGO! User's Guide* and *BinGO! Extended Feature Reference* (Apple Req. Ex. X7) ("*BinGO*"). (OA at 28.) For the reasons discussed below, this rejection should be withdrawn and the claims should be confirmed.

1. The *BinGO* User's Guide Has Not Been Shown to Expressly Incorporate *BinGO EFR*

The Office Action and the Apple Request contend that the *BinGO* user's guide expressly incorporates the contents of the Extra Feature Reference ("*BinGO EFR*"). (OA at 28; Apple Req. at

12.) This is incorrect. Initially, Patent Owner notes that the *BinGO* user's guide, on both pages 22 and 115, refers to a document titled "Extended *Features* Reference." In contrast, *BinGO EFR* is titled "Extended *Feature* Reference." Because these titles are different from one another, it is not clear that *BinGO EFR* is even the correct document that is referenced in the *BinGO* user's guide. Indeed, in the Petition in Opposition to Patent Owner's Petition to Vacate Reexamination *Inter Partes* Reexamination Determination on Certain Prior Art (dated November 21, 2011), Apple pointed to the website <http://web.archive.org/web/19990417093944/http://www.bintec.de/eftp/bingo.html> as alleged evidence of the public availability of *BinGO*. However, this website actually provides a link to a document that is titled "Extended *Features* Reference" (available at <http://web.archive.org/web/20030926214344/http://www.bintec.de/download/brick/doku/71050a.pdf>) instead of "Extended *Feature* Reference."

Furthermore, even if the *BinGO* user's guide did reference *BinGO EFR*, nowhere does the *BinGO* user's guide mention which specific version of *BinGO EFR* is being referenced. *BinGO EFR*, on its first page, explains that it is "Version 1.2," which implies that there are other versions of *BinGO EFR* available. But the "Extended Features Reference" indicates that it is "Version 1.5" on page 3. Because the *BinGO* user's guide does not indicate which version of *BinGO EFR* it references, and *BinGO EFR* and the "Extended Features Reference" indicate different version numbers, the *BinGO EFR* cited by the Office Action and the Apple Request (i.e., version 1.2) cannot be considered to be expressly incorporated into the *BinGO* user's guide and should be considered a separate document. Accordingly, Patent Owner submits that the § 102(a) rejection of claims 1-16 based on both the *BinGO* user's guide and *BinGO EFR* is improper. Reconsideration and withdrawal of this rejection are therefore respectfully requested.

Nevertheless, even assuming the *BinGO EFR* cited by the Office Action and the Apple Request (i.e., version 1.2) is incorporated into the *BinGO* user's guide, claims 1-16 are patentable over both the *BinGO* user's guide and *BinGO EFR* for the reasons set forth below.

2. Overview of *BinGO*

BinGO is a user's guide for a router product used to route information either to the Internet or to a different destination, such as a corporate network. (*BinGO* 13) *BinGO* states that if a user wants to access the Internet, the user must set up the user's Internet service provider (ISP) as a wide area network (WAN) partner on the *BinGO!* router, and if the user wishes to establish a LAN-to-LAN connection (e.g., between the user's LAN and the LAN of a corporate office), the user must configure the LAN of the corporate office as a WAN partner. (*BinGO* 143) In an alternative

Internet-access configuration, instead of setting up an ISP as a WAN partner on the BinGO! router, a user may attempt to access the Internet via another WAN partner's connection to an ISP, for example, via a corporate network's ISP connection. (*BinGO* 90)

BinGO explains that several options are available for domain name resolution: (1) sending requests to a DNS server; (2) employing a BinGO! router as a DNS proxy server; (3) employing a WINS server; and (4) using HOSTS and LMHOSTS files on the user's PC. (*BinGO* 87; Keromytis Decl. ¶ 34.) If the BinGO! router is employed as a DNS proxy server, *BinGO* explains that it will handle domain name server (DNS) requests such that it will first attempt to resolve a request at a primary DNS server, and then if that fails, will send the request to a secondary DNS server. (See, e.g., *BinGO* 87-88; Keromytis Decl. ¶ 34.) These default routes are manually configured by a BinGO! router user. (*BinGO* 199-202; Keromytis Decl. ¶ 34.) In configuring the BinGO! router, encryption for a connection to a WAN partner may be selected. (*Bingo* 149-50, 175; Keromytis Decl. ¶ 34.)

BinGO EFR also describes that for certain BinTec-brand routers—specifically, BRICK routers—a connection may be established between a client and a remote destination via an ISP. (*BinGO EFR* 82-84; Keromytis Decl. ¶ 35.) In particular, *BinGO EFR* describes two scenarios for establishing this alleged connection: a PPTP Client-to-Server scenario and a LAN-to-LAN scenario. (See *BinGO EFR* 83-84; Keromytis Decl. ¶ 35.) Under either scenario, *BinGO EFR* describes that a connection is established to the local ISP first, and then the alleged connection is established over the Internet. (*BinGO EFR* 83-84; Keromytis Decl. ¶ 35.) Although *BinGO EFR* explicitly mentions that other features are available for certain BinGO routers as well as for various BRICK routers (see, e.g., *BinGO EFR* 191), *BinGO EFR* does not identify any of these features as being available for BinGO! routers. (Keromytis Decl. ¶ 35; see also *BinGO EFR* 2, “Depending on your particular product some of the features described in this document may not be available on your system.”)

3. The Office Action and the Apple Request Rely on Two Alternative Embodiments of *BinGO*

The Office Action and the Apple Request rely on two separate, alternative embodiments of *BinGO* in rejecting claims 1-16, referred to as the “ISP Configuration” and the “Non-ISP Configuration” in the following discussion. In the “ISP Configuration,” the BinGO! router is connected to the Internet via an Internet service provider (ISP), and also connected to a local area network (LAN) such as a corporate network, as shown below. (*BinGO* 15; Keromytis Decl. ¶ 36.) In this configuration, the BinGO! router takes measures to resolve requests generated by a user. (*BinGO* 87-90; Keromytis Decl. ¶ 36.)

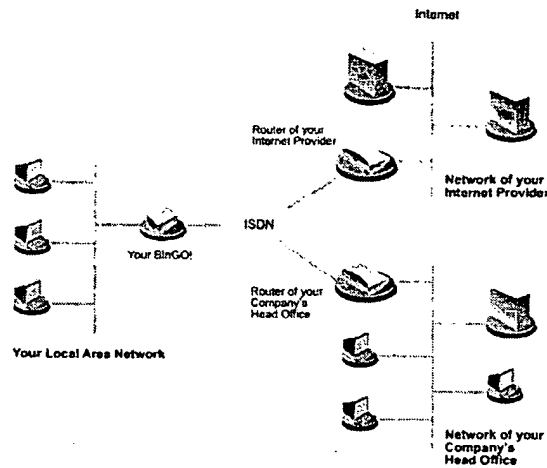


Figure 1-1: Basic scenario

(BinGO 15.)

In the “Non-ISP Configuration,” the BinGO! router does not have a direct ISDN connection to an ISP, and therefore cannot access an ISP unless it does so via a wide area network (WAN) partner. (Keromytis Decl. ¶ 37; see BinGO 90, “If you have not configured Internet access, but your head office has an Internet Service Provider, you can access the Internet via the provider of your WAN partner.”) In this configuration, the BinGO! router simply forwards requests according to a default route to the corporate router, and the corporate router then takes measures to resolve the requests:

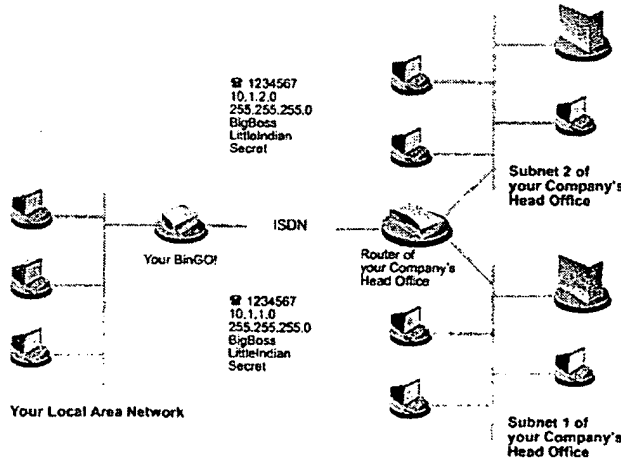


Figure 4-3: Scenario: WAN partner with two subnets

(BinGO 90-92, figure illustrating the absence of a direct ISDN connection to an ISP; Keromytis Decl. ¶ 37.)

Because the Office Action and the Request mix and match these two distinct embodiments in its analysis of the claims, the § 102(b) rejection is improper. M.P.E.P. § 2131. Unless BinGO

“discloses within the four corners of the document not only all of the limitations claimed but also all of the limitations arranged or combined in the same way as recited in the claim, it . . . cannot anticipate under 35 U.S.C. § 102.” *Net MoneyIN*, 545 F.3d at 1371. Thus, the Office Action and the Request may not rely on the ISP Configuration for one element of a claim, and on the alternative Non-ISP Configuration for another element, as “[t]he requirement that the prior art elements themselves be ‘arranged as in the claim’ means that claims cannot be ‘treated . . . as mere catalogs of separate parts, in disregard of the part-to-part relationships set forth in the claims and that give the claims their meaning.’” *Therasense*, 593 F.3d at 1332.

Nevertheless, as demonstrated below, neither the ISP Configuration nor the Non-ISP Configuration discloses the features recited in claims 1-16. Therefore, the rejections of the claims should be withdrawn, and their patentability confirmed.

4. Independent Claim 1

Independent claim 1 is directed to a data processing device storing a domain name server (DNS) proxy module. *BinGO* fails to disclose the combination of features recited in this claim for at least the reasons discussed below.

a. *BinGO* Fails to Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

Independent claim 1 recites “[a] data processing device, comprising memory storing a domain name server (DNS) proxy module that . . . performs the step[] of . . . determining whether the intercepted DNS request corresponds to a secure server.” *BinGO* does not disclose this feature.

The Office Action and the Apple Request assert that *BinGO*, in the ISP Configuration, discloses the above “determining” step of claim 1 because *BinGO* indicates that (1) a user could set up a local DNS server to resolve requests for IP addresses for requested destinations on the local corporate network; and (2) the user could configure the BinGO! router to use primary and secondary DNS servers to route requests either locally or to the ISP, depending on whether an IP address for a computer on the corporate network has been requested. (OA at 28; Apple Req. at 90-92.) This is incorrect. (Keromytis Decl. ¶ 39.)

First, the Office Action and the Request identify the BinGO! router as corresponding to the “data processing device . . . storing a domain name server (DNS) proxy module” of claim 1. (Apple Req. at 90, “[A] BinGO! router is a data processing device comprising memory and . . . the BinGO! router could function as a DNS proxy server.”) But the Office Action and the Request appear to admit that the BinGO! router does not perform the recited “determining” step at all; rather, they

allege that a separate DNS server performs the “determining” step. (*Id.* at 91, “[A]fter receiving a DNS request from a client computer, the BinGO! router *would use the local DNS server . . . to determine if the DNS request specified a secure server,*” emphasis added.) Thus, assuming arguendo that the Request’s interpretation of *BinGO* is correct, it does not show that the alleged DNS proxy module—the BinGO! router—performs the alleged “determining,” because it instead points to the DNS server. Thus, the alleged DNS proxy module in the ISP Configuration of *BinGO* has not been shown to perform the “determining” step of claim 1.

Next, the BinGO! router does not determine whether a DNS request corresponds to a secure server. As shown in the table below, the BinGO! router simply identifies where to refer the query. (*BinGO* 87; Keromytis Decl. ¶ 40.)

Field	Description
Primary Domain Name Server	IP address of BinGO!'s first Domain Name Server (DNS).
Secondary Domain Name Server	IP address of another Domain Name Server.
Primary WINS	IP address of BinGO!'s first WINS (Windows Internet Name Server) or NBNS (NetBIOS Name Server).
Secondary WINS	IP address of another WINS or NBNS.

Table 7-15: IP ► STATIC SETTINGS

(*BinGO* 200.) As *BinGO* explains, “[a]s soon as the primary DNS receives a request, it tries to translate the name. If it is not able to translate the name, it refers the request to the next higher DNS.” (*Id.* at 87, 200.) These primary and secondary DNS servers are manually configured:

DNS in the LAN If you have set up a DNS in your LAN, enter its IP address.

To Do Proceed as follows, if you have not made this entry already (chapter 7.3.2, page 214):

- Go to IP ► STATIC SETTINGS.
- Enter Primary or Secondary Domain Name Server, if applicable.
- Enter Primary or Secondary WINS, if applicable.
- Press SAVE.

(*Id.* at 201.) Accordingly, the BinGO! router simply refers the query to the next domain name server in the sequence without determining whether it corresponds to a “secure server.” (Keromytis Decl. ¶ 40.)

The Request recognizes this shortcoming of *BinGO*, arguing that *BinGO* discloses the “determining” step of claim 1 because the BinGO! router would allegedly send the query first to the local DNS server, and then “[i]f this DNS server did not resolve the address . . . , the BinGO! router would send the request to a secondary DNS server (e.g., one associated with an ISP designated to be the ‘default route’ in the BinGO! router configuration settings).” (Apple Req. at 91-92.) But merely relabeling the primary DNS server as the “local DNS server” does not change the functioning of the BinGO! router disclosed in *BinGO*. (Keromytis Decl. ¶ 41.) Thus, because *BinGO* does not disclose determining “whether the intercepted DNS request corresponds to a secure server,” as recited in claim 1, the § 102(a) rejection of claim 1 should be withdrawn and the claim confirmed.

The Office Action and the Apple Request also assert that *BinGO*, in the Non-ISP Configuration, discloses the “determining” step of claim 1 because all DNS requests that could not be resolved locally would be routed to a DNS server on a corporate network, where the determination would be made if the request was specifying a secure destination or a nonsecure destination. (OA at 28; Apple Req. at 92-93.) This is also incorrect. (Keromytis Decl. ¶ 42.)

Having earlier identified the BinGO! router as the alleged DNS proxy module (Apple Req. at 90), the Request now asserts that a separate component—“a DNS server on a corporate network”—performs the alleged “determining” step recited claim 1 (*id.* at 92, “[A]ll DNS requests that could not be resolved locally . . . would be routed to a DNS server on a corporate network, where the determination would be made.”). Mixing and matching among alternative embodiments to meet the elements of claim 1 is inappropriate to support a § 102 rejection. M.P.E.P. § 2131; *see also Net MoneyIN*, 545 F.3d at 1371. Nevertheless, the Non-ISP Configuration also fails to disclose “determining whether the intercepted DNS request corresponds to a secure server,” as recited in claim 1.

BinGO does not disclose any DNS server on a corporate network that determines “whether the intercepted DNS request corresponds to a secure server,” nor does the Office Action or the Request identify any *BinGO* passage suggesting otherwise. (Keromytis Decl. ¶ 43.) Rather, as quoted by the Request, *BinGO* merely explains that a client may “access the Internet via the provider of your WAN partner,” and that “[d]ue to the fact that your default route leads all unknown packets to your head office, and there another default route in turn sends all unknown packets to its Internet provider, you can access the Internet via your partner’s network.” (*BinGO* 90; *see also* Apple Req. at 92, citing *BinGO* 90.) But this describes nothing about how or whether the DNS server on a

corporate network determines whether an intercepted DNS request corresponds to a secure server. (Keromytis Decl. ¶ 43.)

Furthermore, the remaining *BinGO* passages and figures cited in the Request do not support the assertion that either the BinGO! router or the corporate router performs the “determining” step recited in claim 1. (Apple Req. at 92-93, citing *BinGO* 199-202; Keromytis Decl. ¶ 44.) Rather, the cited portions merely disclose the DNS functions already discussed above with respect to the ISP Configuration, which apply to the BinGO! router—not to the separate DNS server at the corporate network alleged to correspond to the DNS proxy module of claim 1 in the Non-ISP Configuration. (*BinGO* 199-202; Keromytis Decl. ¶ 44.) Indeed, the mere fact that a user may configure the BinGO! router to send “all DNS and WINS requests” to the WAN partner under the Dynamic Client feature (*see* Apple Req. at 93-94) does not indicate whether the DNS server on the corporate network—much less the BinGO! router itself—determines whether an intercepted DNS request corresponds to a secure server (*BinGO* 199-202; Keromytis Decl. ¶ 44).

Thus, for at least the reasons provided above, *BinGO* does not disclose the “determining” step recited in claim 1. Accordingly, Patent Owner requests that the § 102(a) rejection of claim 1 be withdrawn, and its patentability confirmed.

b. *BinGO* Fails to Disclose “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

Independent claim 1 recites “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.” *BinGO* does not disclose this feature.

As discussed above, *BinGO* fails to disclose “determining whether the intercepted DNS request corresponds to a secure server” under either the ISP Configuration or the Non-ISP Configuration. Therefore, *BinGO* cannot disclose that a DNS proxy module (whether alleged to be the BinGO! router or a corporate router) takes any action, much less “automatically initiating an encrypted channel,” when the intercepted DNS request corresponds to a secure server. Accordingly, *BinGO* does not disclose the “automatically initiating an encrypted channel” feature of claim 1.

In support of the rejection for this feature of claim 1, the Office Action and the Request refer back to the earlier overview discussion of *BinGO* in the Request. (OA at 28; Apple Req. at 95, also citing generally *BinGO EFR* 73-98.) This analysis is defective, however, as it fails to illustrate how *BinGO* allegedly discloses each and every aspect of the element of the claim. For example, it is

unclear what is alleged, among other things, to correspond to the “secure server,” how the alleged encrypted channel is “automatically initiated,” and how the alleged channel is “encrypted.” (*See* Apple Req. at 95.) The Request also fails to even analyze the correct claim language at issue, instead addressing different claim language that cannot be used to carry a rejection of the ’151 claims. (*See, e.g., id.*, “automatically initiating the VPN,” “requesting access to a secure target website,” etc., emphasis added.) Because the Office Action and the Request have failed to specifically show with respect to *BinGO* “all of the limitations arranged or combined in the same way as recited in the claim, it . . . cannot anticipate under 35 U.S.C. § 102,” and the rejection should be withdrawn. *Net MoneyIN*, 545 F.3d at 1371.

Furthermore, despite these deficiencies, the *BinGO* and *BinGO EFR* passages broadly cited in the general *BinGO* overview section of the Request fail to disclose “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” as recited in claim 1. (*See* Apple Req. at 82-90.) For example, *BinGO* discloses that any encryption measures for communicating with a WAN partner are established as part of a manual WAN partner configuration. (*BinGO* 145-50; Keromytis Decl. ¶ 46.) But nowhere does *BinGO* explain how or whether this encryption occurs in conjunction with a DNS request, let alone with intercepting a DNS request or determining whether a DNS request corresponds to a secure server. (Keromytis Decl. ¶ 46.)

Rather, *BinGO* merely discloses that its limited encryption feature “[d]efines the type of encryption that should be used for data traffic with the WAN partner,” providing no guidance on what steps occur before the alleged “automatically initiating an encrypted channel” occurs. (*BinGO* 149; Keromytis Decl. ¶ 46.) Thus, the encryption feature of *BinGO* may come into effect based on entirely different criteria than those recited in claim 1. (Keromytis Decl. ¶ 46.) For example, a BinGO! router might initiate its encryption based on the establishment of a connection, not “when the intercepted DNS request corresponds to a secure server.” (*Id.*) As another example, the BinGO! router might require manual initiation of encryption instead of “automatically initiating” encryption. (*Id.*) It is impossible to know because *BinGO* simply does not disclose how its encryption feature operates. (*Id.*) Meanwhile, in the Non-ISP Configuration, where all DNS requests are routed through a corporate network, the manual configuration of the corporate network WAN partner might result in all communications being encrypted, whether directed to a computer on the corporate network (i.e., an alleged secure server) or to a computer on the Internet (i.e., an alleged nonsecure server). (*See BinGO* 90, “If you have only configured a partner network and not an Internet provider,

the Wizard simply uses the route to your partner's network as a default route.”; Keromytis Decl. ¶ 46.)

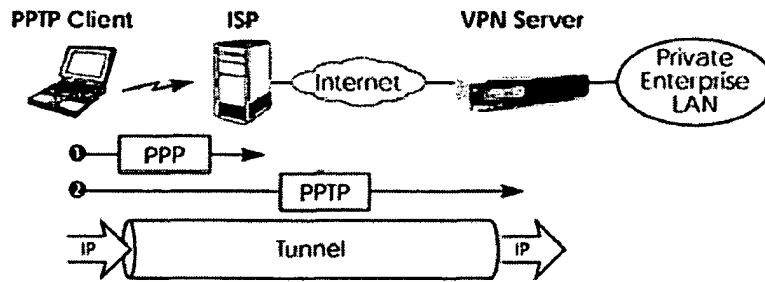
Thus, because *BinGO* has not been shown to explicitly or inherently disclose this feature of claim 1, the § 102(b) rejection is improper and should be withdrawn. *See In re Robertson*, 169 F.3d at 745 (requiring for inherent anticipation that a reference “make clear that the missing descriptive matter is *necessarily* present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill.”) (emphasis added).

The Request also cites in particular to a passage in *BinGO EFR* that fails to discuss the BinGO! router at all. (Apple Req. at 95, citing *BinGO EFR* 73-98.) Here, *BinGO EFR* discusses a connection feature for a BRICK router, not the BinGO! router described in *BinGO*. Indeed, *BinGO EFR* initially indicates on the copyright page that “[t]his manual provides a complete description of all the complex, separately licensable features available for the BinTec BIANCA/BRICK and BinGO! routers” (*BinGO EFR* at copyright page), but then explains that “[d]epending on your particular product some of the features described in this document may not be available on your system” (*id.* at 2). With specific respect to this connection feature, *BinGO EFR* only explains how to configure a BRICK router to enable it. (*See, e.g., id.* at 74-76, 78, 80-81, 90-91, 95-98, all figures labeled as “BRICK Setup Tool” and/or “BinTec Communications AG myrouter”; *see also id.* at 94-96, explaining “Configuration on SupplierNet BRICK” and “Configuration on Central Site BRICK.”) Neither *BinGO* nor *BinGO EFR* discloses that a “BinGO” router (the alleged data processing device) is the same as a BRICK router, nor do they disclose that a “BinGO” router necessarily has the same features as a BRICK router. Nor do the Office Action and Request assert that *BinGO* or *BinGO EFR* discloses such information. In fact, *BinGO EFR* explains that “some of the features described in this document may not be available on your system.” (*Id.* at 2.) *BinGO EFR* explicitly mentions that certain features are available for BinGO Plus/Professional routers as well as various BRICK routers (*see, e.g., id.* at 191), but it does not specify that the connection feature cited by the Request applies to any BinGO! routers. Thus, the cited portions of *BinGO EFR* have not been shown to disclose the “encrypted channel” feature of claim 1, and the rejection should be withdrawn.

Nevertheless, the cited portion of *BinGO EFR* describes two general configurations, neither of which discloses the “encrypted channel” feature of claim 1 even if performed by a BinGO! router instead of the described BRICK router. These are the Client-to-LAN and LAN-to-LAN scenarios. (*Id.* at 82-84.) Under either configuration, *BinGO EFR* discloses that a client must first connect to an

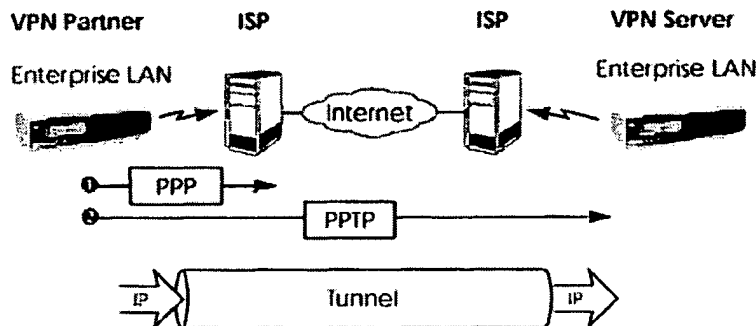
ISP, as illustrated below. In the first configuration, “[t]he remote client (mobile Win95 host) first establishes a standard PPP connection to a local ISP.” (*Id.*; *see also id.* at 86, instructing a user configuring a Client-to-LAN setup to “[s]pecify a name for the ISP this host will be using.”) In the second configuration, *BinGO EFR* explains that either end of the connection “may initiate a standard PPP link to a local ISP.” (*Id.*; *see also id.* at 94.)

Scenario 1. PPTP Client-to-VPN Server



(*Id.* at 83.)

Scenario 2. LAN-to-LAN VPN



(*Id.* at 84.)

But with respect to the “determining” step of claim 1, the Office Action and the Request assert that a client would connect to an ISP *only if the request did not specify a secure destination*, i.e., did not specify a “computer[] on a corporate network.” (OA at 28; *see, e.g.*, Apple Req. at 90-92.) Indeed, the Request contends that “[i]f [the local] DNS server did not resolve the address (*i.e., because the request did not specify a secure destination*), the BinGO! router would send the request to a secondary DNS server (*e.g., one associated with an ISP . . .*).” (Apple Req. at 91-92, emphasis added, distinguishing “a computer on the corporate network” from “a public web site on the Internet.”)

Thus, after asserting that a client would only connect to an ISP if a DNS request specified a *nonsecure server*, the Request now changes course and directly contradicts its “determining” step

arguments by asserting that a client would in fact connect to an ISP if the DNS request specified a secure server. Accordingly, the cited feature of *BinGO EFR* is irreconcilable with the Request's arguments concerning the "determining" step of claim 1, and, as a result, one or both of the "determining" or the "encrypted channel" steps of claim 1 are necessarily absent from *BinGO*. (Keromytis Decl. ¶¶ 47-48.) Thus, the § 102(a) rejection of claim 1 should be withdrawn. *Therasense*, 593 F.3d at 1332 ("[C]laims cannot be 'treated . . . as mere catalogs of separate parts, in disregard of the part-to-part relationships set forth in the claims.'").

Finally, the Office Action and the Request have failed to demonstrate that the alleged encrypted channel is "automatically initiated" at all, much less automatically initiated based on the criterion of "when the intercepted DNS request corresponds to a secure server." (Keromytis Decl. ¶ 49.) For this claim feature, the Office Action and the Request again cite to the earlier general discussion of *BinGO*, and broadly assert that the "BinGO! router automatically establishes an encrypted ISDN connection after authentication between client computers on a LAN with the BinGO! router and destination computers inside a secure corporate network." (OA at 28; Apple Req. at 95.) This is incorrect. (Keromytis Decl. ¶ 49.)

Within the Request's general overview discussion, it cites to the "automatic dialing" feature of *BinGO*. (Apple Req. at 83-84, citing *BinGO* 17, 41.) But these *BinGO* passages fail to disclose anything about encrypted communications, much less how any alleged encryption would be automatically initiated. (*Id.*, citing *BinGO* 17, 41.) Moreover, when connecting to a corporate network (i.e., to an alleged "secure" destination), *BinGO* specifies that authentication must occur before every connection. (*BinGO* 40; Keromytis Decl. ¶ 50.) "This authentication is based on a common password and two codes that you and your partner use for the connection." (*BinGO* 40.) But *BinGO* does not disclose how such additional steps would be incorporated into any "automatic dialing" feature. (*See id.* at 17, 40.) Accordingly, by requiring users to enter authentication credentials before any connection is established, *BinGO* illustrates that connecting to a corporate network is not necessarily "automatic." (Keromytis Decl. ¶ 50.) Indeed, the Request itself recognizes that *BinGO* involves a *manual* authentication process: "You can log in to BinGO! in several different ways . . . but logging in is always protected by a password. Every failed attempt is logged by a syslog message indicating the source and creates a relevant SNMP trap." (Apple Req. at 95, quoting *BinGO* 240.) By comparison, such credentials (and the resulting extra steps for connecting) are not required for accessing the Internet via an ISP (i.e., an alleged *nonsecure* destination). (*BinGO* 39-40; Keromytis Decl. ¶ 50.) Thus, because *BinGO* fails to explain how

encryption would occur with any “automatic dialing,” and the Request fails to explain how that is even possible, *BinGO* does not disclose the “automatically initiating an encrypted channel” feature of claim 1.

Thus, for all of these reasons, Patent Owner requests that the § 102(a) rejection of claim 1 be withdrawn, and its patentability confirmed.

5. Independent Claims 7 and 13

Independent claim 7 recites features similar to those described above for claim 1. For example, claim 7’s recited feature of “determining whether the intercepted DNS request corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Also, claim 7’s recited feature of “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” is similar to the “automatically initiating” feature of claim 1, also discussed above. Accordingly, *BinGO* does not disclose these features of claim 7 for similar reasons as those discussed above with respect to claim 1.

Independent claim 13 also recites features similar to those described above for claim 1. For example, claim 13’s recited feature of “determining whether a DNS request sent by a client corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Independent claim 13 also recites features that differ from the features recited in claim 1 and these features are not even addressed by the Apple Request or the Office Action. For example, claim 13 recites “when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel” The Apple Request, however, ignores this difference in claim language and instead quotes another portion of independent claim 1 when purporting to reject claim 13. (*See, e.g.,* Apple Req. at 109-14.) By ignoring the language of claim 13 and instead analyzing a feature of claim 1, the rejection of claim 13 in view of *BinGO* is improper for failing to consider all of the words in the claim. M.P.E.P. § 2131; *see also id.* at § 2143.04 (“*All words* in a claim must be considered in judging the patentability of that claim against the prior art.”) (emphasis added) (internal citations omitted). Moreover, to the extent the Requester and the Office later assert that the features recited in claim 13 are similar to the features recited in claim 1, Patent Owner asserts that *BinGO* does not disclose these features of claim 13 for similar reasons as those discussed above with respect to claim 1.

Thus, for these reasons, Patent Owner requests that the rejection of claims 7 and 13 under § 102(a) be withdrawn, and their patentability confirmed.

6. Dependent Claims 2, 8, and 14

Dependent claims 2, 8, and 14 depend from independent claims 1, 7, and 13, respectively, and include all of their features. Thus, claims 2, 8, and 14 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 2, 8, and 14 also distinguish over *BinGO* for additional reasons. For example, claims 2, 8, and 14 recite “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.” *BinGO* does not disclose this feature.

BinGO does not disclose “sending a request to the secure server to establish an encrypted channel,” as recited in step (b) of claims 2, 8, and 14. Nor do the Office Action and the Request assert that it does. (See OA at 28; Apple Req. at 95-96.) The Request only addresses whether *BinGO* discloses the step of “determining whether the client is authorized to access the secure server” recited in step (a) of claim 2, primarily citing *BinGO* passages describing authorization for a user to access the BinGO! router itself—not any alleged “secure server.” (Apple Req. at 95-96.) Thus, the rejection of claims 2, 8, and 14 should be withdrawn, and their patentability confirmed.

Indeed, neither *BinGO* nor *BinGO EFR* discloses any request relating to any encryption, much less an “encrypted channel.” (Keromytis Decl. ¶ 52.) *BinGO* briefly discusses encryption with respect to manually configuring the BinGO! router, as shown below, but does not indicate how or whether this encryption feature involves “sending a request to the secure server.” (See *BinGO* 149; Keromytis Decl. ¶ 52.)

<i>Encryption</i>	Defines the type of encryption that should be used for data traffic with the WAN partner. Only possible if STAC compression is not activated for the connection. Possible values: <ul style="list-style-type: none">■ MPPE 40: only when Encapsulation = PPP■ MPPE 128: only when Encapsulation = PPP and Authentication = MS-CHAP■ none:
-------------------	---

(*BinGO* 149.) Neither does *BinGO EFR* disclose any particular “request” relating to its features cited by the Request. (See, e.g., *BinGO EFR* 82-84; Keromytis Decl. ¶ 52.)

Finally, neither is “sending a request to the secure server to establish an encrypted channel” inherent in *BinGO* or *BinGO EFR*. (Keromytis Decl. ¶ 52.) There is simply no reason why the features disclosed in these references would *necessarily* require employing the recited “request.” (*Id.*) For example, the alleged encryption in *BinGO* could be preset during the manual configuration of the *BinGO* router, without requiring sending a specific request to activate it. (*Id.*) Thus, because *BinGO* and *BinGO EFR* do not “make clear that the missing descriptive matter is *necessarily* present

in the thing described in the reference, and that it would be so recognized by persons of ordinary skill,” the rejection of claims 2, 8, and 14 is improper and should be withdrawn. *In re Robertson*, 169 F.3d at 745 (emphasis added).

Furthermore, the Request asserts that *manual* login procedures disclose “determining whether the client is authorized to access the secure server,” as recited in claim 2. (Apple Req. at 95-96, citing *BinGO* 130 (ISDN login), 240 (access security); Keromytis Decl. ¶ 53.) But this “determining” step is included within the “automatically initiating an encrypted channel” feature of claim 1, as claim 2 depends upon claim 1, step (iii). Thus, because the *BinGO* features cited as corresponding to “determining whether the client is authorized” are *manual* login features, it would be impossible for such features to occur as part of “automatically initiating an encrypted channel,” as recited in claim 1. (Keromytis Decl. ¶ 53.)

Thus, for all of these reasons, Patent Owner requests that the § 102(a) rejection of claims 2, 8, and 14 be withdrawn, and their patentability confirmed.

7. Dependent Claims 3, 9, and 15

Claims 3, 9, and 15 depend from one or more of claims 1, 2, 7, 8, 13, and 14, and include all of their features. Thus, *BinGO* does not anticipate claims 3, 9, and 15, and the rejection of these claims should be withdrawn for at least the reasons discussed above with respect to those claims. Claims 3, 9, and 15 also distinguish over *BinGO* for additional reasons. For example, claims 3, 9, and 15 recite “when the client is not authorized to access the secure server, returning a host unknown error message to the client.” *BinGO* does not disclose at least these features.

The Office Action and the Apple Request admit by relying on an inherency argument that *BinGO* does not explicitly disclose “returning a host unknown error message to the client.” (See OA at 28; Apple Req. at 96-97.) But in support of its inherency argument, the Request merely cites to a second reference that also does not disclose returning any error message at all, much less returning a “host unknown error message to the client.” Indeed, the Request quotes RFC 1994 (Exhibit Y11), which merely explains that if CHAP authentication fails, a CHAP packet with the code field set to indicate failure must be sent. (Apple Req. at 97, quoting RFC 1994 at 8-9.) But neither the Request nor RFC 1994 explains how this CHAP packet corresponds to a “host unknown error message” or how it is returned “to the client” within the scope of claims 3, 9, and 15.

Instead, the Request asserts, without support, that “[o]nce terminated, the TCP/IP response will return an error as well, which *commonly* is the host unknown error message.” (*Id.*, emphasis added.) But other error messages may also result from a CHAP authentication failure, as the

Request's "commonly" qualifier appears to admit. (Keromytis Decl. ¶ 55.) Thus, the Office Action and the Request have fallen far short of establishing that the missing descriptive matter alleged to be inherent "is *necessarily* present in the thing described in the reference." *In re Robertson*, 169 F.3d at 745. Moreover, it is inappropriate to rely solely on interpretation or "common knowledge" in the art without evidentiary support in the record. M.P.E.P. § 2144.03. Thus, Patent Owner respectfully submits that the Office Action and the Request may not properly make the unsupported assertion that "when the client is not authorized to access the secure server, returning a host unknown error message to the client" would be inherent in *BinGO*.

Thus, for these reasons, Patent Owner requests that the § 102(a) rejection of claims 3, 9, and 15 be withdrawn, and their patentability confirmed.

8. Dependent Claims 4, 10, and 16

Dependent claims 4, 10, and 16 depend from one or more of claims 1-3, 7-9, and 13-15, and include all of their features. Thus, *BinGO* does not anticipate any of these claims for at least the reasons discussed above in conjunction with claims 1-3, 7-9, and 13-15. Thus, the rejection of claims 4, 10, and 16 under 35 U.S.C. § 102(a) based on *BinGO* should be withdrawn, and their patentability confirmed.

9. Dependent Claims 5 and 11

Dependent claims 5 and 11 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, claims 5 and 11 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 5 and 11 also distinguish over *BinGO* for additional reasons. For example, claims 5 and 11 recite that the feature of automatically initiating the encrypted channel between the client and the secure server "comprises establishing an [IP] address hopping scheme between the client and the secure server." *BinGO* does not disclose this feature.

The Office Action and the Apple Request point generally to the NAT protocol disclosed by *BinGO* and conclude, without any support, that it "creat[es] an IP hopping scheme between the client and destination computers." (OA at 28; Apple Req. at 97-98.) But "when the PTO asserts that there is an explicit or implicit teaching or suggestion in the prior art, it must indicate *where* such a teaching or suggestion appears in the prior art." *In re Rijckaert*, 9 F.3d at 1533 (emphasis added). *BinGO* merely discloses that its NAT features (1) hide the internal host addresses, (2) control external-to-internal access, (3) ensure that connection partners use only a single IP address, and (4) provides permanent monitoring of connections. (*BinGO* 244-46.) It is unclear from the Request how this discloses any IP hopping scheme, especially considering that *BinGO*'s NAT features "ensure[] that a

connection partner uses only a single IP address.” (*Id.* at 245.) Consequently, since the Request has not articulated any reasoning whatsoever to support its conclusion that the referenced “NAT protocol is an IP hopping scheme within the meaning of claim 5,” anticipation has not been shown, and the rejection should be withdrawn.

The Request also cites to *BinGO EFR*, asserting that the “Open Shortest Path First” (“OSPF”) feature discloses an IP hopping scheme. (Apple Req. at 98, citing *BinGO EFR* 17.) But regardless of whether OSPF corresponds to an IP hopping scheme, *BinGO EFR* fails to disclose that its OSPF feature is available for *BinGO!* routers, the alleged data processing device. Rather, it only discloses that its OSPF feature applies to BRICK routers. In the OSPF section, *BinGO EFR* explains that it will provide “an example OSPF installation using different BinTec routers” (*BinGO EFR* 5), but then only provides examples for the following routers: BRICK-XL, BRICK-XS, BRICK-XM, and ALL BRICKs (*see id.* at 28-42). Indeed, the figures in the OSPF section specify: “BRICK Setup Tool.” (*See, e.g., id.* at 6-9, 12-14.)

Thus, the rejection of claims 5 and 11 should be withdrawn because the OSPF feature as disclosed in *BinGO EFR* has not been shown to apply to any *BinGO!* router. Neither *BinGO* nor *BinGO EFR* discloses that a *BinGO!* router is the same as a BRICK router, nor do they disclose that a *BinGO!* router necessarily has the same features as a BRICK router. Nor do the Office Action and the Request assert that *BinGO* or *BinGO EFR* discloses such information. In fact, *BinGO EFR* explains that “some of the features described in this document may not be available on your system.” (*Id.* at 2.) Indeed, *BinGO EFR* explicitly mentions that certain features are available for various *BinGO!* routers as well as various BRICK routers (*see, e.g., id.* at 191), but it does not specify that the disclosed OSPF feature applies to any *BinGO!* routers. Thus, the OSPF feature of *BinGO EFR* has not been shown to correspond to the “IP hopping scheme” feature of claims 5 and 11, and the rejection should be withdrawn.

For these reasons, Patent Owner requests that the § 102(a) rejection of claims 5 and 11 be withdrawn, and their patentability confirmed.

10. Dependent Claims 6 and 12

Dependent claims 6 and 12 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, claims 6 and 12 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 6 and 12 also distinguish over *BinGO* for additional reasons. For example, claims 6 and 12 recite that the feature of automatically initiating the

encrypted channel between the client and the secure server “avoids sending a true IP address of the secure server to the client.” *BinGO* does not disclose this feature.

The Office Action and the Request assert that in the LAN-to-LAN configuration disclosed in *BinGO EFR*, the client computers on each LAN communicate only with a BinGO! router and therefore do not receive the IP address of the remote host. (OA at 28; Apple Req. at 98-99.) This is incorrect.

As discussed above, this feature of *BinGO EFR* has not been shown to correspond to the “encrypted channel” in the claims, given that *BinGO EFR* only discloses it in conjunction with BRICK routers. Indeed, in the excerpt and figure quoted in the Request, *BinGO EFR* specifies “using two BRICKs as follows.” (Apple Req. at 99, quoting *BinGO EFR* 94, emphasis added.) Thus, because the configuration alleged to “avoid[] sending a true IP address of the secure server to the client” has not been shown to apply to the BinGO! router, the alleged data processing device, the rejection should be withdrawn and claims 6 and 12 should be confirmed.

Furthermore, in the LAN-to-LAN configuration illustrated above, the alleged “encrypted channel” requires connecting to an ISP. (*BinGO EFR* 94.) But with respect to the “determining” step incorporated into dependent claims 6 and 12 via independent claims 1 and 7, the Office Action and the Request assert that a client would connect to an ISP *only if the request did not specify a secure destination*. (OA at 28; *see, e.g.*, Apple Req. at 90-92.) Indeed, the Apple Request explains that “[i]f [the local] DNS server did not resolve the address (*i.e., because the request did not specify a secure destination*), the BinGO! router would send the request to a secondary DNS server (*e.g., one associated with an ISP . . .*).” (Apple Req. at 91-92, emphases added, distinguishing “a computer on the corporate network” from “a public web site on the Internet.”) Thus, because the Office Action and the Request mix and match inconsistent and incompatible embodiments of *BinGO* to attempt to satisfy the claim language of claims 6 and 12, they have failed to demonstrate that *BinGO* discloses “all of the limitations arranged or combined in the same way as recited in the claim.” *Net MoneyIN*, 545 F.3d at 1371. Indeed, “[t]he requirement that the prior art elements themselves be ‘arranged as in the claim’ means that claims cannot be ‘treated . . . as mere catalogs of separate parts, in disregard of the part-to-part relationships set forth in the claims and that give the claims their meaning.’” *Therasense*, 593 F.3d at 1332. Thus, the rejection of claims 6 and 12 is improper, and the claims should be confirmed.

The Request also asserts, without any apparent support, that because a client computer on a LAN in *BinGO EFR* would allegedly only communicate with a proxy router, it would not receive the

IP address of the remote host. (Apple Req. at 99.) But the Request appears to admit that *BinGO* does not explicitly disclose “avoid[ing] sending a true IP address of the secure server to the client,” as the Request simply cites to the general discussion of encryption in *BinGO* and the mere fact that in the connection scenarios discussed earlier, “data encryption/decryption is performed at each end of the tunnel.” (*Id.*) Yet, the mere use of encryption does not necessarily require avoiding “sending a true IP address of the secure server to the client,” and, therefore, this feature has not been shown to be inherent in *BinGO*. (Keromytis Decl. ¶ 57.) And the Request fails to specifically explain how the mere use of encryption either explicitly or inherently would avoid sending a true IP address to the client. Indeed, the true IP address could in fact be sent to the client in *BinGO*, for example, in the payload of packets transmitted over the tunnel. (*Id.*) Thus, because the rejection does not “make clear that the missing descriptive matter is *necessarily* present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill,” the rejection is improper and should be withdrawn. *In re Robertson*, 169 F.3d at 745 (emphasis added); *see also* M.P.E.P. § 2112. Thus, the rejection of claims 6 and 12 should be withdrawn, and their patentability should be confirmed.

For at least the foregoing reasons, Patent Owner respectfully requests that the rejection of claims 1-16 under § 102(a) based on *BinGO* be withdrawn, and that the patentability of claims 1-16 be confirmed.

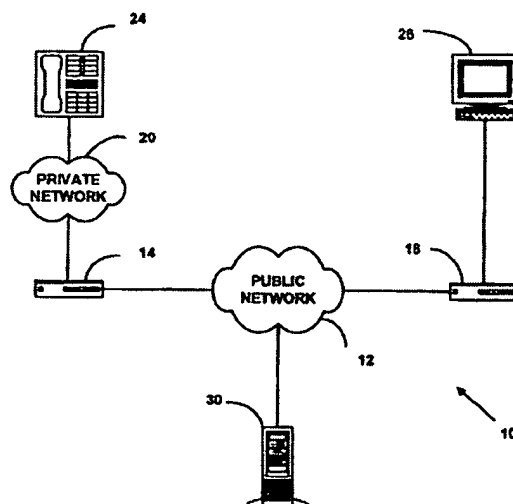
E. The Rejection of Claims 1, 2, 4-8, 10-14, and 16 Under 35 U.S.C. § 103(a) Based on *Beser* in View of *Kent* Should Be Withdrawn (Issue 4)

The Office Action rejects claims 1, 2, 4-8, 10-14, and 16 under 35 U.S.C. § 103(a) based on U.S. Patent No. 6,496,867 to *Beser et al.* (Apple Req. Ex. X5) (“*Beser*”) in view of *Kent*, “Security Architecture for IP,” RFC 2401 (Apple Req. X6) (“*Kent*”). For the reasons discussed below, these rejections should be withdrawn, and the claims should be confirmed.

1. Overview of *Beser*

Beser discloses a system for initiating a tunneling connection that hides the identity of the originating and terminating ends of the tunneling association from other users. (*Beser* Abstract.) With reference to Fig. 1, reproduced below, *Beser* discloses that a trusted-third-party network device 30 is informed of a request to initiate a tunneling connection made by an originating telephony device 24. (*Id.* at 7:62-8:4, 10:2-6, 11:9-10.)

FIG. 1



The request to initiate a tunneling connection includes a unique identifier for a terminating telephony device 26. (*Id.* at 10:4-6.) After being informed of the request, trusted-third-party network device 30 associates an identifier of terminating telephony device 26 with a public IP address of a second network device 16. (*Id.* at 11:26-32.) Then, private IP addresses for each of the originating telephony device 24 and the terminating telephony device 26 are negotiated and distributed to the second network device 16 and the first network device 14, respectively. (*See, e.g., id.* at 11:59-12:54.) This way, the tunneling connection “hides the identity of the originating and terminating ends of the tunneling association from the other users of the public network.” (*Id.* at 2:36-39.)

2. *Beser* Cannot Be Combined with *Kent*

Beser is directed towards “initiating . . . a virtual tunnel,” primarily in the context of voice-over-IP (“VoIP”) communications. (*Id.* at 6:58-59.) But *Beser* does not disclose *encrypting* traffic in its tunneling scheme. Indeed, *Beser* explains that encryption is “infeasible” and/or “inappropriate” in VoIP applications. (*Id.* at 1:58-2:17.)

Kent is a Request for Comments that discloses IPsec, a type of security protocol. (Apple Req. at 122-23.) The Request alleges that *Beser* could be modified with the security protocol disclosed by *Kent* to form the encrypted channel of claim 1. (*Id.* at 129.) Specifically, the Request points to column 1, lines 54-56, of *Beser* in an attempt to support an allegation that “*Beser* . . . shows that the IPsec protocol can and should be used to hand the encryption of the traffic being sent through the IP tunnel.” (*Id.*) The cited portions of *Beser*, however, are directed to the problems with encrypting packets in VoIP applications that do not use IP tunnels, and have nothing to do with IP tunnels. (*See Beser* 1:54-67; Keromytis Decl. ¶ 62.)

In fact, *Beser* specifically teaches against using IPsec and other encryption techniques in tunneling connections and VoIP applications, the technology with which *Beser* is primarily concerned. *Beser* explains that encryption may be “infeasible” for VoIP due to system strain on computing power and increased investment in VoIP equipment, and “inappropriate” for the transmission of multimedia or VoIP packets. (*Id.* at 1:58-2:17; Keromytis Decl. ¶ 63.) For these reasons, *Beser* discloses a system and method directed to initiating a tunneling association, in which IP packets are not encrypted because of the problems with encryption. (*Id.* at 2:36-40.) One of ordinary skill, therefore, would have understood *Beser*’s technique as an alternative to encryption. (Keromytis Decl. ¶ 63.)

Because *Kent* discloses IPsec, the very protocol that *Beser* explicitly teaches as being problematic, a person of ordinary skill in the art at the time of the invention would not have looked to combine these references. (*Id.* at ¶ 64.) Indeed, *Beser*’s disclosed system and method for initiating a tunneling association is intended as an alternative to encryption to address the drawbacks that arise from the teachings of *Kent* (e.g., high computing power), not to encourage the use of encryption. (*Id.*)

“[W]hen a prior art reference teaches away from a combination, that combination is more likely to be nonobvious.” *In re ICON Health & Fitness Inc.*, 496 F.3d 1374, 83 U.S.P.Q.2d 1746 (Fed. Cir. 2007) (citing *KSR Int’l Co. v. Teleflex Inc.*, 126 S. Ct. 1727, 1741 (2007)). The portions of *Beser* teaching against using encryption, including IPsec, because it may be “infeasible” and “inappropriate” cannot be ignored in determining whether claims are prima facie obvious in view of the references. M.P.E.P. § 2141.02(VI). Accordingly, a person of ordinary skill in the art would not have looked to modify *Beser*’s technique of initiating a tunneling association by adding IPsec, a protocol having characteristics that *Beser* specifically teaches not to use. (See Keromytis Decl. ¶¶ 61-64.)³ Moreover, by defining the solution to the problem that would have allegedly been solved by combining *Beser* and *Kent* in terms of the ’151 patent’s claim language, i.e., “an encrypted channel,” (Apple Req. at 129), the Request has infected its obviousness analysis with impermissible hindsight bias. “It is difficult but necessary that the decisionmaker . . . cast the mind back to the time

³ Patent Owner also notes that it is not proper to use the invention to define the problem that the invention solves. *Mintz v. Dietz & Watson, Inc.*, Case No. 2010-1341, at 9 (Fed. Cir. May 30, 2012). The challenger of a patent must provide evidence that a person of ordinary skill in the relevant art at the time of the invention would have recognized the problem recognized by the inventors and found it obvious to produce the claimed invention disclosed to solve that problem. *Id.*

the invention was made . . . to occupy the mind of one skilled in the art.” *W.L. Gore & Assocs., Inc. v. Garlock, Inc.*, 721 F.2d 1540, 1553 (Fed. Cir. 1983).

Thus, for all of these reasons, one of ordinary skill in the art would not have combined *Beser* and *Kent* to render the claims of the ’151 patent obvious.

3. Independent Claim 1

Independent claim 1 is directed to a data processing device storing a DNS proxy module. As explained above, one of ordinary skill in the art would not have combined *Beser* and *Kent*. In addition, even if the references were combined, the combination would fail to disclose or suggest the combination of features recited in this claim for at least the reasons discussed below.

a. The Combination of *Beser* and *Kent* Fails to Disclose “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”

Independent claim 1 recites, among other things, “a domain name server (DNS) proxy module that intercepts DNS requests sent by a client.” The combination of *Beser* and *Kent* does not disclose or suggest this feature.

The Office Action and the Apple Request assert that the trusted-third-party network device of *Beser* corresponds to a “DNS proxy module.” (OA at 28-29; Apple Req. at 128.) *Beser*, however, merely discloses that the trusted-third-party network device may be a “domain name server.” (*Beser* 11:32-36.) Nothing in *Beser* discloses, teaches, or suggests that the trusted-third-party network device of *Beser* may function as a DNS proxy module that intercepts DNS requests sent by a client. (Keromytis Decl. ¶ 65.)

The Request also appears to allege that a DNS request is disclosed by *Beser* because a user “initiates a request by taking an action” and that a domain name is “sent to the trusted third party network device.” (Apple Req. at 128.) But these allegations do not disclose a DNS request. (Keromytis Decl. ¶ 66.) Rather, *Beser* merely discloses a request to initiate a VoIP association. (*Beser* 10:2-3; Keromytis Decl. ¶ 66.) While *Beser* discloses that the request may include a unique identifier (*Beser* 8:1-3, 10:5-6), which may be, in some instances, a domain name (*id.* at 10:41), merely including a domain name in the request to initiate a VoIP association does not transform it into a request for an IP address. (Keromytis Decl. ¶ 66.)

Additionally, the way *Beser* initiates a VoIP association is not disclosed as being “transparent to the user, who simply initiates a request by taking action,” as proposed by the Request. The Request points to portions of *Beser* that describe a request for a VoIP connection may include an electrical signal resulting from a phone going “off-hook.” (Apple Req. at 128, citing *Beser* 10:22-

36.) The Request, however, does not describe how including the “off-hook” electrical signal discloses initiating a request for the VoIP connection including a unique identifier (*Beser* 8:1-3), much less “intercepting” a DNS request. Similarly, *Beser* utterly fails to support the Request’s suggestion that a user of *Beser*’s system would simply “enter[] the website destination of a WebTV device” or otherwise take action that could result in initiating a “DNS request.” (See Apple Req. at 128, citing *Beser* 4:43-54.)

Kent does not remedy these deficiencies of *Beser*. For instance, *Kent* does not disclose or suggest, and the Request and the Office Action do not rely on *Kent* to show, any DNS proxy module or DNS request. (*Id.* at 127-29.) Nor has the Request identified any reasoning that would have led a person of ordinary skill to modify *Beser* alone to include these features. M.P.E.P. § 2141(III)(G).

Thus, for at least the reasons provided above, *Beser* in view of *Kent* does not disclose or suggest “a domain name server (DNS) proxy module that intercepts DNS requests sent by a client,” as recited in claim 1. Accordingly, the § 103(a) rejection of claim 1 should be withdrawn, and its patentability should be confirmed.

**b. The Combination of *Beser* and *Kent* Fails to Disclose
“Determining Whether the Intercepted DNS Request
Corresponds to a Secure Server”**

Independent claim 1 recites, among other things, “determining whether the intercepted DNS request corresponds to a secure server.” The combination of *Beser* and *Kent* does not disclose or suggest this feature.

The Office Action asserts that “an edge router or a network device behind an edge router that communicates through an authenticated and encrypted channel is reasonably construed as a ‘secure server.’” (OA at 28-29.) But the Office Action and the Request do not identify how either reference, alone or in combination, performs the step of “determining” whether an intercepted request corresponds to this alleged secure server. Rather, for these claim features, the Request relies on *Beser* alone, which does not disclose or suggest the features for the reasons discussed below. (Apple Req. at 129, discussing only *Beser*.)

The Request points generally to a portion of *Beser* describing that IP addresses between a first and second network device may be negotiated if an E.164 number is found in a directory database on the trusted-third-party network device. (*Id.*, citing *Beser* 11:45-59.) The Request then asserts that *Beser* discloses making a determination of whether a domain name is specifying a secure destination server because “under the inherent operation of [*Beser*’s] process, if a domain name . . .

specifies a destination that is unknown to the third-party-network device, it will not route the request further.” (*Id.*) This is incorrect.

Comparing unique identifiers (e.g., an E.164 number) to a directory database in *Beser* fails to indicate that the unique identifier specifies a destination that is an “edge router or a network device behind an edge router that communicates through an authenticated and encrypted channel,” i.e., a “secure server” as defined by the Office Action. (*See Beser* 11:45-59; Keromytis Decl. ¶ 69.) Indeed, a list of numbers does not characterize any of the numbers, much less disclose that one or more of those numbers may be for a secure destination while others are not. (Keromytis Decl. ¶ 69.) In fact, not only does *Beser* fail to disclose or suggest that the directory database indicates whether a destination “communicates through an authenticated and encrypted channel” as defined by the Office Action, but rather *Beser* expressly describes encryption as “infeasible” and “inappropriate” for VoIP. (*Beser* 1:58-2:17; Keromytis Decl. ¶ 69.)

The Request, recognizing these weaknesses of *Beser*, nevertheless asserts that it is inherent in *Beser*’s E.164 directory feature that if a domain name is unknown to the trusted-third-party network, “it will not route the request further,” and that this allegedly discloses the “determining” step of claim 1. But *Beser* simply does not disclose that this list of numbers has any purpose related to security, nor does the Request offer any specific reasons why this would *necessarily* be the case. (Keromytis Decl. ¶ 69.) Thus, because it is inappropriate for inherency purposes to rely solely on either interpretation or “common knowledge” in the art without evidentiary support in the record, the Request’s inherency assertions are not properly supported and fall far short of establishing that the missing descriptive matter “is *necessarily* present in the thing described in the reference.” *In re Robertson*, 169 F.3d at 745 (emphasis added); *see also* M.P.E.P. § 2112. Furthermore, neither has the Request identified any particular reasoning that would have led a person of ordinary skill to modify *Beser* to include these claim features. M.P.E.P. § 2141(III)(G).

Finally, as noted above, *Kent* does not make up for the deficiencies of *Beser*. Nor do the Office Action and the Request assert that it does. (Apple Req. at 129, relying on *Beser* alone for this claim feature.)

Thus, for at least the reasons provided above, *Beser* and *Kent*, alone or in combination, do not disclose or suggest the “determining” step of claim 1. Accordingly, Patent Owner requests that the rejection of claim 1 be withdrawn, and its patentability confirmed.

**c. The Combination of *Beser* and *Kent* Fails to Disclose
“When the Intercepted DNS Request Does Not
Correspond to a Secure Server, Forwarding the DNS
Request to a DNS Function that Returns an IP Address of
a Nonsecure Computer”**

Independent claim 1 recites, among other things, “when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer.” The combination of *Beser* and *Kent* does not disclose or suggest this feature.

The Office Action and the Apple Request allege that *Beser* discloses the “forwarding” step of claim 1 because the trusted-third-party network device of *Beser* “will, by its nature of being a DNS server, simply return the IP address associated with the (non-secure) domain.” (OA at 28; Apple Req. at 130.) For these claim features, the Request relies on *Beser* alone, which does not disclose or suggest this feature for the reasons discussed below. (Apple Req. at 129-30, discussing only *Beser*.)

First, the Request alleges that in *Beser*’s system, a DNS request is sent to the trusted-third-party network device, which will resolve and return the IP address if the requested destination does not cause the establishment of an IP tunnel between the first and second network devices. (*Id.* at 130.) This is apparently because the trusted-third-party network device will, “by its nature of being a DNS server, simply return the IP address associated with the [non-secure] domain.” (*Id.*) But even assuming arguendo that this is correct, this argument fails to allege that any “forwarding” step is performed by any component of *Beser*. Rather, the Request is asserting that the trusted-third-party network device will retain the request and resolve it on its own instead of forwarding it to a “DNS function,” as recited in claim 1. Thus, *Beser* and *Kent* do not disclose or suggest the “forwarding” step of claim 1, and the rejection should be withdrawn.

Second, the Request asserts with respect to the “determining” step of claim 1 that “if a domain name sent to the trusted-third-party-network device . . . is unknown to the trusted-third-party-network device,” (i.e., thereby allegedly corresponding to a non-secure server), “it will not route the request further.” (OA at 28; Apple Req. at 129, emphasis added.) Thus, the Request’s argument for the “determining” step of claim 1 explicitly relies on the premise that with a nonsecure server, the alleged request does not pass beyond the trusted-third-party network device. As a result, assuming arguendo that the alleged “determining” step occurs as argued by the Request, it would be impossible for the “forwarding” step to occur if the alleged DNS request does not correspond to a secure server, because the alleged “determining” step affirmatively rules out the possibility of forwarding the

request further. Accordingly, *Beser* does not disclose or suggest the “determining” and “forwarding,” features and, therefore, the § 103(a) rejection should be withdrawn.

Kent does not make up for the deficiencies of *Beser* because *Kent* also does not disclose or suggest “forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer,” as recited in claim 1. Nor does the Request assert that it does. (Apple Req. at 129-30, relying only on *Beser* for the “forwarding” feature.)

Thus, for at least the reasons provided above, *Beser* and *Kent*, alone or in combination, do not disclose or suggest the “forwarding” step of claim 1. Accordingly, Patent Owner requests that the rejection of claim 1 be withdrawn, and its patentability confirmed.

d. The Combination of *Beser* and *Kent* Fails to Disclose “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

Independent claim 1 recites, among other things, “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.” The combination of *Beser* and *Kent* does not disclose or suggest these features.

The Office Action and the Apple Request assert that “*Beser* explains that ordinarily all IP traffic within an IP tunnel (or channel) will be encrypted” and that “IPsec should be used to encrypt traffic in IP tunnels.” (OA at 28; Apple Req. at 131.) This is incorrect.

As explained previously, *Beser* teaches away from using encryption techniques in tunneling connections and VoIP applications, the technology with which *Beser* is primarily concerned. Furthermore, the Request’s cited portions of *Beser* describing hiding “identities” have nothing to do with any alleged encrypted channel. (See Apple Req. at 130-31, citing *Beser* 2:36-40, 11:9-25, 12:6-19; Keromytis Decl. ¶¶ 70-71.) Merely hiding “identities” does not entail encryption of any kind between a client and an alleged secure server, i.e., “an edge router or a network device behind an edge router,” as defined by the Office Action. (Keromytis Decl. ¶ 71.) Instead, hiding identities is part of *Beser*’s proposed alternative to encryption, as it recites that “[i]t is therefore desirable to establish a tunneling association that hides the identity of the originating and terminating ends of the tunneling.” (*Beser* 2:36-38, emphasis added; Keromytis Decl. ¶ 71; see also *Beser* 1:56-58, explaining that “accumulating all [encrypted] packets from one source address may provide the hacker with sufficient information to decrypt the message.”)

Indeed, the only instance of encryption disclosed in *Beser* and cited by the Apple Request is concerned with encrypting a unique identifier before it is sent to the trusted-third-party network

device, i.e., the alleged DNS proxy module, for association with the address of a second network device. (See Apple Req. at 128, 130, citing *Beser* 11:22-24.) Yet this preliminary communication only occurs between the client and the alleged DNS proxy module, and it occurs *before* any tunneling is established. (See *Beser* 11:22-24; Keromytis Decl. ¶ 72.) Thus, encrypting a unique identifier on its way to the trusted-third-party network device does not disclose initiating an encrypted channel between the client and the alleged secure server. (*Id.*)

Kent does not make up for these deficiencies of *Beser*. Not only does *Beser* explicitly teach away from using IPSec—the specific security protocol described in *Kent*—but *Kent* also is not concerned with initiating encryption based on a *DNS request*. (*Id.* at ¶ 73.) Accordingly, even if *Beser* and *Kent* could have been combined, the combination would not reveal “*when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.*” *Ex parte Burgess*, Appeal No. 2008-2820, 2009 WL 291172, at *3 (P.B.A.I. Feb. 6, 2009) (to support an obviousness rejection, “all of the claim limitations must be taught or suggested by the prior art applied and . . . all words in a claim must be considered”).

Thus, for at least the reasons provided above, *Beser* and *Kent*, alone or in combination, do not disclose or suggest this feature of claim 1. Accordingly, Patent Owner requests that the rejection of claim 1 be withdrawn, and its patentability be confirmed.

4. Independent Claims 7 and 13

Independent claim 7 recites features similar to those described above for claim 1. For example, claim 7’s recited feature of “a domain name server (DNS) proxy module . . . intercepting a DNS request sent by a client” is similar to the “domain name server (DNS) proxy module that intercepts DNS requests sent by a client” feature of claim 1. Also, claim 7’s recited feature of “determining whether the intercepted DNS request corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Additionally, claim 7’s recited feature of “forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer” is similar to the “forwarding” feature of claim 1. Furthermore, claim 7’s recited feature of “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” is similar to the “automatically initiating” feature of claim 1, also discussed above. Accordingly, *Beser* in view of *Kent* does not disclose or suggest these features of claim 7 for similar reasons as those discussed above with respect to claim 1.

Independent claim 13 also recites features similar to those described above for claim 1. For example, claim 13’s recited feature of “determining whether a DNS request sent by a client

corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Also, claim 13’s recited feature of “forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer” is similar to the “forwarding” feature of claim 1. But independent claim 13 also recites features that differ from the features recited in claim 1, and these features are not even addressed by the Apple Request or the Office Action. For example, claim 13 recites “when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel” The Apple Request, however, ignores this difference in claim language and instead quotes another portion of independent claim 1 when purporting to reject claim 13. (*See, e.g.*, Apple Req. at 143-47.) By ignoring the language of claim 13 and instead analyzing a feature of claim 1, the rejection of claim 13 in view of *Beser* in view of *Kent* is improper for failing to consider all of the words in the claim. M.P.E.P. § 2143.03 (“*All words* in a claim must be considered in judging the patentability of that claim against the prior art.”) (emphasis added) (internal citations omitted). Moreover, to the extent the Requester and the Office later assert that the features recited in claim 13 are similar to the features recited in claim 1, Patent Owner asserts that *Beser* in view of *Kent* does not disclose or suggest these features of claim 13 for similar reasons as those discussed above with respect to claim 1.

Thus, for these reasons, Patent Owner requests that the rejection of claims 7 and 13 under § 103(a) be withdrawn, and their patentability confirmed.

5. Dependent Claims 2, 8, and 14

Dependent claims 2, 8, and 14 depend from independent claims 1, 7, and 13, respectively, and include all of their features. Thus, claims 2, 8, and 14 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 2, 8, and 14 also distinguish over *Beser* and *Kent* for additional reasons. For example, claims 2, 8, and 14 recite “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.” *Beser* and *Kent*, alone or in combination, do not disclose or suggest this feature.

The Office Action and the Apple Request allege that authentication of a client computer is inherent in *Beser* because an “authentication is required at the initiation of” the tunneling association (i.e., the alleged encrypted channel). (OA at 28-29; Apple Req. at 132, citing *Beser* 11:23-25.) This is incorrect.

The Request merely provides a generalized example of how servers requiring authentication request credentials. (Apple Req. at 132.) But the cited portions of *Beser* fail to disclose requiring

authentication of a client computer in conjunction with a tunneling association. (Keromytis Decl. ¶ 75.) As previously explained, these portions of *Beser* are merely concerned with encrypting or authenticating a *unique identifier* sent to the trusted-third-party network device for association with a second network device. (See Apple Req. at 132, citing *Beser* 11:22-24; Keromytis Decl. ¶ 75.)

Moreover, *Beser* discloses no reason for requiring that the client be authorized, or that any form of authorization or authentication would *necessarily* be present for accessing the third-party network device of *Beser*, much less the alleged secure server. *In re Robertson*, 169 F.3d at 745. Indeed, that requirement is not found within *Beser*, and the system of *Beser* is disclosed as operating sufficiently without authorization or authentication. (Keromytis Decl. ¶ 75.)

Kent does not make up for these deficiencies of *Beser*. *Kent* is merely directed to IPsec, and is not concerned with determining, at a data processing device, whether a client is authorized to access a secure server. Thus, the rejection of claims 2, 8, and 14 over *Beser* in view of *Kent* should be withdrawn.

The Request also fills nearly two single-spaced pages of argument directed at combining *Beser* and *Kent* in order to show an encrypted channel, none of which attempts to show “*sending a request to the secure server* to establish an encrypted channel between the secure server and the client” (emphasis added), as recited by claim 2. Furthermore, the superfluous argument directed to establishing an encrypted channel fails for the additional reason that *Beser* teaches away from using encryption, as discussed above.

The Request also reiterates the same arguments it made earlier with respect to a DNS request and transparency, possibly in an attempt to further evidence “sending a request to the secure server to establish an encrypted channel between the secure server and the client.” However, the Request has not explained how the request to initiate a VoIP association may be both the DNS request and the request sent to the secure server to establish the encrypted channel when the client is authorized to access the secure server. Consequently, a person of ordinary skill in the art would not have understood *Beser* to disclose the elements of claims 2, 8, and 14.

Thus, for at least the reasons provided, *Beser* and *Kent* do not render claims 2, 8, and 14 obvious. Accordingly, Patent Owner requests that the rejection of claims 2, 8, and 14 be withdrawn, and their patentability confirmed.

6. Dependent Claims 4, 10, and 16

Dependent claims 4, 10, and 16 depend from one or more of claims 1-3, 7-9, and 13-15, and include all of their features. Thus, claims 4, 10, and 16 should be confirmed for at least the reasons

discussed above with respect to those claims. Claims 4, 10, and 16 also distinguish over *Beser* and *Kent* for additional reasons. For example, claims 4, 10, and 16 recite the data processing device “wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.” *Beser* and *Kent*, alone or in combination, do not disclose or suggest this feature.

The Request appears to admit that *Beser* and *Kent* do not explicitly disclose the additional feature of claims 4, 10, and 16, as it simply points to a limited disclosure in *Beser* of where certain standards may be located and generally states, without any support, that a person of ordinary skill in the art “would recognize that among the most common methods of initiating communication with a remote server is through software applications such as a web browser.” (See, e.g., Apple Req. at 134, discussing claim 4.) But *Beser* merely discloses that some network devices may “interact” with the network system of *Beser* based on the cited standards, not that any particular aspect found within one of those standards could be used to initiate the tunneling association of *Beser*. (See *Beser* 4:55-5:2.) Thus, *Beser* and *Kent* do not disclose or suggest the additional “web browser” feature of claims 4, 10, and 16.

The Request’s analysis is improper for at least two additional reasons. First, the Request bases its arguments on what a person of ordinary skill in the art “would recognize” today (Apple Req. at 161, 172, 183), which is improper for determining what a person of ordinary skill in the art would have recognized “at the time of the invention.” M.P.E.P. §§ 2141.01, 2141.02; see also *Mintz v. Dietz & Watson, Inc.*, Case No. 2010-1341, at 9 (Fed. Cir. May 30, 2012). Second, citing to references that do not disclose or suggest a recited feature and merely stating, without any support, that the level of ordinary skill in the art would render the recited feature obvious contravenes the requirement that “analysis supporting a rejection under 35 U.S.C. [§] 103 should be made explicit.” M.P.E.P. § 2142. Indeed, a rejection based on obviousness “cannot be sustained with mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *KSR*, 126 S. Ct. at 1741 (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)).

Since there is no articulated reasoning as to how *Beser* and *Kent* render claim 4 obvious, or evidence that a person of ordinary skill in the relevant art *at the time of the invention* would have recognized how to perform the step of “automatically initiating an encrypted channel between the client and the secure server” in the claimed manner using software applications such as a web browser, a prima facie case of obviousness has not been established.

Thus, for at least the reasons provided, *Beser* and *Kent* do not render claims 4, 10, and 16 obvious. Accordingly, Patent Owner requests that the rejection of claims 4, 10, and 16 be withdrawn, and their patentability confirmed.

7. Dependent Claims 5 and 11

Dependent claims 5 and 11 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, claims 5 and 11 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 5 and 11 also distinguish over *Beser* and *Kent* for additional reasons. For example, claims 5 and 11 recite that the feature of automatically initiating the encrypted channel between the client and the secure server “comprises establishing an [IP] address hopping scheme between the client and the secure server.” *Beser* and *Kent*, alone or in combination, do not disclose or suggest this feature.

The Office Action and the Apple Request point generally to the NAT protocol disclosed by *Beser*, and conclude without any support that the NAT protocol “is an IP hopping scheme within the meaning of claim 5.” (*See, e.g.*, OA at 28-29; Apple Req. at 135, discussing claim 5.) But a rejection based on obviousness “cannot be sustained with mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *KSR*, 126 S. Ct. at 1741 (citing *In re Kahn*, 441 F.3d at 988)). Since the Request has not provided any reasoning whatsoever to support its conclusion that the referenced “NAT protocol is an IP hopping scheme within the meaning of claim 5,” a prima facie case of obviousness has not been established with respect to claim 5, and the rejection of claim 5 under 35 U.S.C. § 103 cannot be sustained.

Moreover, *Beser* discloses NAT (network address translation) only to show that it is *not* to be used with VoIP applications—the technology with which *Beser* is primarily concerned. (*Beser* 2:18-35; Keromytis Decl. ¶ 77.) Specifically, *Beser* teaches that

network address translation [is] computationally expensive, *causes security problems* by preventing certain types of encryption from being used, or *breaks a number of existing applications* in a network that cannot provide network address translation (*e.g.*, File Transfer Protocol (“FTP”)). What is more, network address translation *interferes with* the end-to-end routing principal of the Internet that recommends that packets flow end-to-end between network devices without changing the contents of any packet along a transmission route Once again, due to computer power limitations, this form of tunneling may be *inappropriate* for the transmission of multimedia or VoIP packets.

(*Id.* at 2:22-35, emphases added; Keromytis Decl. ¶ 77.) Accordingly, *Beser* specifically teaches against using NAT, especially in connection with encryption and VoIP applications. (Keromytis

Decl. ¶ 77.) Thus, *Beser* discloses a system and method directed to initiating a tunneling association in which NAT is not used because of the problems with NAT.

One skilled in the art at the time of the invention, upon reviewing *Beser*, would have had no objective reason to vary the tunneling association taught by *Beser* to include NAT, a protocol of the type that *Beser* specifically teaches not to use. (*Id.*) The portions of *Beser* teaching against using NAT because it “causes security problems,” “breaks a number of existing applications,” “interferes” with routing principles, and may further be “inappropriate” and “computationally expensive,” cannot be ignored in a prima facie obviousness determination. (*Id.*) See M.P.E.P. § 2141.02(VI); see also *In re ICON Health & Fitness Inc.*, 496 F.3d 1374 (Fed. Cir. 2007) (citing *KSR*, 126 S. Ct. at 1741). Thus, *Beser* does not render claims 5 and 11 obvious.

Additionally, *Kent* does not disclose this feature, nor does the Request assert that it does. (See Apple Req. at 135, 142-43, discussing only *Beser* with respect to claims 5 and 11.)

Thus, for at least the reasons provided, *Beser* and *Kent*, alone or in combination, do not render claims 5 and 11 obvious. Accordingly, Patent Owner requests that the rejection of claims 5 and 11 be withdrawn, and their patentability confirmed.

8. Dependent Claims 6 and 12

Dependent claims 6 and 12 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, claims 6 and 12 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 6 and 12 also distinguish over *Beser* and *Kent* for additional reasons. For example, claims 6 and 12 recite that the feature of automatically initiating the encrypted channel between the client and the secure server “avoids sending a true IP address of the secure server to the client.” *Beser* and *Kent*, alone or in combination, do not disclose or suggest this feature.

The Apple Request’s allegation that *Beser* implements measures that “prevent ‘the identity of the originating and terminating ends of the tunneling association form [sic] the users of a public network’” is misleading. (Apple Req. at 135.) The Request has omitted the word “other” from the portion of *Beser* relied upon to support its argument. The cited portion reads in full:

It is therefore desirable to establish a tunneling association that hides the identity of the originating and terminating ends of the tunneling association from the *other users* of a public network. Hiding the identities may prevent a hacker from intercepting all media flow between the ends.

(*Beser* 2:36-40, emphasis added; Keromytis Decl. ¶¶ 78-79.) Thus, the sentence cited by the Request, and relied upon for the rejection of claim 6, specifically teaches that *other users* would be

prevented from learning the identity of the terminating end of the tunneling association—not the users at the originating or terminating ends of the tunneling association. (Keromytis Decl. ¶ 79.) By omitting the word “other,” the Request gives a reader the false impression that *Beser* teaches that the identity would be concealed from a user who is participating in the tunneling association, i.e., at the originating or terminating ends. (*Id.*) In fact, the immediately ensuing sentence, which was also omitted by the Request, makes it clear that the specific intent of *Beser* is to hide identities from a “hacker.” (*Id.*) Regardless, claim 6 does not recite *hiding* identities but *avoiding sending* the true IP address of the secure server to the client to begin with. Hiding something, for example, in a tunnel, is not the same as not sending it in the first place.

The Request also points to a portion of *Beser* that describes potentially hiding a “source IP address.” (Apple Req. at 135, citing *Beser* 2:12-14.) This “source IP address,” however, does not have anything to do with what the Office Action points to as the secure server (i.e., “an edge router or a network device behind an edge router that communicates through an authenticated and encrypted channel”), or preventing the user who sent the packet from receiving the true IP address of the packet recipient. (See OA at 29; *Beser* 2:1-17; Keromytis Decl. ¶ 79.) Accordingly, nothing in *Beser* discloses or suggests that “automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to client,” as recited in claim 1.

Furthermore, *Kent* does not disclose this feature, nor does the Request assert that it does. (See Apple Req. at 135, 143, discussing only *Beser* with respect to claims 6 and 12.)

Thus, *Beser* and *Kent* do not render claims 6 and 12 obvious, and the rejection of claims 6 and 12 should be withdrawn and their patentability confirmed.

For the foregoing reasons, Patent Owner respectfully requests that the rejection of claims 1, 2, 4-8, 10-14, and 16 under § 103(a) based on *Beser* in view of *Kent* be withdrawn, and the patentability of these claims be confirmed.

F. The Rejections Based on *Kiuchi* Should Be Withdrawn

The Office Action rejects claims 1-4, 6-10, and 12-16 under 35 U.S.C. § 102(b) based on Takahiro Kiuchi and Shigekoto Kaihara, “C-HTTP — The Development of a Secure, Closed HTTP-based Network on the Internet” (Cisco Req. Ex. D-1) (“*Kiuchi*”), and rejects claims 1-16 under 35 U.S.C. § 103(a) based on *Kiuchi* in view of certain secondary references. For the reasons discussed below, these rejections should be withdrawn, and the claims should be confirmed.

1. Overview of *Kiuchi*

Kiuchi proposes a technique called “closed HTTP” (C-HTTP) for providing secure HTTP communications “within a closed group of institutions on the Internet, where each member is protected by its own firewall.” (*Kiuchi* 64.) According to *Kiuchi*, C-HTTP is useful in the medical community, where “there is a strong need for closed networks among hospitals and related institutions” to handle patient data and other sensitive medical information. (*Id.*)

C-HTTP requires three main components: “1) a client-side proxy on the firewall of one institution, 2) a server-side proxy on the firewall of another institution, and 3) a C-HTTP name server, which manages a given C-HTTP-based network and the information for [all of its] proxies.” (*Id.*) When an institution wants to participate in a C-HTTP network, it must, among other things, install a client-side and/or server-side proxy on its firewall, register an IP address and a hostname for its proxy, and give the proxy’s public key to the C-HTTP name server. (*Id.* at 65.) During C-HTTP communications, “[a] client-side proxy and server-side proxy communicate with each other using a secure, encrypted protocol (C-HTTP).” (*Id.* at 64.)

When a user agent computer behind a client-side proxy wants to establish a C-HTTP session with a server behind a server-side proxy, the following C-HTTP setup process occurs:

- (1) The client-side proxy asks the C-HTTP name server whether it can communicate with the server.
- (2) The C-HTTP name server determines whether the server-side proxy is in the closed network and whether the connection is permitted.
- (3) If so, the C-HTTP name server sends the IP address and public key of the server-side proxy, as well as request and response Nonce values, to the client-side proxy.
- (4) The client-side proxy sends a connection request to the server-side proxy, encrypted with the server-side proxy’s public key.
- (5) The server-side proxy asks the C-HTTP name server whether the client-side proxy is also in the closed network and whether the connection is permitted.
- (6) If so, the C-HTTP name server sends to the server-side proxy the IP address and public key of the client-side proxy, as well as the same request and response Nonce values previously sent to the client-side proxy.
- (7) The server-side proxy then authenticates the client-side proxy, generates a connection ID, generates a second symmetric key for C-HTTP response

encryption, and sends this information to the client-side proxy. When the client-side proxy accepts and checks this information, the connection is established.

- (8) Once the connection is established, a client-side proxy forwards requests from the user agent in encrypted form using C-HTTP format.

(*Id.* at 65-66.) *Kiuchi* explains that “[t]he [C-HTTP] session is finished when the client accesses another C-HTTP server.” (*Id.* at 65.)

2. The Rejection of Claims 1-4, 6-10, and 12-16 Under 35 U.S.C. § 102(b) Based on *Kiuchi* Should Be Withdrawn (Issue 7)

The Office Action rejects claims 1-4, 6-10, and 12-16 under 35 U.S.C. § 102(b) based on *Kiuchi*. (OA at 29.) For the reasons discussed below, this rejection should be withdrawn and the claims should be confirmed.

a. Independent Claim 1

Independent claim 1 is directed to a data processing device storing a DNS proxy module. *Kiuchi* fails to disclose the combination of features recited in this claim for at least the reasons discussed below.

(i) *Kiuchi* Fails to Disclose “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”

Independent claim 1 recites, among other things, “a *domain name server (DNS)* proxy module that intercepts *DNS* requests sent by a client” (emphases added). *Kiuchi* does not disclose this feature.

The Office Action and the Cisco Request assert that *Kiuchi*’s client-side proxy corresponds to the DNS proxy module of claim 1 because it attempts to resolve a domain name to an IP address by requesting the IP address from the C-HTTP name server, and if that fails, requesting the IP address through a DNS lookup. (OA at 29; Cisco Req. Ex. E-1 at 4.) The Request and the Office Action further assert that the client-side proxy of *Kiuchi* intercepts “DNS requests” as recited in claim 1 because when a “resource name[] with a connection ID . . . is selected and requested by an end-user, the client-side proxy takes off the connection ID and forwards the stripped, . . . original resource name to the [C-HTTP name] server.” (OA at 29; Cisco Req. Ex. E-1 at 5, quoting *Kiuchi* 65.) These assertions are incorrect.

Kiuchi does not disclose the features of claim 1 because *Kiuchi* expressly teaches that C-HTTP does not involve DNS. (Keromytis Decl. ¶ 84.) “In a C-HTTP-based network, *instead of DNS*, a C-HTTP-based secure, encrypted name and certification service is used.” (*Kiuchi* 64,

emphasis added.) *Kiuchi* further explains that “[t]he *DNS name service is not used for hostname resolution* as the original secure name service, including certification, is used for the C-HTTP-based network.” (*Id.*, emphasis added.) Because *Kiuchi* expressly states that its C-HTTP techniques do not involve DNS, one of ordinary skill in the art would not have understood *Kiuchi* to disclose the features of claim 1, such as “a *domain name server (DNS) proxy module that intercepts DNS requests.*” (Keromytis Decl. ¶ 84.)

Indeed, in a related reexamination proceeding, the Office previously recognized that *Kiuchi* does not disclose or suggest using DNS, and thus declined to adopt all proposed rejections of any claim reciting a “DNS request” under § 102 and § 103 based on *Kiuchi*. (Office Action in control no. 95/001,679 at 7, 8.) Patent Owner respectfully submits that the Office’s conclusion in control no. 95/001,679 proceeding is correct, and should compel withdrawal of the rejections based on *Kiuchi* here. Accordingly, for the same reasons as those presented in the Office Action in control no. 95/001,679, *Kiuchi* does not disclose “a domain name server (DNS) proxy module that intercepts DNS requests,” as recited in claim 1, and, therefore, the rejection of claim 1 should be withdrawn and its patentability confirmed.

(ii) *Kiuchi* Fails to Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

Independent claim 1 recites, among other things, “a domain name server (DNS) proxy module that . . . performs the step[] of . . . determining whether the intercepted DNS request corresponds to a secure server.” *Kiuchi* fails to disclose this feature.

As explained above, *Kiuchi*’s techniques do not involve DNS. Thus, *Kiuchi* does not disclose or suggest “determining whether the intercepted *DNS request* corresponds to a secure server” (emphasis added), as recited in claim 1.

Still, the Office Action and the Cisco Request assert that *Kiuchi* discloses a DNS proxy module that performs the “determining” step of claim 1 because “[i]f the [C-HTTP] name server confirms that the query is legitimate, it examines whether the requested server-side proxy is registered in the closed network and is permitted to accept the connection from the client-side proxy.” (OA at 29; Cisco Req. Ex. E-1 at 7, quoting *Kiuchi* 65.) The Office Action and the Cisco Request are incorrect for the following additional reasons.

Claim 1 recites that the DNS proxy module performs the step of determining whether the intercepted DNS request corresponds to a secure server. In attempting to meet the claim language, the Request first asserts that the client-side proxy is the alleged DNS proxy module (Cisco Req.

Ex. E-1 at 4, 5), but then points to the C-HTTP name server as performing the recited “determining” step (*id.* at 7). Indeed, in the portion of *Kiuchi* quoted by the Request, the client-side proxy merely “asks the C-HTTP name server whether it can communicate with the host specified in a given URL.” (*Id.*, quoting *Kiuchi* 65.) Then, as the Request emphasizes, it is the C-HTTP name server, not the client-side proxy, which “examines whether the requested server-side proxy is registered in the closed network and is permitted to accept the connection from the client-side proxy.” (*Id.*, quoting *Kiuchi* 65.) Thus, the Request, improperly mixing and matching the components of *Kiuchi*, fails to even allege that the supposed DNS proxy module—the client-side proxy—performs the alleged “determining” step. See *Net MoneyIN*, 545 F.3d at 1371 (to anticipate, a reference must disclose “all of the limitations arranged or combined in the same way as recited in the claim”). Indeed, as demonstrated, it is the C-HTTP name server that examines whether the server-side proxy is registered in the closed network, i.e., making the alleged determination. (Keromytis Decl. ¶ 86.) Accordingly, not only does *Kiuchi* fail to disclose the “determining” feature of claim 1, but the Request and the Office Action have failed altogether to make out a prima facie case of anticipation of claim 1 based on *Kiuchi*.

For these reasons, the rejection of claim 1 should be withdrawn and its patentability confirmed.

(iii) *Kiuchi* Fails to Disclose “Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

Independent claim 1 recites, among other things, “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.” *Kiuchi* does not disclose this feature.

Again, as set forth above, *Kiuchi*’s techniques do not involve DNS. For this reason, *Kiuchi* cannot meet the above feature of the claim, which calls for automatically initiating an encrypted channel between the client and the secure server “when the intercepted *DNS request* corresponds to a secure server” (emphasis added). Since *Kiuchi* is deficient regarding DNS, this criterion cannot be met in the C-HTTP system.

Still, the Office Action and the Request assert that *Kiuchi* discloses the “automatically initiating” feature because of various connection-request steps that *Kiuchi* allegedly takes *in response* to the alleged “determining” step. (OA at 29; Cisco Req. Ex. E-1 at 10.) This is incorrect.

The mere fact that a connection request follows the alleged “determining” step fails to disclose “automatically initiating an encrypted channel,” as recited in claim 1. *Kiuchi* does not

disclose how its C-HTTP system operates at a broader functional level, only discussing the individual steps of the C-HTTP communication process without regard to how C-HTTP connections are ultimately established. (*See Kiuchi* 65-57; Keromytis Decl. ¶ 88.) Thus, *Kiuchi* is silent on whether any “automatic initiating” occurs at all, much less in connection with initiating the alleged “encrypted channel.” (Keromytis Decl. ¶ 88.) Indeed, given *Kiuchi*’s limited disclosure, the C-HTTP system may in fact require its intended hospital personnel users to actively participate in establishing C-HTTP connections. (*Id.*) As a result, *Kiuchi* does not disclose the “automatically initiating” feature of claim 1, and the rejection should be withdrawn.

Moreover, the Request has failed to establish that *Kiuchi* inherently discloses any “automatically initiating” features. *Kiuchi*’s C-HTTP system might indeed require user interaction during connection establishment, as nothing in *Kiuchi* necessarily requires “automatically initiating” the C-HTTP connections. (*Id.* at ¶89.); *In re Robertson*, 169 F.3d at 745.

Thus, for these reasons, *Kiuchi* does not disclose all of the claimed elements in claim 1, and the rejection should be withdrawn, and the patentability of claim 1 confirmed.

b. Independent Claims 7 and 13

Independent claim 7 recites features similar to those described above for claim 1. For example, claim 7’s recited feature of “a domain name server (DNS) proxy module . . . intercepting a DNS request sent by a client” is similar to the “domain name server (DNS) proxy module that intercepts DNS requests sent by a client” feature of claim 1. Also, claim 7’s recited feature of “determining whether the intercepted DNS request corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Furthermore, claim 7’s recited feature of “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” is similar to the “automatically initiating” feature of claim 1, also discussed above. Accordingly, *Kiuchi* does not disclose these features of claim 7 for similar reasons as those discussed above with respect to claim 1.

Independent claim 13 also recites features similar to those described above for claim 1. For example, claim 13’s recited “domain name server (DNS)” feature is similar to the “domain name server (DNS)” feature of claim 1. Also, claim 13’s recited “determining whether a DNS request sent by a client corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Independent claim 13 also recites features that differ from the features recited in claim 1. For example, claim 13 recites “when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel” But given the arguments presented in the Cisco

Request and the Office Action (Cisco Req. Ex. E-1 at 21-23), however, Patent Owner asserts that *Kiuchi* does not disclose these features of claim 13 for similar reasons as those discussed above with respect to claim 1.

Thus, for these reasons, Patent Owner requests that the rejection of claims 7 and 13 under § 102(b) be withdrawn, and their patentability confirmed.

c. Dependent Claims 3, 9, and 15

Dependent claims 3, 9, and 15 ultimately depend from independent claims 1, 7, and 13, respectively, and include all of their features. Thus, *Kiuchi* does not anticipate these claims, and the rejection of these claims should be withdrawn, for at least the reasons discussed above in connection with independent claims 1, 7, and 13. Claims 3, 9, and 15 also distinguish over *Kiuchi* for additional reasons. For example, dependent claims 3, 9, and 15 each recite “when the client is not authorized to access the secure server, returning a host unknown error message to the client.” *Kiuchi* does not disclose at least these features.

The Cisco Request asserts that *Kiuchi* anticipates these claims via a multistep process. (Cisco Req. Ex. E-1 at 13-14.) First, if the C-HTTP name server determines that a connection is not permitted between a client-side proxy and a server-side proxy, the C-HTTP name server will “send[] a status code which indicates an error” to the client-side proxy. (*Id.*, quoting *Kiuchi* 65.) Next, if the client-side proxy receives an error status, it then performs a DNS lookup. (*Id.* at 13-14, quoting *Kiuchi* 65.) Finally, if the client-side proxy then attempts to look up the server-side proxy’s secure hostname using DNS, the Request asserts that this might result in an error message if the server-side proxy’s DNS name is different than its C-HTTP hostname. (*Id.* at 14, quoting *Kiuchi* 68.) Thus, the Request appears to admit that *Kiuchi* fails to explicitly disclose the additional features of dependent claims 3, 9, and 15 by instead resorting to an inherency argument. This argument is incorrect.

As quoted by the Request, *Kiuchi* explains that when no connection is permitted between a client-side proxy and a server-side proxy, the C-HTTP name server sends a status code indicating an error to the client-side proxy:

If the connection is permitted, the C-HTTP name server sends the IP address and public key of the server-side proxy and both request and response Nonce values. If it is not permitted, it sends a status code which indicates an error. If a client-side proxy receives an error status, then it performs DNS lookup, behaving like an ordinary HTTP/1.0 proxy.

(*Kiuchi* 65.) This passage fails to disclose the features of claims 3, 9, and 15 for at least two reasons. (Keromytis Decl. ¶ 91.) First, claims 3, 9, and 15, incorporating the features of independent claims 1, 7, and 13, recite that the *DNS proxy module* returns a host unknown error message. But in *Kiuchi*,

the C-HTTP name server sends the “status code which indicates an error,” not the client-side proxy identified as the alleged DNS proxy module. (*Id.*; see Cisco Req. Ex. E-1 at 4-5.) Second, claims 3, 9, and 15 recite that the host unknown error message is returned “to the client.” But in *Kiuchi*, the “status code” is returned not to the client, but to the client-side proxy. (*Kiuchi* 65; Keromytis Decl. ¶ 91.) Thus, a person of ordinary skill would not understand the C-HTTP name server sending a “status code which indicates an error” to disclose “returning a host unknown error message to the client,” as recited in claims 3, 9, and 15. (Keromytis Decl. ¶ 91.)

An error message allegedly resulting inherently from the client-side proxy’s subsequent “DNS lookup” also fails to disclose the additional features of claims 3, 9, and 15, as the mere fact that a certain result or characteristic might occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. *In re Rijckaert*, 9 F.3d at 1534; *In re Robertson*, 169 F.3d at 745; see also M.P.E.P. § 2112.

Indeed, the Requester’s assertion that “a lookup of a server’s secure hostname using DNS will result in a host-not-found error *when the [C-HTTP and DNS hostnames] are different*” fails to establish that a host unknown error message will invariably be returned to the client in *Kiuchi*. (Cisco Req. Ex. E-1 at 14, emphasis added; Keromytis Decl. ¶ 92.) In fact, there are many scenarios in which *Kiuchi*’s system could return something other than a host unknown error message to the client. (Keromytis Decl. ¶ 92.) As one example, the C-HTTP and DNS hostnames for a server might be the same, which *Kiuchi* acknowledges as a possibility by simply indicating that these hostnames are “not necessarily the same.” (*Kiuchi* 68; Keromytis Decl. ¶ 92.) Thus, a lookup where the C-HTTP and DNS hostnames are the same might not result in any error at all, but could instead return an IP address. (Keromytis Decl. ¶ 92.) And even if the C-HTTP and DNS hostnames were different, the subsequent DNS lookup could fail in a manner that generates an error message other than a host-not-found error. (*Id.*) And even if a host-not-found error were returned in this scenario, it would be for a reason outside the context of C-HTTP. (*Id.*) Thus, the features of claims 3, 9, and 15 are neither disclosed by, nor inherent in, *Kiuchi*.

For these reasons, *Kiuchi* fails to disclose “when the client is not authorized to access the secure server, returning a host unknown error message to the client,” as recited in claims 3, 9, and 15, and the rejection of these claims should be withdrawn, and their patentability confirmed.

d. Dependent Claims 6 and 12

Claims 6 and 12 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, *Kiuchi* does not anticipate these claims, and the rejection of these claims should

be withdrawn for at least the reasons discussed above in connection with independent claims 1 and 7. Claims 6 and 12 also distinguish over *Kiuchi* for additional reasons. For example, claims 6 and 12 recite that the feature of automatically initiating the encrypted channel between the client and the secure server “avoids sending a true IP address of the secure server to the client.” *Kiuchi* has not been shown to disclose this feature.

The Office Action and the Cisco Request allege *Kiuchi* discloses this feature because “[a]s C-HTTP includes its own secure name service, which contains a certification mechanism, *it is impossible to know the IP address of a server-side proxy* even if its C-HTTP hostname (not necessarily the same as its DNS name) is known and vice versa.” (OA at 29; Cisco Req. Ex. E-1 at 17, quoting *Kiuchi* 68.) But the mere fact that C-HTTP includes its own name service and requires certification to resolve an IP address has nothing whatsoever to do with initiating an encrypted channel that involves avoiding sending the true IP address of the secure server to the client. (Keromytis Decl. ¶¶ 93, 94.) And moreover, the “vice versa” indicates that the IP address of a server-side proxy might in fact be known, but simply could not be used to learn the corresponding C-HTTP hostname due to the fact that C-HTTP employs its own name service with a certification mechanism. (*Id.* at ¶ 94.) Thus, *Kiuchi* discloses that the existence of the C-HTTP name server merely dictates that, absent certification, (1) a C-HTTP hostname cannot be used to learn the IP address of a server-side proxy, and (2) an IP address of a server-side proxy cannot be used to learn the C-HTTP hostname. (*Kiuchi* 68; Keromytis Decl. ¶ 94.) Thus, *Kiuchi* explicitly acknowledges that it is in fact possible to know the IP address of the server-side proxy. (Keromytis Decl. ¶ 94.)

The Request additionally argues, without support, that because the client-side proxy acts as an HTTP/1.0 compatible proxy, “it avoids sending the true IP address of the secure server to the client.” (Cisco. Req. Ex. E-1 at 16-17.) This is incorrect. *Kiuchi* does not disclose that the client-side proxy avoids sending the true IP address of the secure server to the client, and the Request notably does not identify any such disclosure in *Kiuchi*, instead apparently resorting to an inherency argument. (*See id.*; Keromytis Decl. ¶ 95.) *See also In re Rijckaert*, 9 F.3d at 1533 (“[W]hen the PTO asserts that there is an explicit or implicit teaching or suggestion in the prior art, it must indicate *where* such a teaching or suggestion appears in the prior art.”) (emphasis added).

But given that *Kiuchi* discloses that the client-side proxy is “a proxy server for external (outside the firewall) access,” i.e., is an *outward-looking* proxy, there is no reason why the client-side proxy would *necessarily* block or prevent a true IP address from being sent by the alleged secure server to the client. (Keromytis Decl. ¶ 95.) There is furthermore no reason why doing so would

necessarily occur as part of the process alleged to correspond to “automatically initiating the encrypted channel,” as recited in claim 1. (*Id.*) Thus, the Request has not shown that *Kiuchi* explicitly discloses the features of claims 6 and 12, and also has not shown them to be inherent by “mak[ing] clear that the missing descriptive matter is *necessarily* present in the thing described in the reference, and that it would be so recognized by persons of ordinary skill.” *In re Robertson*, 169 F.3d at 745 (emphasis added).

Accordingly, for these reasons, the § 102(b) rejection of claims 6 and 12 should be withdrawn, and their patentability confirmed.

e. Dependent Claims 2, 4, 8, 10, 14, and 16

Remaining claims 2, 4, 8, 10, 14, and 16 depend from one of independent claims 1, 7, and 13, and include all of their features. Thus, *Kiuchi* does not anticipate any of these claims for at least the reasons discussed above in conjunction with independent claims 1, 7, and 13. Claims 4, 10, and 16 also depend from one of claims 3, 9, and 15, which in turn depend from one of independent claims 1, 7, and 13, and include all of their features. Thus, *Kiuchi* further does not anticipate claims 4, 10, and 16 for at least the reasons discussed above in conjunction with claims 3, 9, and 15. For the reasons set forth above, the rejection of claims 2, 4, 8, 10, 14, and 16 under 35 U.S.C. § 102(b) based on *Kiuchi* should be withdrawn, and their patentability confirmed.

3. The Rejection of Claims 5 and 11 Under 35 U.S.C. § 103(a) Based on *Kiuchi* in View of *Martin* Should Be Withdrawn (Issue 8)

The Office Action rejects claims 5 and 11 under 35 U.S.C. § 103(a) based on *Kiuchi* in view of D.M. Martin, “A Framework for Local Anonymity in the Internet” (Cisco Req. Ex. D-6) (“*Martin*”). (OA at 29; Cisco Req. Ex. E-1 at 26-28.) For the reasons discussed below, the rejection should be withdrawn.

Claim 5 depends from independent claim 1, and claim 11 depends from independent claim 7. As explained above, *Kiuchi* does not disclose or suggest the features of claims 1 and 7, and thus does not support the rejection of those claims. The rejection of claims 5 and 11 should also be withdrawn because *Martin* does not remedy the deficiencies of *Kiuchi* with respect to independent claims 1 and 7. For instance, *Martin* does not disclose or suggest, and the Cisco Request and the Office Action do not rely upon *Martin* to show, at least “a domain name server (DNS) proxy module that intercepts DNS requests sent by a client, and for each intercepted DNS request, performs the step[] of . . . determining whether the intercepted DNS request corresponds to a secure server.” Thus, for at least the reasons set forth above, the rejection of claims 5 and 11 over *Kiuchi* in view of *Martin* should be withdrawn.

4. The Rejection of Claims 1-4, 6-10, and 12-16 Under 35 U.S.C. § 103(a) Based on *Kiuchi* in View of *Edwards* Should Be Withdrawn (Issue 14)

The Office Action rejects claims 1-4, 6-10, and 12-16 under 35 U.S.C. § 103(a) based on *Kiuchi* in view of Nigel Edwards and Owen Rees, “High Security Web Servers and Gateways” (Cisco Req. Ex. D-5) (“*Edwards*”). (OA at 30.) For the reasons discussed below, the rejection should be withdrawn, and the claims should be confirmed.

a. Overview of *Edwards*

Edwards discloses a “secure object gateway . . . to give fine grain access control” to services located on the back end of a web server. (*Edwards* 932.) With regard to Fig. 4, reproduced below, *Edwards* discloses that a naming interceptor at an object gateway intercepts a request to resolve a name corresponding to one of the services that is accessible from the back-end of the web server. (*Id.*) The naming interceptor returns a reference to a service interceptor also located at the object gateway. (*Id.*) A plugin at the requesting web server then invokes the service interceptor to reach the backend service. (*Id.*)

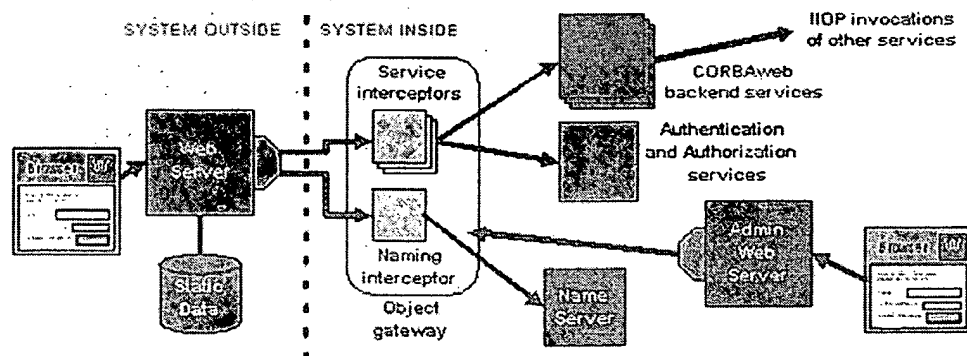


Fig. 4. Secure object gateway.

Edwards discloses that the plugin invoking the services in this manner may first need to be authenticated and authorized. (*Id.*) However, *Edwards* also discloses that an administrator may remove the authentication and/or authorization requirements for accessing these services (*Edwards* also refers to the services as “available targets”). (*See id.* at 933.)

b. Independent Claim 1

Independent claim 1 is directed to a data processing device storing a DNS proxy module. *Kiuchi* and *Edwards*, alone or in combination, fail to disclose or suggest the combination of features recited in this claim for at least the reasons discussed below.

(i) **The Combination of *Kiuchi* and *Edwards* Fails to Disclose or Suggest “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”**

As discussed above, *Kiuchi* explicitly teaches away from the features of claim 1, as a different type of service is used to resolve host names.

Edwards does not make up for these deficiencies of *Kiuchi* because, in contrast to the Request’s contentions, *Edwards* does not disclose employing DNS requests in any manner, let alone intercepting DNS requests. (Keromytis Decl. ¶ 98.) Rather, *Edwards* discloses requests to return a name of “services which are accessible from the back-end of the web server.” (*Edwards* 932; Keromytis Decl. ¶ 98.) With *Edwards*’s requests, the requesting device merely receives “a reference to a service interceptor,” which it later invokes, instead of receiving a network address or IP address as would occur with a successful DNS request. (*Edwards* 932; Keromytis Decl. ¶ 98.) In fact, the Request appears to admit that *Edwards* does not intercept DNS requests, as it asserts that *Edwards* merely “intercepts name service requests.” (Cisco Req. Ex. E-5 at 7.)

Furthermore, it would not have been obvious to modify *Kiuchi* to utilize any DNS features allegedly disclosed or suggested by *Edwards*. Indeed, in a related reexamination, the Office previously recognized that *Kiuchi* taught away from using DNS. (Office Action in control no. 95/001,679 at 7, 8.) Given “the explicit teachings of *Kiuchi* that a different type of service is used to resolve a host name in the secure C-HTTP environment,” the Office declined to adopt all proposed rejections of any claim reciting a “DNS request” under § 102 and § 103 based on *Kiuchi*. (*Id.* at 8.) Specifically, the Office stated that a proposed combination of *Kiuchi* with a reference directed to DNS requests “would not overcome the explicit teachings of *Kiuchi* that a different type of service is used to resolve a host name in the secure C-HTTP environment.” (*Id.*) Because *Kiuchi* explicitly teaches away, a person of ordinary skill in the art would not have understood that *Kiuchi* and *Edwards* should be combined to disclose or suggest “a domain name server (DNS) proxy module that intercepts DNS requests” (emphases added), as recited in claim 1.

Thus, *Kiuchi* and *Edwards*, alone or in combination, do not disclose or suggest these features of claim 1. Consequently, the § 103(a) rejection of claim 1 should be withdrawn, and its patentability should be confirmed.

(ii) The Combination of *Kiuchi* and *Edwards* Fails to Disclose or Suggest “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

As discussed above, *Kiuchi* fails to disclose the “determining” step recited in claim 1, as the Cisco Request first identifies the client-side proxy as the alleged DNS proxy module, but then points to the C-HTTP name server as performing the alleged “determining” step. In the § 103(a) rejection of claim 1, the Office Action and the Request again mix and match these components of *Kiuchi*, failing yet again to even allege that the supposed DNS proxy module performs the “determining” step recited in claim 1. (OA at 31; Cisco Req. Ex. E-5 at 9-10.) Thus, *Kiuchi* has not been shown to render claim 1 obvious, and the rejection should be withdrawn.

Moreover, any proposed modification of *Kiuchi* to perform the “determining” step at the alleged DNS proxy module—the client-side proxy—would be improper because doing so would render *Kiuchi*’s system inoperable and unsatisfactory for its intended purpose. (Keromytis Decl. ¶ 99.) *Kiuchi*’s C-HTTP name server plays a crucial intermediary role in providing the client- and server-side proxies with each other’s public keys and Nonce values, which would be defeated if the client-side proxy performed these functions. (*Kiuchi* 64, 65.) Indeed, only after a series of interactions and exchanges mediated by the C-HTTP name server may the client- and server-side proxies even obtain these items and establish a connection. (*Id.* at 66.) This intermediary role of the C-HTTP name server enables many of *Kiuchi*’s security features, as it makes it “impossible to know the IP address of a server-side proxy even if its C-HTTP hostname (not necessarily the same as its DNS name) is known and vice versa,” presenting several additional hurdles for would-be hackers. (*Id.* at 68; Keromytis Decl. ¶ 99.) Thus, *Kiuchi* does not disclose or suggest that the client-side proxy performs the “determining” step recited in claim 1, and any proposed modification to remedy this shortcoming would render *Kiuchi*’s system inoperable and unsatisfactory for its intended purpose. (Keromytis Decl. ¶ 99.) Thus, there is no particular reason that would have led a person of ordinary skill to modify *Kiuchi* to include these features. M.P.E.P. § 2141(III)(G).

Edwards also fails to disclose or suggest “determining whether the intercepted DNS request corresponds to a secure server,” and therefore does not remedy the deficiencies of *Kiuchi* with respect to claim 1. Indeed, *Edwards* merely discloses determining whether an intercepted name request specifies an *available target*. (*Edwards* 933; *see also* Cisco Req. Ex. E-5 at 10-11.) But an “available target” in *Edwards* does not correspond to a “secure server” as alleged by the Request because this is directly contradicted by *Edwards* itself and by the Request’s own assertions. (Keromytis Decl. ¶ 100.) For example, while the Request asserts that “Edwards describes the list of

available targets as being only those services that have authentication and authorization enabled,” *Edwards* explicitly teaches that an administrator can remove these access restrictions from an available target:

When a target is made available, authentication and authorization are enabled for that target; this is indicated by the ‘AA’ before the name in the “Available Targets” list. *Once the target is available, the administrator can adjust the access control as required.*

(*Edwards* 933, emphasis added; Keromytis Decl. ¶ 100.) Indeed, the Request points to this very same passage to show an alleged example of a *nonsecure* computer in *Edwards* for a different element of claim 1. (Cisco Req. Ex. E-5 at 12-13.) Thus, a target’s presence on the “Available Targets” list has absolutely nothing to do with whether it is a “secure server,” since an administrator may disable any and all security measures for communicating with that target. (Keromytis Decl. ¶ 100.) For all of these reasons, merely determining whether a name request specifies an available target does not disclose determining whether a DNS request corresponds to a secure server, as recited in claim 1.

Accordingly, *Kiuchi* and *Edwards*, alone or in combination, do not disclose or suggest the “determining” step of claim 1, and the § 103(a) rejection of claim 1 should be withdrawn.

(iii) The Combination of *Kiuchi* and *Edwards* Fails to Disclose or Suggest “Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

As explained above, *Kiuchi* fails to disclose, either explicitly or inherently, that its C-HTTP connections are automatically initiated. *Edwards* does not remedy this deficiency of *Kiuchi* with respect to claim 1, nor does the Request contend that it does. (Cisco Req. Ex. E-5 at 16-17, relying solely on *Kiuchi* for this claim feature.) Thus, *Kiuchi* and *Edwards*, alone or in combination, do not disclose or suggest the “automatically initiating” feature of claim 1, and the § 103(a) rejection of claim 1 should be withdrawn, and its patentability confirmed.

c. Independent Claims 7 and 13

Independent claim 7 recites features similar to those described above for claim 1. For example, claim 7’s recited feature of “a domain name server (DNS) proxy module . . . intercepting a DNS request sent by a client” is similar to the “domain name server (DNS) proxy module that intercepts DNS requests sent by a client” feature of claim 1. Also, claim 7’s recited feature of “determining whether the intercepted DNS request corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Furthermore, claim 7’s recited feature of “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel

between the client and the secure server,” is similar to the “automatically initiating” feature of claim 1, also discussed above. Accordingly, *Kiuchi* and *Edwards* do not disclose or suggest these features of claim 7 for similar reasons as those discussed above with respect to claim 1.

Independent claim 13 also recites features similar to those described above for claim 1. For example, claim 13’s recited “domain name server (DNS)” feature is similar to the “domain name server (DNS)” feature of claim 1. Also, claim 13’s recited “determining whether a DNS request sent by a client corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Independent claim 13 also recites features that differ from the features recited in claim 1. For example, claim 13 recites “when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel” But given the arguments presented in the Cisco Request and the Office Action (Cisco Req. Ex. E-5 at 30-31), however, Patent Owner asserts that *Kiuchi* and *Edwards* do not disclose or suggest these features of claim 13 for similar reasons as those discussed above with respect to claim 1.

Thus, for these reasons, Patent Owner requests that the rejection of claims 7 and 13 under § 103(a) be withdrawn, and their patentability confirmed.

d. Dependent Claims 3, 9, and 15

Dependent claims 3, 9, and 15 ultimately depend from independent claims 1, 7, and 13, respectively, and include all of their features. Thus, claims 3, 9, and 15 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 3, 9, and 15 also distinguish over *Kiuchi* and *Edwards* for additional reasons. For example, dependent claims 3, 9, and 15 recite “when the client is not authorized to access the secure server, returning a host unknown error message to the client.” *Kiuchi* and *Edwards*, alone or in combination, do not disclose or suggest this feature.

As discussed above, *Kiuchi* fails to disclose, either explicitly or inherently, “when the client is not authorized to access the secure server, returning a host unknown error message to the client,” as recited in claim 1. The Office Action and the Cisco Request, however, assert that because a host’s secure C-HTTP hostname is not necessarily the same as a host’s DNS name, it would have been obvious to return a “host unknown error” when a client lacks authorization to access the secure C-HTTP host. (OA at 31; Cisco Req. Ex. E-5 at 20.) This is incorrect.

Kiuchi explains that when no connection is permitted between a client-side proxy and a server-side proxy, the C-HTTP name server sends a status code indicating an error to the client-side proxy. (*Kiuchi* 65.) Upon receiving the error status code, the client-side proxy then performs a DNS

lookup. (*Id.*) The Request asserts that as a result of this DNS lookup, a “host unknown error” may result, and that this satisfies the additional elements of claims 3, 9, and 15. (Cisco Req. Ex. E-5 at 20.) But any such potential “host unknown error” returned to the client would only occur if (1) either the server-side proxy is not registered in the closed network or is not permitted to accept the connection from the client-side proxy, and (2) the DNS lookup fails in a manner that causes a “host unknown error” (as opposed to other types of errors) to be returned to the client. (Keromytis Decl. ¶ 102.) Thus, given the stark differences in criteria between claim 1 and *Kiuchi* for returning host unknown error messages, *Kiuchi* fails to disclose or suggest returning a host unknown error message to the client *when the client is not authorized to access the secure server*, as recited in claim 1. (*Id.* at ¶ 103.)

Edwards does not make up for the deficiencies of *Kiuchi*. *Edwards* discloses that an “object not found” error will occur when a requested name “is not in the list of available targets.” (*Edwards* 933.) But this does not disclose returning an error when a client is not authorized to access a secure server. (Keromytis Decl. ¶ 104.) “Available targets” in *Edwards* are “services for which the object gateway has created service interceptors.” (*Edwards* 933; Keromytis Decl. ¶ 104.) The mere existence or nonexistence of an object in a list of available targets has nothing to do with whether a particular client is authorized to access that target, as “the administrator can adjust the access control” to either enable or disable authentication and authorization requirements for the target. (Keromytis Decl. ¶ 104.) Thus, *Edwards* also does not disclose returning an error message of any kind, let alone a host unknown error message, *when the client is not authorized to access the secure server*, as recited in claims 3, 9, and 15.

Despite the fact that *Kiuchi* and *Edwards* do not disclose returning a host unknown error message as recited in claims 3, 9, and 15, the Request includes several paragraphs asserting why it would have been obvious to “translate the idea of *Edwards*’ ‘object not found’ error into a ‘host unknown error’ in the system of *Kiuchi*.” However, as discussed above, even if *Kiuchi* and *Edwards* are combined in this manner, the combination still does not disclose returning a host unknown error *when the client is not authorized to access the secure server*, as recited in claim 1, given the alternative criteria of *Kiuchi* and the fact that an “available target” in *Edwards* has no bearing on any security status.

Thus, for at least the reasons provided above, *Kiuchi* and *Edwards* do not render claims 3, 9, and 16 obvious. Accordingly, Patent Owner requests that the rejection of claims 3, 9, and 15 be withdrawn and the claims confirmed.

e. Dependent Claims 6 and 12

Dependent claims 6 and 12 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, claims 6 and 12 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 6 and 12 also distinguish over *Kiuchi* and *Edwards* for additional reasons. For example, dependent claims 6 and 12 recite that the feature of automatically initiating the encrypted channel between the client and the secure server “avoids sending a true IP address of the secure server to the client.” *Kiuchi* and *Edwards*, alone or in combination, do not disclose or suggest this feature.

As discussed above, *Kiuchi* does not disclose or suggest avoiding “sending a true IP address of the secure server to the client” because *Kiuchi* in fact admits that it is possible to know the IP address of the secure server, and also because there is no reason why the client-side proxy (“a proxy server for external (outside the firewall) access,” i.e., is an *outward-looking* proxy) would block or prevent a true IP address from being sent by the alleged secure server to the client. *Edwards* does remedy these deficiencies.

The Office Action and the Cisco Request assert that *Edwards* discloses or suggests this claim feature because “[o]bject references will not be usable by outside clients unless they are references to service interceptors known to the proxy.” (OA at 31; Cisco Req. Ex. E-5 at 24.) This is incorrect. *Edwards* merely explains that this feature prevents the *client’s* object reference from being sent to other computers *outside* of the internal network. (*Edwards* 936; Keromytis Decl. ¶ 106.) Indeed, *Edwards* explains that this feature “prevents services in the internal network accidentally subverting security by handing object references to a client in the outside network.” (*Edwards* 936; Keromytis Decl. ¶ 106.) Thus, because *Edwards* is concerned with avoiding distributing an object reference of the *client* to computers “in the outside network,” rather than avoiding “sending a true IP address of the *secure server* to the client” (emphasis added), *Edwards* does not disclose or suggest the features of claims 6 and 12.

Thus, for at least the reasons provided above, *Kiuchi* and *Edwards* do not render claims 6 and 12 obvious. Accordingly, Patent Owner requests that the rejection of claims 6 and 12 be withdrawn and the claims confirmed.

f. Dependent Claims 2, 4, 8, 10, 14, and 16

Remaining claims 2, 4, 6, 8, 10, 12, 14, and 16 depend from one or more of allowable claims 1, 3, 7, 9, 13, and 15, and include all of their features. Thus, the combination of *Kiuchi* and *Edwards* does not render claims 2, 4, 6, 8, 10, 12, 14, and 16 obvious, and the rejections of claims 2, 4, 6, 8,

10, 12, 14, and 16 should be withdrawn and these claims should be confirmed for at least the reasons discussed above in connection with claims 1, 3, 7, 9, 13, and 15.

5. The Rejection of Claims 5 and 11 Under 35 U.S.C. § 103(a) Based on *Kiuchi* in View of *Edwards* and *Martin* Should Be Withdrawn (Issue 15)

The Office Action rejects claims 5 and 11 under 35 U.S.C. § 103(a) based on *Kiuchi* in view of *Edwards* and further in view of *Martin*. (OA at 32.) Claim 5 depends from independent claim 1, and claim 11 depends from independent claim 7. As explained above, *Kiuchi* and *Edwards*, alone or in combination, do not disclose or suggest the features of claims 1 and 7, and thus do not support the rejection of those claims. *Martin* does not remedy the deficiencies of *Kiuchi* and *Edwards* discussed above with respect to independent claims 1 and 7, nor do the Request and the Office Action assert that *Martin* does. Thus, the rejection of claims 5 and 11 under § 103(a) based on *Kiuchi* in view of *Edwards* and further in view of *Martin* should be withdrawn, and the claims should be confirmed.

Accordingly, Patent Owner respectfully requests that the rejection of claims 1-4, 6-10, and 12-16 under 35 U.S.C. § 102(b) based on *Kiuchi* and the rejection of claims 1-16 under 35 U.S.C. § 103(a) based on *Kiuchi* in view of *Edwards* and *Martin* be withdrawn, and the patentability of these claims be confirmed.

G. The Rejections Based on *Wesinger* Should Be Withdrawn

The Office Action rejects claims 1-4, 6-10, and 12-16 under 35 U.S.C. § 102(e) based on U.S. Patent No. 5,898,830 to *Wesinger* et al. (Cisco Req. Ex. D-2) ("*Wesinger*") and rejects claims 1-16 under 35 U.S.C. § 103(a) based on *Wesinger* in view of certain secondary references. For the reasons discussed below, these rejections should be withdrawn, and the claims should be confirmed.

1. The Rejection of Claims 1-4, 6-10, and 12-16 Under 35 U.S.C. § 102(e) Based on *Wesinger* Should Be Withdrawn (Issue 9)

The Office Action rejects claims 1-4, 6-10, and 12-16 under § 102(e) based on *Wesinger*. (OA at 29.) For the reasons discussed below, this rejection should be withdrawn, and the claims should be confirmed.

a. Overview of *Wesinger*

Wesinger relates to a "firewall providing enhanced network security and user transparency." (*Wesinger* Title.) The firewall "selectively allows 'acceptable' computer transmissions to pass through it and disallows other non-acceptable computer transmissions." (*Id.* at 1:8-12.)

In *Wesinger*, "[w]hen a connection request is received, the firewall spawns a process, or execution thread, to create a virtual host VHN to handle that connection request." (*Id.* at 15:9-12.) "Each virtual host has a separate configuration sub-file (sub-database) C1, C2, etc., that may be

derived from a master configuration file, or database, 510. The configuration sub-files are text files that may be used to enable or disable different functions for each virtual host, specify which connections and types of traffic will be allowed and which will be denied, etc.” (*Id.* at 14:46-52.) “Also as part of the configuration file of each virtual host, an access rules database is provided governing access to and through the virtual host, *i.e.*, which connections will be allowed and which connections will be denied.” (*Id.* at 15:24-28.) In contrast to determining whether access to a secure website or computer is requested, the process in *Wesinger* uses the access rules database to “allow only a connection from a specified secure client.” (*Id.* at 10:14-16.) Each virtual host in *Wesinger* is secure to the extent that it is supported by a firewall that implements a security policy for the virtual host. (*See, e.g., id.* at 7:59-8:15.)

In addition to explaining how connection requests are handled, *Wesinger* also discusses how DNS requests are handled:

When client C tries to initiate a connection to host D using the name of D, DNS operates in the usual manner to propagate a name request to successive levels of the network until D is found. The DNS server for D returns the network address of D to a virtual host on the firewall 155. The virtual host returns its network address to the virtual host on the firewall 157 from which it received the lookup request, and so on, until a virtual host on the firewall 105 returns its network address (instead of the network address of D) to the client C.

(*Wesinger* 9:16-24.)

b. Independent Claim 1

Independent claim 1 is directed to a data processing device comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client. *Wesinger* fails to disclose the combination of features recited in this claim for at least the reasons discussed below.

(i) *Wesinger* Does Not Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

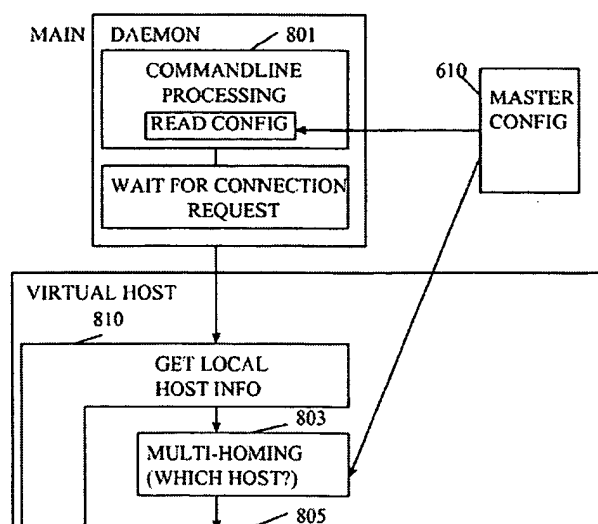
Independent claim 1 recites, among other things, “determining whether the intercepted DNS request corresponds to a secure server.” *Wesinger* fails to disclose this feature for the reasons provided below. In particular, *Wesinger* does not disclose performing the alleged determining with respect to a DNS request. Moreover, the alleged determining in *Wesinger* does not determine whether a request, much less a DNS request, corresponds to a secure server.

(a) Locating and Applying a Security Policy in *Wesinger* Does Not Disclose “Determining

Whether the Intercepted DNS Request Corresponds to a Secure Server”

The Cisco Request notes that each host name in *Wesinger* is associated with a security policy—access rules database 513 including an Allow portion 515 and/or a Deny portion 517. (Cisco Req. Ex. E-2 at 7, citing *Wesinger* 14:66-15:5.) According to *Wesinger*, “[u]sing the access rules database 513, the firewall selectively allows and denies connections to implement a network security policy.” (*Wesinger* 15:2-4.) The Request contends that “*Wesinger* describes locating and applying the network security policy for a requested domain name,” and “[i]f a network security policy exists for a particular host name, then the corresponding server is a ‘secure server’.” (Cisco Req. Ex. E-2 at 8, citing *Wesinger* 16:22-60.) Thus, the Request contends that locating and applying a security policy, the access rules database 513, in *Wesinger* discloses determining whether the intercepted DNS request corresponds to a secure server, as recited in claim 1. This is incorrect.

Wesinger discusses connection requests and DNS requests separately, but only discloses locating and applying the access rules database 513 when a *connection request* is received. (*Wesinger* 16:22; Keromytis Decl. ¶ 112.) Specifically, “[w]hen a connection request is received, the daemon spawns a process to handle the connection request. This process then . . . check[s] on the local side of the connection and the remote side of the connection to determine, in accordance with the appropriate Allow and Deny databases, whether the connection is to be allowed.” (*Wesinger* 16:22-28; *see also id.* at 15:5-19; Keromytis Decl. ¶ 112.) The flowchart in FIG. 8 of *Wesinger*, reproduced in part below, also shows that the firewall launches the process to check the access rules database 513 (i.e., the Allow and Deny databases) when a connection request is received.



(*Wesinger* FIG. 8, “wait for connection request”; Keromytis Decl. ¶ 112) Applying the access rules database 513 when a *connection request* is received, as taught by *Wesinger*, is distinct from determining whether the intercepted *DNS request* corresponds to a secure server, as recited in claim 1. (Keromytis Decl. ¶ 112.)

In addition to connection requests, *Wesinger* also discusses separate DNS requests. But *Wesinger* does not teach locating and applying the access rules database 513 based on a *DNS request*. (*Id.* at ¶ 113.) For example, with reference to FIG. 1, *Wesinger* describes how a DNS request is processed:

When client C tries to initiate a connection to host D using the name of D, DNS operates in the usual manner to propagate a name request to successive levels of the network until D is found. The DNS server for D returns the network address of D to a virtual host on the firewall 155. The virtual host returns its network address to the virtual host on the firewall 157 from which it received the lookup request, and so on, until a virtual host on the firewall 105 returns its network address (instead of the network address of D) to the client C.

(*Wesinger* 9:16-24; Keromytis Decl. ¶ 113) *Wesinger*'s DNS process, however, does not involve locating and applying the access rules database 513, which the Request contends is determining whether the intercepted *DNS request* corresponds to a secure server. (Keromytis Decl. ¶ 113.) *Wesinger* just explains that the name request propagates “in the usual manner” until the network address of D is found and returned to the virtual host on the firewall 155 of the destination D and the network address of the virtual host for the destination D on the firewall 105 is returned to the client C. (*Wesinger* 9:16-24; Keromytis Decl. ¶ 113.) The firewall does not locate or apply the security policy during this process. (Keromytis Decl. ¶ 113.)

Given that *Wesinger* discusses DNS requests and connection requests separately, but clearly only teaches locating and applying the access rules database 513 when a *connection request* is received, *Wesinger* does not disclose determining whether the intercepted *DNS request* corresponds to a secure server, as claimed. *Wesinger* is silent on locating or applying the access rules database 513 based on a DNS request.

**(b) Locating and Applying a Security Policy in
Wesinger Does Not Disclose Determining
Whether a Request Corresponds to a Secure
Server**

As discussed above, the Cisco Request contends that *Wesinger*'s firewall locating and applying the access rules database 513 discloses determining whether a request corresponds to a

secure server. But since this process occurs when the firewall receives a connection request and not based on any DNS request, locating and applying the access rules database 513 does not disclose determining whether the intercepted *DNS request* corresponds to a secure server, as recited in claim 1. Setting aside this DNS request issue, *Wesinger's* firewall system makes no determination whether a connection request (much less the claimed DNS request) corresponds to a secure server, as every server in *Wesinger* is supported by a firewall and presumably secure. (*Id.* at ¶ 114.)

As explained previously, *Wesinger* relates to a firewall system for providing enhanced network security and user transparency. (*See Wesinger* Title.) *Wesinger's* system, among other things, performs the firewall function of “selectively allow[ing] ‘acceptable’ computer transmissions to pass through it and disallow[ing] other non-acceptable computer transmissions.” (*Wesinger* 1:8-12.) Because *Wesinger's* is a firewall system, determining whether a connection request, let alone a DNS request, corresponds to a secure website is not a concern. (Keromytis Decl. ¶ 114.) Every server in *Wesinger* is secure to the extent it is behind a firewall that implements a security policy, so there is no need to determine whether a connection request (much less the claimed DNS request) corresponds to a secure server. Rather, *Wesinger's* firewall system performs the firewall function of scrutinizing the *remote host machine requesting the connection* (i.e., the client) to determine whether to allow or deny the requested connection. (*See Wesinger* 15:4-28, 16:22-28; Keromytis Decl. ¶ 114.)

Despite this shortcoming, the Request attempts to show that *Wesinger* discloses determining whether a request corresponds to a secure web site. (*See Cisco* Req. Ex. E-2 at 9-12.) Namely, the Request contends that “locating and applying the network security policy for a requested domain name” is determining whether a request corresponds to a secure server. (*Id.* at 8, “[i]f a network security policy exists for a particular host name, then the corresponding server is a ‘secure server’”; *id.* at 10, “the existence of a security policy indicates that the associated server computer is a ‘secure server,’”; *see also id.*, “*Wesinger* describes searching through the master configuration to identify whether a security policy exists for a requested domain or host name.”) The Request’s position is misplaced. Neither locating nor applying a security policy in *Wesinger* involves determining whether a connection request, much less a DNS request, corresponds to a secure server, because every server in *Wesinger* is secure to the extent it is behind a firewall that implements a security policy. (Keromytis Decl. ¶ 115.) This renders moot any determination whether a connection request, much less a DNS request, corresponds to a secure server. (Keromytis Decl. ¶ 115.)

The Request highlights a passage in *Wesinger* allegedly explaining how a security policy is “located.” (Cisco Req. Ex. E-2 at 8, citing *Wesinger* 16:22-60.) In that passage, *Wesinger* explains that when a connection request is received, “the master configuration database is scanned to see if a corresponding sub-database exists for that virtual host. If so, the sub-database is set as the configuration database of the virtual host.” (*Id.*, citing *Wesinger* 16:32-35; Keromytis Decl. ¶ 116.) Otherwise, “by default the master configuration database is used as the configuration database.” (*Wesinger* 16:36-38; Keromytis ¶ 116.) As such, in either scenario, the firewall sets a configuration database when it receives a connection request—either the sub-database or the master configuration database. (Keromytis Decl. ¶ 116.) *Wesinger* further explains that, “as part of the configuration file of each virtual host, an access rules database is provided governing access to and through the virtual host, *i.e.*, which connections will be allowed and which connections will be denied.”⁴ (*Wesinger* 15:24-28, emphasis added; *see also id.* at 14:46-54.) Thus, the firewall in *Wesinger* always “locates” a security policy for a virtual host, either the security policy defined by the sub-database or the security policy described by the master configuration database. (Keromytis Decl. ¶ 116.) Since *Wesinger* implements a security policy for each virtual host, determining whether a connection request (let alone a DNS request) corresponds to a secure server is not a concern.

At the same time, “applying” the located security policy in *Wesinger* does not involve determining whether a connection request (or a DNS request) corresponds to a secure server. . (*Id.* ¶ 117.) As discussed above, *Wesinger* discloses a firewall system that performs a firewall function to “selectively allow[] ‘acceptable’ computer transmissions to pass through it and disallow[] other non-acceptable computer transmissions.” (*Wesinger* 1:8-12.) *Wesinger* explains that the system “is preferably configured so as to allow only a connection from a specified *secure client*.” (*Id.* at 10:14-16, emphasis added.) To accomplish this, *Wesinger* explains that the firewall uses configuration files that “specify which *connections and types of traffic* will be allowed and which will be denied.” (*Id.* at 14:48-52, emphasis added.) Because *Wesinger*’s firewall applies the security policy to connection requests so that only connection requests from “specified secure client[s]” or certain “types of traffic” are allowed, at best, *Wesinger* analyzes whether *the remote host (i.e., client)* requesting a connection is secure. (Keromytis Decl. ¶ 117.) But the reference does not additionally disclose determining whether a connection request, much less a DNS request, corresponds to a secure *server*. (*Id.*)

⁴ *Wesinger* uses the terms “configuration file” and “configuration database” interchangeably. (*See, e.g., id.* at 14:45-48.)

The Request also points to *Wesinger* 16:22-60, contending that this passage discloses determining whether a connection request corresponds to a secure server. (Cisco Req. Ex. E-2 at 8.) The passage does not support that position. Rather, here, *Wesinger* explains how the firewall scrutinizes the *remote host* requesting a connection before allowing or denying the connection. (*Wesinger* 16:22-60; Keromytis Decl. ¶ 118.) As is clear from *Wesinger*'s explanation, the remote host corresponds to the *client* requesting a connection and not to the server. (Keromytis Decl. ¶ 118.) Thus, while *Wesinger*'s firewall may scrutinize the remote host requesting the connection to determine whether it is a "specified secure client" (*Wesinger* 10:14-16), the firewall does not additionally determine whether a connection request (or a DNS request) corresponds to a *secure server*. (Keromytis Decl. ¶ 118.) Indeed, *Wesinger* discloses a firewall system and simply does not contemplate determining whether a connection request corresponds to a secure server, as each is protected by a firewall and is presumably secure. (*Id.*)

To illustrate, FIG. 7 of *Wesinger*, reproduced below, shows an exemplary access rules database for several virtual host domains. (*Id.* ¶ 119.) The access rules database for each virtual host domain determines which specified secure clients are permitted to connect to that virtual host domain. (*Id.*) For example, the access rules database for the virtual host WWW.SANJOSE.NET indicates that connection requests from specified secure clients *.SRMC.COM, 205.138.192.* and 205.138.192.0/23 are allowed. (*Id.*) Similarly, the access rules database for the virtual host MJU.SRMC.COM indicates that connection requests from specified secure client 192.168.0.0/23 are allowed. (*Id.*)

```
PORT      = 80
RULE1     = {
    TIME   = "1AM-12PM"
}
WWW.SRMC.COM = {
    .CGI  = "PROCESSCGI"
    ROOT  = "/HOME/SRMC/HTML"
}
WWW.HONOLULU.NET = {
    .CGI  = ""
    ROOT  = "/HOME/HONOLULU/HTML"
}
WWW.SANJOSE.NET = {
    .CGI  = "PROCESSCGI"
    ALLOW = {
        *.SRMC.COM
        205.138.192.*
        205.138.192.0/23
    }
    DENY  = {
        MISTERPAIN.COM
    }
}
WWPROXY.SRMC.COM = {
    MODE  = RT_SERVERPROXY
}
NS.SRMC.COM = {
    ALLOW = {
        192.168.0.*
        192.168.1.* = RULE1
        192.168.2.* = {
            TIME   = "1AM-12PM"
        }
        192.168.3.* = 192.168.2.*
    }
}
MJU.SRMC.COM = {
    ALLOW = {
        192.168.0.0/23 = RULE1
    }
    DENY  = {
        192.168.0.* = {
            TIME   = "12PM-1AM"
        }
    }
}
}
```

FIG. 7

Thus, applying *Wesinger*'s access rules database may determine whether the *remote host* (the alleged client computer) requesting a connection is a secure client but does not determine whether a connection request, much less a DNS request, corresponds to a secure server. (*Id.*)

For at least the reasons provided above, *Wesinger* fails to disclose "determining whether the intercepted DNS request corresponds to a secure server," as recited by independent claim 1.

(ii) ***Wesinger* Fails to Disclose "When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer"**

Claim 1 further recites, "when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer." *Wesinger* also fails to disclose these features of claim 1.

As explained above, *Wesinger* does not disclose the claimed step of "determining whether the intercepted DNS request corresponds to a secure server." Thus, *Wesinger* cannot disclose doing anything dependent upon that step, much less forwarding the DNS request to a DNS function that

returns an IP address of a nonsecure computer when the intercepted DNS request does not correspond to a secure server, as claimed.

Additionally, *Wesinger* does not disclose the above features of claim 1 for the reasons below. As discussed in more detail below, the Request's anticipation analysis is improper because its treatment of the above "forwarding" features of claim 1 is inconsistent with its view of the "determining" features of claim 1. Further, the alleged determination that a request does not correspond to a secure server in *Wesinger* is not a determination with respect to a DNS request. Moreover, *Wesinger* does not disclose a relationship between the alleged forwarding of a DNS request in *Wesinger* and the alleged determination that a request corresponds to a secure server.

**(a) The Cisco Request's Analysis of the
"Forwarding" Feature Is Inconsistent with Its
Analysis of the "Determining" Feature**

As an initial matter, the Cisco Request's analysis of the "forwarding" element of claim 1 is inconsistent with its analysis of the "determining" element of claim 1. As discussed above, when analyzing the "determining" element of claim 1, the Request contends that "[i]f a network security policy *exists* for a particular host name, then the corresponding server is a 'secure server.'" (Cisco Req. Ex. E-2 at 8, emphasis added) But when turning to the "forwarding" element of claim 1, the Request changes its interpretation of a secure server. For that element, the Request takes the position that "[i]f the hostname is not in the rules database (*there is no entry in the Allow database and no entry in the Deny database*) then it does not correspond to a secure server." (*Id.* at 10, emphasis added, citing *Wesinger* 15:20-32.) However, it is possible for a security policy for a virtual host in *Wesinger* to exist without having any entries in its Allow and Deny databases.⁵ Following the position taken in the Request, then, such a server would be both secure and unsecure at the same time, which is impossible.

Thus, the Request's analysis regarding claim 1 under 35 U.S.C. § 102 is improper. By changing its interpretation of what an alleged secure server is in *Wesinger* from element-to-element within the same claim, the Request has not properly alleged, much less demonstrated, that *Wesinger* discloses each and every element of claim 1 as arranged in claim 1, and the rejection should be withdrawn. M.P.E.P. § 2131; *Net MoneyIN*, 545 F.3d at 1371. Instead, the analysis treats the claim "as mere catalogs of separate parts, in disregard of the part-to-part relationships set forth in the claims and that give the claims their meaning." *Therasense*, 593 F.3d at 1332.

⁵ As explained above, *Wesinger* teaches that the firewall implements a security policy for all virtual hosts.

(b) *Wesinger's Firewall Finding No Entries in the Allow and Deny Databases Does Not Disclose Determining "When the Intercepted DNS Request Does Not Correspond to a Secure Server"*

The Cisco Request contends that *Wesinger* discloses the claimed criterion "when the intercepted DNS request does not correspond to a secure server" because "[i]f the hostname is not in the rules database (there is no entry in the Allow database and no entry in the Deny database) then it does not correspond to a secure server." (Cisco Req. Ex. E-2 at 10, citing *Wesinger* 15:20-32.) This is incorrect.

As discussed above, the firewall in *Wesinger* only accesses and applies the access rules database when processing a *connection request*. The firewall thus could only determine the presence or absence of entries in the Allow and Deny databases for a particular virtual host upon receiving a connection request for that virtual host. (Keromytis Decl. ¶ 121.) Accordingly, even assuming that the absence of a hostname in the rules database in *Wesinger* somehow means that the hostname does not correspond to a secure server, as alleged, the firewall never makes such a determination with respect to a *DNS request*. (*Id.*) Thus, *Wesinger's* firewall determining the absence of an entry in the Allow and Deny databases for a virtual host does not disclose determining "when the intercepted *DNS request* does not correspond to a secure server" (emphasis added), as recited in claim 1.

(c) *Wesinger Discloses No Relationship Between the Firewall Finding No Entries in the Allow and Deny Databases and "Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer"*

Additionally, claim 1 recites "forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer" "*when* the intercepted DNS request does not correspond to a secure server" (emphasis added). The Request overlooks this connection between these two aspects of the claim feature.

The Request takes the position that the firewall in *Wesinger*, finding no entries in the Allow and Deny databases for a virtual host, discloses a scenario when the intercepted DNS request does not correspond to a secure server, as claimed. (Cisco Req. Ex. E-2 at 10.) Although the Request is incorrect for the reasons provided above, the Request correctly notes that the firewall allows the connection when it finds no entries in the Allow and Deny databases for the virtual host. (Keromytis Decl. ¶ 122.) The Request, however, goes further astray when it struggles to show a relationship between the firewall in *Wesinger* finding no entries in the Allow and Deny databases

(and allowing the connection) and forwarding a DNS request. (*Id.*) Specifically, the Request contends that “*Wesinger* further teaches that when the connection is allowed, the request is forwarded to a DNS function” that returns an IP address of a nonsecure computer. (Cisco Req. Ex. E-2 at 11, citing *Wesinger* 8:33-48; *see id.*, citing *Wesinger* 8:63-9:15.) This is simply not correct.

Wesinger discloses no relationship between the firewall finding no entries in the Allow and Deny databases for a virtual host, the alleged criterion of “when the intercepted DNS request does not correspond to a secure server, and forwarding any DNS request. (Keromytis Decl. ¶ 123.) Indeed, at the point the firewall in *Wesinger* checks the access rules database for a virtual host, finds no entries in the Allow and Deny databases, and allows the connection, any DNS processing (and the alleged forwarding) would have already occurred. (*Id.*)

Consistent with this, the passages of *Wesinger* that the Request cites as disclosing forwarding a DNS request do not show any relationship between the firewall finding no entries in the Allow and Deny databases, the alleged scenario “when the intercepted DNS request does not correspond to a secure server, and a DNS server forwarding a DNS request. (*Id.*) In fact, the DNS processing as described in *Wesinger* is independent of the firewall analyzing the access rules database of a virtual host. (*Id.*)

In the first DNS passage cited by the Request, *Wesinger* explains:

When a client needs a particular piece of information (e.g., the IP address of homer.odyssey.com), it asks its local DNS server for that information. *The local DNS server first examines its own local memory, such as a cache, to see if it already knows the answer to the client’s query. If not, the local DNS server asks other DNS servers, in turn, to discover the answer to the client’s query.*

(*Wesinger* 8:33-40, emphasis added.) Here, at best, *Wesinger* discloses that a local DNS server forwards a DNS request *when it does not have the answer in its own local memory.* (Keromytis Decl. ¶ 124.) But it certainly does not disclose forwarding a DNS request when the firewall checks the access rules database for a virtual host and finds no entries in the Allow and Deny databases—the alleged criterion of “when the intercepted DNS request does not correspond to a secure server” of claim 1. (*Id.*) Whether or not a DNS server forwards a DNS query in *Wesinger* depends on whether or not it has the answer to the query, not on the access rules database or on any determination that the intercepted DNS request does not correspond to a secure server. (*Id.*)

The Request cites the second passage for its disclosure of transparently “mapping” a DNS name to its “real” network address, which the Request contends discloses forwarding a DNS request. (Cisco Req. Ex. E-2 at 11, citing *Wesinger* 8:63-9:15.) Still, this passage describes no relationship

between the firewall finding no entries in the Allow and Deny databases, or even checking the access rules database at all, and mapping a DNS name to its real network address. (Keromytis Decl. ¶ 125.) A DNS server in *Wesinger* forwards a DNS query if it does not have the answer stored locally, not when the firewall finds no entries in the Allow and Deny databases, and certainly not when an intercepted DNS request does not correspond to a secure server. (*Id.*)

For at least the reasons above, *Wesinger* does not disclose “when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer,” as recited in claim 1. Further, because it ignores the claim criterion of “when the intercepted DNS request does not correspond to a secure server,” the Request’s anticipation analysis is improper because it does not even properly allege that *Wesinger* discloses each and every element “arranged or combined in the same way as recited in the claim,” as required by *Net MoneyIN*, 545 F.3d at 1371, but instead treats the claim “as mere catalogs of separate parts, in disregard of the part-to-part relationships set forth in the claims and that give the claims their meaning,” *Therasense*, 593 F.3d at 1332.

(iii) ***Wesinger* Fails to Disclose or Suggest “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”**

Independent claim 1 additionally recites “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.” *Wesinger* also fails to disclose these features of the claim for at least two reasons. First, *Wesinger* does not disclose a relationship between the alleged automatically initiating an encrypted channel and a DNS request. Second, setting aside the DNS request issue, *Wesinger* does not disclose a relationship between the alleged automatically initiating the encrypted channel and the alleged determining when a request corresponds to a secure server.

As explained above, *Wesinger* does not disclose the claimed step of “determining whether the intercepted DNS request corresponds to a secure server.” Thus, *Wesinger* cannot disclose doing anything dependent upon that step, much less automatically initiating an encrypted channel between the client and the secure server. Moreover, *Wesinger* does not disclose the above features of claim 1 because *Wesinger* does not disclose a connection between a DNS request and automatically initiating an encrypted channel.

The Request contends that *Wesinger* discloses an encrypted channel in two ways. (Cisco Req. Ex. E-2 at 13-14.) The Request first contends that “[t]he ‘protocol-based connection processing’ with ‘encryption’ shows initiating the encrypted channel” (*Id.*, citing *Wesinger* 17:1-7.) The Request also contends that “[c]ombining encryption capabilities with programmable transparency” amounts to an encrypted channel. (*Id.*, citing *Wesinger* 12:23-27.) But *Wesinger* does not teach a relationship between automatically initiating (a) the “protocol-based connection processing” required for the first alleged encrypted channel in *Wesinger* or (b) the “encryption” required for both of the alleged encrypted channels in *Wesinger* and a *DNS request*. (Keromytis Decl. ¶ 126.)

Regarding (a), the passage of *Wesinger* the Request relies upon explains that a virtual host performs the protocol-based connection processing “**once the connection has been allowed.**” (Cisco Req. Ex. E-2 at 13, citing *Wesinger* 17:1-7; Keromytis Decl. ¶ 127.) Earlier in the discussion, *Wesinger* makes clear the virtual host performs the protocol-based connection based on the result of processing a *connection request*. (*Wesinger* 16:22-67, “[w]hen a connection request is received”; Keromytis Decl. ¶ 127.) But *Wesinger* does not teach any link between a *DNS request* and automatically initiating the alleged encrypted channel, the protocol-based connection processing,

much less automatically initiating it “when the intercepted DNS request corresponds to a secure server,” as claimed. Confirming this, the DNS-related passages of *Wesinger* are silent regarding automatically initiating the cited protocol-based connection processing or any encrypted channel. (Keromytis Decl. ¶ 127.)

Similarly, regarding (b), *Wesinger* does not teach a relationship between automatically initiating *Wesinger*’s encryption, the alleged encrypted channel, and a DNS request. Just like the protocol-based connection processing, *Wesinger* explains that encryption is performed, if at all, as part of channel processing “*once the connection has been allowed.*” (Cisco Req. Ex. E-2 at 13, citing *Wesinger* 17:1-7; Keromytis Decl. ¶ 128.) Thus, like the protocol-based connection processing, *Wesinger* makes clear that the virtual host performs the referenced encryption based on the result of processing a *connection request*. (*Wesinger* 16:22-67, “[w]hen a connection request is received”; Keromytis Decl. ¶ 128.) But *Wesinger* also does not teach any link between a *DNS request* and the encryption, much less automatically initiating it “when the intercepted DNS request corresponds to a secure server,” as claimed. Even further dissociating *Wesinger*’s encryption and a DNS request, *Wesinger* explains that the channel processing, which includes the encryption, is an “optional” feature of the connection. (*Wesinger* 17:1-5; Keromytis Decl. ¶ 128.)

As discussed above, the Cisco Request contends that *Wesinger*’s firewall accessing and applying the access rules database corresponds to determining whether a DNS request corresponds to a secure server. (See Cisco Req. Ex. E-2 at 7-10.) And, in particular, it contends that “[t]he DNS request corresponds to a secure server if there is an entry in the [access] rules database.” (*Id.* at 13.) Notwithstanding the DNS request issue explained previously, the Cisco Request’s analysis of claim 1 goes further awry here because *Wesinger* does not disclose that entries in the access rules database have any bearing on whether *Wesinger*’s encryption is used, let alone on whether an encrypted channel is automatically initiated. (Keromytis Decl. ¶ 129.) In fact, *Wesinger* explains that its channel processing, which includes the protocol-based connection processing and encryption, depends upon the configuration of the virtual host, not on any entry in the access rules database. (See *Wesinger* 11:36-38; Keromytis Decl. ¶ 129.) Since *Wesinger*’s encryption is independent of any entry in the access rules database, *Wesinger* does not disclose automatically initiating an encrypted channel when the alleged intercepted DNS request allegedly corresponds to a secure server. (Keromytis Decl. ¶ 129.)

For at least the reasons above, *Wesinger* does not disclose “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” as recited in claim 1.

In light of the reasoning above, *Wesinger* does not anticipate claim 1, and the rejection of claim 1 should be withdrawn and the claim should be confirmed.

c. Independent Claims 7 and 13

Independent claim 7 recites features similar to those described above for claim 1. For example, claim 7’s recited feature of “determining whether the intercepted DNS request corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Also, claim 7’s recited feature of “when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer,” is similar to the “forwarding” feature of claim 1. Furthermore, claim 7’s recited feature of “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” is similar to the “automatically initiating” feature of claim 1, also discussed above. Accordingly, *Wesinger* does not disclose these features of claim 7 for similar reasons as those discussed above with respect to claim 1.

Independent claim 13 also recites features similar to those described above for claim 1. For example, claim 13’s recited “determining whether a DNS request sent by a client corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Also, claim 13’s recited feature of “when the DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer,” is similar to the “forwarding” feature of claim 1. Independent claim 13 also recites features that differ from the features recited in claim 1. For example, claim 13 recites “when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel” But given the arguments presented in the Cisco Request and the Office Action (Cisco Req. Ex. E-2 at 23-26), however, Patent Owner asserts that *Wesinger* does not disclose these features of claim 13 for similar reasons as those discussed above with respect to claim 1.

Thus, for these reasons, Patent Owner requests that the rejection of claims 7 and 13 under § 102(e) be withdrawn, and their patentability confirmed.

d. Dependent Claims 2, 8, and 14

Dependent claims 2, 8, and 14 depend from independent claims 1, 7, and 13, respectively, and include all of their features. Thus, claims 2, 8, and 14 should be confirmed for at least the

reasons discussed above with respect to those claims. Claims 2, 8, and 14 also distinguish over *Wesinger* for additional reasons. For example, dependent claims 2, 8, and 14 recite “determining whether the client is authorized to access the secure server; and (b) when the client is authorized to access the secure server, sending a request to the secure server to establish an [encrypted or secure] channel between the secure server and the client.” The Request has not demonstrated that *Wesinger* discloses these features of the claims because its analysis of these dependent claims is inconsistent with its analysis of the independent claims. Further, *Wesinger* does not disclose the claimed features.

**(i) The Request’s Analysis of Claims 2, 8, and 14 Is
Inconsistent with Its Analysis of Independent Claims 1, 7,
and 13**

Claim 2 recites, among other things, “determining whether the client is authorized to access the secure server.” According to the Request, *Wesinger*’s disclosure of “[c]hecking on ‘the host requesting the connection’ and applying ‘the appropriate level of access scrutiny’ shows determining whether the client is authorized to access the secure server as recited by the claim.” (*Id.* at 14-15, citing *Wesinger* 16:29-33, 48-67.) When analyzing claim 1, however, the Request contends that this same aspect of *Wesinger* means something else entirely. Specifically, the Request contends that *Wesinger*’s firewall checking on the host requesting the connection and applying the appropriate level of access scrutiny discloses determining whether the intercepted DNS request corresponds to a secure server, as recited in claim 1. (*See id.* at 8, citing *Wesinger* 16:22-60.) But when reaching claim 2, the Request changes its interpretation of *Wesinger* and the claims and contends that this same aspect of *Wesinger*, checking on the host requesting the connection and applying the appropriate level of access scrutiny, is determining whether the *client* is authorized to access the secure server. Checking on the *host requesting the connection* and applying the appropriate level of access scrutiny cannot disclose both determining whether the intercepted DNS request corresponds to a *secure server* and determining whether the *client* is authorized to access the secure server, as the Request alleges.

In light of this inconsistency in the Request’s analysis of claim 2 and its independent claim 1, the Request has not properly alleged, much less demonstrated, that *Wesinger* discloses each and every element “arranged or combined in the same way as recited in the claim,” as required by *Net MoneyIN*, 545 F.3d at 1371, but instead treats the claim “as mere catalogs of separate parts, in disregard of the part-to-part relationships set forth in the claims and that give the claims their meaning,” *Therasense*, 593 F.3d at 1332. The same applies for dependent claims 8 and 14, as the

Request largely incorporates by reference its analysis of claim 2 for claims 8 and 14. (See Cisco Req. Ex. E-2 at 21, 26.)

(ii) ***Wesinger* Does Not Disclose “Sending a Request to the Secure Server to Establish an Encrypted Channel When the Client Is Authorized to Access the Secure Server”**

Dependent claim 2 additionally recites “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client.” Claims 8 and 14 recite similar features. As discussed above, due to its inconsistent analysis of claims 1 and 2, the Cisco Request has not properly shown that *Wesinger* discloses determining whether the client is authorized to access the secure server, as recited in claim 2. And because the Request has not shown *Wesinger* to disclose such a determination of whether the client is authorized to access a secure server, it also has not shown *Wesinger* to disclose sending a request to the secure server to establish an encrypted channel *when the client is authorized to access the secure server*, as recited in claim 2.

The Request contends that *Wesinger* discloses “when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted channel between the secure server and the client,” as recited in claim 1. (*Id.* at 15-16.) In support of its position, the Request points to claim 1 of *Wesinger*, which states “issuing a request for a connection from the first computer to the second computer.” (*Id.* at 16, citing *Wesinger* 17:32-35.) The Request then concludes, “[i]ssuing the request for connection from the first computer to the second computer shows that when the client is authorized to access the secure server, it sends a request to the secure server to establish an encrypted channel between the secure server and the client as recited by the claim.” (*Id.*) *Wesinger* does not support the Request’s position. (Keromytis Decl. ¶ 130.)

Nothing in claim 1 of *Wesinger* indicates that the recited “issuing a request for a connection from the first computer to the second computer” depends upon the client being authorized to access the secure server. (*Id.*) It just states “issuing a request for a connection from the first computer to the second computer” without any condition attached. (*Id.*)

Furthermore, as explained above, the Request takes the position that *Wesinger*’s disclosure of “[c]hecking on ‘the host requesting the connection’ and applying ‘the appropriate level of access scrutiny’ shows determining whether the client is authorized to access the secure server as recited by the claim.” (*Id.* at 14-15, citing *Wesinger* 16:29-33, 16:48-67.) But the firewall does not even check the host requesting the connection and apply the appropriate level of access scrutiny *until after it receives a connection request*. (See *Wesinger* 16:22-28, 15:5-19; Keromytis Decl. ¶ 132.) In other

words, in *Wesinger*, the connection request triggers checking the host requesting the connection and applying the appropriate level of access scrutiny (the alleged determining whether the client is authorized to access the secure server), not vice versa. *Wesinger* sends a connection request irrespective of whether the client is allegedly authorized, and it is not until after it receives a connection request that the firewall allegedly determines whether the client is authorized. (Keromytis Decl. ¶ 132.) Because it again ignores the claimed criterion of sending a request to the secure server to establish an encrypted channel “when the client is authorized to access the secure server,” the Request’s anticipation analysis is improper because it does not even allege that *Wesinger* discloses each and every element “arranged or combined in the same way as recited in the claim,” as required by *Net MoneyIN*, 545 F.3d at 1371, but instead treats the claim “as mere catalogs of separate parts, in disregard of the part-to-part relationships set forth in the claims and that give the claims their meaning,” *Therasense*, 593 F.3d at 1332.

In light of the reasoning above, *Wesinger* does not anticipate claim 2, and the rejection of claim 2 should be withdrawn and the claim should be confirmed. The same applies for dependent claims 8 and 14, as the Request largely incorporates by reference its analysis of claim 2 for claims 8 and 14. (See Cisco Req. Ex. E-2 at 21, 26.)

e. Dependent Claims 3, 9, and 15

Dependent claims 3, 9, and 15 each depend from one or more of claims 1, 2, 7, 8, 13, and 14, and include all of their features. Thus, claims 3, 9, and 15 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 3, 9, and 15 also distinguish over *Wesinger* for additional reasons. For example, these claims recite “when the client is not authorized to access the secure server, returning a host unknown error message to the client.”

The Cisco Request contends that *Wesinger* discloses these because claim 11 of *Wesinger* recites “if the requested connection is not allowed, refusing the connection.” (*Id.* at 17.) According to the Request, *Wesinger* refusing a connection would result in returning a host unknown error message to the client, as claimed, because RFC 1035, “Domain Names—Implementation and Specification” (“RFC 1035”) explains that “refusing a *DNS connection* results in a host unknown error message.” (Cisco Req. Ex. E-2 at 17, emphasis added, citing RFC 1035 at 27.) The Request, again, conflates connection requests and DNS requests. As explained with regard to claim 1, *Wesinger* distinguishes between connection requests and DNS requests, and separately discusses the functionality that occurs in response to each type of request. Here, the cited portion of claim 11 relates to a *connection request*, not a DNS request. (Keromytis Decl. ¶ 133.) Thus, even if RFC

1035 specifies that a refused *DNS request* results in a host unknown error message, this says nothing about what happens when a *connection request* is refused. (*Id.*) Further, by the time *Wesinger* sends a connection request, any DNS processing has already occurred and the virtual host has already been found and identified to the client, so no host unknown error message would be returned. (*Id.*) *Wesinger* augmented with the disclosure of RFC 1035 does not disclose returning a host unknown error message to the client when the client is not authorized to access the secure server, as claimed.

In light of the reasoning above, *Wesinger* does not anticipate claims 3, 9, and 15, and the rejection of these claims should be withdrawn and the claims should be confirmed.

f. Dependent Claims 6 and 12

Dependent claims 6 and 12 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, claims 6 and 12 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 6 and 12 also distinguish over *Wesinger* for additional reasons. For example, dependent claims 6 and 12 recite that “automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.” *Wesinger* does not disclose this feature.

The Cisco Request contends that *Wesinger*’s DNS process avoids sending a true IP address of the secure server to the client because it “returns the address of the virtual host so that the virtual host appears as the secure server ‘D’.” (Cisco Req. Ex. E-2 at 18-19, citing *Wesinger* 9:25-35.) As explained with respect to claim 1, however, *Wesinger* initiates encryption, if at all, upon processing a connection request and allowing the connection, not in relation to DNS. (See *Wesinger* 16:22-17:7; Keromytis Decl. ¶ 134.) Accordingly, *Wesinger* initiates the encryption separately from the DNS process the Request contends avoids sending a true IP address of the secure server to the client. (Keromytis Decl. ¶ 134.) Thus, the Request has not shown *Wesinger* to disclose that *automatically initiating the encrypted channel* between the client and the secure server avoids sending a true IP address of the secure server to the client, as claimed.

In light of the reasoning above, *Wesinger* does not anticipate claims 6 and 12, and the rejection of these claims should be withdrawn and the claims should be confirmed.

g. Remaining Dependent Claims 4, 10, and 16

Dependent claims 4, 10, and 16 depend from one or more of allowable claims 1, 3, 7, 9, 13, and 15, and thus include all of the features of the claims from which they depend. *Wesinger* does not render claims 4, 10, and 16 obvious, and the rejections of claims 4, 10, and 16 should be withdrawn

and these claims should be confirmed at least for the reasons discussed above in connection with claims 1, 3, 7, 9, 13, and 15.

2. The Rejection of Claims 5 and 11 Under 35 U.S.C. § 103(a) Based on *Wesinger* in View of *Martin* Should Be Withdrawn (Issue 9)

The Office Action rejects claims 5 and 11 under § 103(a) based on *Wesinger* in view of *Martin*. (OA at 29.) Claim 5 depends from independent claim 1, and claim 11 depends from independent claim 7. As explained above, *Wesinger* does not disclose or suggest the features of claims 1 and 7, and thus do not support the rejection of those claims. The above-listed rejection of claims 5 and 11 should also be withdrawn and the claims should be confirmed because *Martin* does not remedy the deficiencies of *Wesinger* discussed above with respect to independent claims 1 and 7. Nor do the Cisco Request and the Office Action assert that *Martin* does. Accordingly, the rejection of claims 5 and 11 under § 103 based on *Wesinger* and in view of *Martin* should be withdrawn and the claims should be confirmed.

3. The Rejection of Claims 1-4, 6-10, and 12-16 Under 35 U.S.C. § 103(a) Based on *Wesinger* in View of *Edwards* Should Be Withdrawn (Issue 16)

The Office Action rejects claims 1-4, 6-10, and 12-16 under § 103(a) based on *Wesinger* in view of *Edwards*. (OA at 32.) For the reasons discussed below, this rejection should be withdrawn and the claims should be confirmed.

1. Independent Claim 1

Independent claim 1 is directed to a data processing device storing a domain name server (DNS) proxy module. *Wesinger* and *Edwards*, alone or in combination, fail to disclose or suggest the combination of features recited in this claim for at least the reasons discussed below.

a. The Combination of *Wesinger* and *Edwards* Fails to Disclose or Suggest “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

Independent claim 1 recites, “determining whether the intercepted DNS request corresponds to a secure server.” The combination of *Wesinger* and *Edwards* does not disclose or suggest this feature.

As explained above with respect to the rejection under 35 U.S.C. § 102(b) based on *Wesinger*, *Wesinger* does not disclose or suggest determining whether the intercepted DNS request corresponds to a secure server. And for the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*, the Cisco Request repeats its assertions regarding *Wesinger* for this claim feature. (*Compare* Req. Ex. E-2 at 7-10 *with* Req. Ex. E-6 at 9-12.) Thus, Patent Owner’s arguments above

regarding *Wesinger* apply equally to the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*.

Moreover, *Edwards* does not make up for the deficiencies of *Wesinger*. In fact, the Request appears to admit that *Edwards* does not even intercept *DNS requests*, but instead asserts that *Edwards* “intercepts name service requests.” (Cisco Req. Ex. E-6 at 7.) But the name service requests of *Edwards* are not *DNS requests*, as they do not request an IP address or network address of any kind. (Keromytis Decl. ¶ 135.) Instead, the name service requests are requests to return a name of “services which are accessible from the back-end of the web server.” (*Edwards* 932.) Instead of receiving a network address or IP address, the requesting device merely receives “a reference[] to a service interceptor,” which it later invokes. (*Edwards* 932; Keromytis Decl. ¶ 135.) In fact, the Request does not assert that *Edward’s* name service request is a *DNS request*. Thus, *Edwards* cannot make up for these deficiencies of *Wesinger* because *Edwards* also does not disclose determining whether the intercepted *DNS request* corresponds to a secure server, as claimed.

Setting aside the *DNS request* issue, the Request additionally contends that *Edwards* discloses determining whether the name service request corresponds to a secure server. (Cisco Req. Ex. E-6 at 12-13, citing *Edwards* 933.) This is incorrect. *Edwards* discloses determining whether an intercepted name request specifies an *available target*. (*Edwards* 933.) But an “available target” in *Edwards* does not correspond to a “secure server.” (Keromytis Decl. ¶ 135.) The Request’s assertions to the contrary are incorrect because they directly contradict both *Edwards* and other assertions in the Request itself. (*Id.*) For example, the Request asserts that “*Edwards* describes the list of available targets as being *only those services* that have authentication and authorization enabled,” and that “a target with authentication and authorization enabled corresponds to a ‘secure server.’” (Cisco Req. Ex. E-6 at 13, emphasis added.) The Request is incorrect, however, that available targets are only those services that have authentication and authorization enabled because *Edwards* explicitly teaches that an administrator can remove these controls from an available target:

When a target is made available, authentication and authorization are enabled for that target; this is indicated by the ‘AA’ before the name in the “Available Targets” list. *Once the target is available, the administrator can adjust the access control as required.*

(*Edwards* 933, emphasis added.) Thus, an available target may or may not have authentication and authorization enabled. In fact, the Request points to this *same* passage to show an alleged example of a *nonsecure* computer in *Edwards* for a different element of claim 1. (Cisco Req. Ex. E-6 at 15-16, citing to the above passage in *Edwards* and asserting that by adjusting the access control, an

administrator can make a target *not* correspond to a secure server.) Accordingly, an available target in *Edwards* does not necessarily perform authentication and authorization, and merely determining whether a name request specifies an available target thus cannot determine whether the name request, much less a DNS request, corresponds to a secure server. (Keromytis Decl. ¶ 135.)

Accordingly, *Wesinger* and *Edwards*, alone or in combination, do not disclose or suggest determining whether a DNS request corresponds to a secure server.

b. The Combination of *Wesinger* and *Edwards* Fails to Disclose or Suggest “When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer”

Claim 1 further recites, “when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer.” The combination of *Wesinger* and *Edwards* does not disclose or suggest this feature.

As explained above with respect to the rejection under 35 U.S.C. § 102(b) based on *Wesinger*, *Wesinger* does not disclose or suggest when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer. And for the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*, the Cisco Request repeats its assertions regarding *Wesinger* for this claim feature. (Compare Cisco Req. Ex. E-2 at 10-12 with Cisco Req. Ex. E-6 at 13-15.) Thus, Patent Owner’s arguments above regarding *Wesinger* apply equally to the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*.

Moreover, *Edwards* does not remedy the deficiencies of *Wesinger* with respect to the above features of claim 1. As explained above, neither reference discloses or suggests the claimed step of “determining whether the intercepted DNS request corresponds to a secure server.” Thus, the combination cannot disclose doing anything dependent upon that step, much less forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer when the intercepted DNS request does not correspond to a secure server, as claimed.

Initially, as explained above, *Edwards* discloses receiving a name service request, not intercepting a DNS request. And due to its silence regarding a DNS request, *Edwards* cannot disclose forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer when the intercepted DNS request does not correspond to a secure server, as claimed.

When addressing this feature of claim 1, the Request notes that an administrator in *Edwards* can adjust the access controls for an available target by toggling its authentication and authorization

settings on and off. (Cisco Req. Ex. E-6 at 15-16, citing *Edwards* 933, Fig. 5.) The Request then contends that an available target with its authentication or authorization setting disabled “would not [sic] ‘not correspond to a secure server.’” (*Id.* at 16.) Still, nothing in the Request or in *Edwards* teaches that an available target with authentication or authorization disabled causes forwarding a DNS request at all, much less forwarding a DNS request to a DNS function that returns the IP address of a nonsecure computer. (Keromytis Decl. ¶ 137.)

Viewing (improperly) *Edwards*’s name service request as an intercepted DNS request does not salvage the Request’s position. (Keromytis Decl. ¶ 138.) Nothing in the Request or in *Edwards* shows that when an available target has authentication or authorization disabled, a name service request is forwarded anywhere, much less to a function that returns the address of a nonsecure computer. (*Id.*) Consistent with this, the Request concludes its analysis of *Edwards* after allegedly showing what a “nonsecure” server is, without attempting to show the interrelated feature of forwarding a DNS request. (Cisco Req. Ex. E-6 at 16-17.) At the same time, the Request also does not attempt to explain why it would have been obvious, in spite of the lack of disclosure in either reference, to forward a DNS request to a DNS function that returns an IP address of a nonsecure computer in the case that an administrator has disabled authentication or authorization for an available target. *Edwards* discloses no relationship between how name service requests (much less DNS requests) proceed and whether an available target’s authentication or authorization is enabled. (Keromytis Decl. ¶ 138.)

Accordingly, *Wesinger* and *Edwards*, alone or in combination, do not disclose or suggest “when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer,” as recited in claim 1.

c. The Combination of *Wesinger* and *Edwards* Fails to Disclose or Suggest “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

Claim 1 further recites, “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.” The combination of *Wesinger* and *Edwards* does not disclose or suggest this feature.

As explained above with respect to the rejection under 35 U.S.C. § 102(b) based on *Wesinger*, *Wesinger* does not disclose or suggest when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer. And for the rejection under 35 U.S.C. § 103(a) based on *Wesinger*

and *Edwards*, the Cisco Request repeats its assertions regarding *Wesinger* for this claim feature. (*Compare* Cisco Req. Ex. E-2 at 12-14 with Cisco Req. Ex. E-6 at 17-20.) Thus, Patent Owner's arguments above regarding *Wesinger* apply equally to the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*.

Moreover, *Edwards* does not remedy the deficiencies of *Wesinger* with respect to the above features of claim 1. As explained above, neither reference discloses or suggests the claimed step of "determining whether the intercepted DNS request corresponds to a secure server." Thus, the combination cannot disclose doing anything dependent upon that step, much less automatically initiating an encrypted channel between the client and the secure server, as claimed.

When addressing this feature of claim 1, the Request contends that a "secure server" in *Edwards* is an available target for which the administrator has enabled authentication and authorization. (*See* Cisco Req. Ex. E-6 at 17-18, citing *Edwards* 933.) The Request continues, "when enabled, an authorization check is performed for every request." (*Id.* at 18, citing *Edwards* 935.) Although the Request does not expressly state it, the Request apparently views *Edwards* performing an authorization check as automatically initiating an encrypted channel. The Request is incorrect at least because (1) *Edwards* does not disclose automatically initiating the authorization check based on a *DNS request*, and (2) performing the authorization check does not involve automatically initiating an encrypted channel. (Keromytis Decl. ¶ 140.)

First, the Request has mischaracterized the cited passage of *Edwards*. In that passage, *Edwards* does not disclose performing an authorization request for every *request*, as alleged, but "for each *invocation of service*." (*Edwards* 935, emphasis added; Keromytis Decl. ¶ 141) An invocation of service is not the same as a name service request, and is certainly not the same as a DNS request. (Keromytis Decl. ¶ 141.) *Edwards* discloses no relationship between performing the "authorization check" and a *DNS request*. (*Id.*) Indeed, as explained above and acknowledged by the Request, *Edwards* is silent regarding intercepting a DNS request in the first place. (*Id.*) Thus, *Edwards* does not disclose, "when the intercepted *DNS request* corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server" (emphasis added), as recited in claim 1.

Additionally, *Edwards* does not disclose that its "authorization check" involves automatically initiating an encrypted channel. (*Id.* at ¶ 142.) *Edwards* does not explain what exactly the authorization check entails, just that "the naming interceptor controls whether or not a service is available to any external clients." (*Edwards* 935.) But *Edwards* certainly does not teach that the

authorization check additionally involves automatically initiating an encrypted channel. (Keromytis Decl. ¶ 142.) Thus, even if *Edwards* disclosed performing an authorization check based on a DNS request (which it does not), *Edwards* still would not disclose “automatically initiating an *encrypted channel* between the client and the secure server” (emphasis added), as recited in claim 1. Additionally, the Request does not attempt to explain why, in spite of the lack of disclosure in either reference, these features of claim 1 nonetheless would have been obvious.

Accordingly, *Wesinger* and *Edwards*, alone or in combination, do not disclose or suggest, “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” as recited in claim 1.

2. Independent Claims 7 and 13

Independent claim 7 recites features similar to those described above for claim 1. For example, claim 7’s recited feature of “determining whether the intercepted DNS request corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Also, claim 7’s recited feature of “when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer,” is similar to the “forwarding” feature of claim 1. Furthermore, claim 7’s recited feature of “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” is similar to the “automatically initiating” feature of claim 1, also discussed above. Accordingly, *Wesinger* and *Edwards* do not disclose or suggest these features of claim 7 for similar reasons as those discussed above with respect to claim 1.

Independent claim 13 also recites features similar to those described above for claim 1. For example, claim 13’s recited “determining whether a DNS request sent by a client corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Also, claim 13’s recited feature of “when the DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer,” is similar to the “forwarding” feature of claim 1. Independent claim 13 also recites features that differ from the features recited in claim 1. For example, claim 13 recites “when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel” But given the arguments presented in the Cisco Request and the Office Action (Cisco Req. Ex. E-6 at 31-34), however, Patent Owner asserts that *Wesinger* and *Edwards* do not disclose or suggest these features of claim 13 for similar reasons as those discussed above with respect to claim 1.

Thus, for these reasons, Patent Owner requests that the rejection of claims 7 and 13 under § 103(a) be withdrawn, and their patentability confirmed.

3. Dependent Claims 2, 8, and 14

Dependent claims 2, 8, and 14 depend from independent claims 1, 7, and 13, respectively, and include all of their features. Thus, claims 2, 8, and 14 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 2, 8, and 14 also distinguish over *Wesinger* and *Edwards* for additional reasons. For example, dependent claims 2, 8, and 14 recite “when the client is authorized to access the secure server, sending a request to the secure server to establish an [encrypted or secure] channel between the secure server and the client.”

As explained above with respect to the rejection under 35 U.S.C. § 102(b) based on *Wesinger*, *Wesinger* does not disclose or suggest this feature. And for the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*, the Cisco Request repeats its assertions regarding *Wesinger* for this claim feature, without reference to *Edwards*. (*Compare* Cisco Req. Ex. E-2 at 15-16 with Cisco Req. Ex. E-6 at 21-22.) Thus, Patent Owner’s arguments above regarding *Wesinger* apply equally to the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*. Additionally, *Edwards* does not make up for the deficiencies of *Aziz*, because *Edwards* also does not disclose or suggest this feature. Nor do the Request and the Office Action assert that it does. (*See, e.g.*, Cisco Req. Ex. E-6 at 21-22, relying only on *Wesinger* as allegedly disclosing this feature.)

For at least the reasons provided above, *Wesinger* and *Edwards* do not render claims 2, 8, and 14 obvious. Accordingly, Patent Owner requests that the rejection of claims 2, 8, and 14 be withdrawn and the claims be confirmed.

4. Dependent Claims 3, 9, and 15

Dependent claims 3, 9, and 15 each depend from one or more of claims 1, 2, 7, 8, 13, and 14, and include all of their features. Thus, claims 3, 9, and 15 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 3, 9, and 15 also distinguish over *Wesinger* and *Edwards* for additional reasons. For example, dependent claims 3, 9, and 15 recite “when the client is not authorized to access the secure server, returning a host unknown error message to the client.” *Wesinger* and *Edwards*, alone or in combination, do not disclose or suggest this feature.

As explained above with respect to the rejection under 35 U.S.C. § 102(b) based on *Wesinger*, *Wesinger* does not disclose returning a host unknown error message to the client when the

client is not authorized to access the secure server. And for the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*, the Cisco Request repeats its assertions regarding *Wesinger* for this claim feature. (*Compare* Cisco Req. Ex. E-2 at 16-18 *with* Cisco Req. Ex. E-6 at 22-23.) Thus, Patent Owner's arguments above regarding *Wesinger* apply equally to the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*.

Edwards does not make up for the deficiencies of *Aziz*. As pointed out in the Request, *Edwards* discloses that an "object not found" error will occur when a requested name "is not in the list of available targets." (Cisco Req. Ex. E-6, citing *Edwards* 933.) But this does not disclose returning an error when a client is not authorized to access a secure server. (Keromytis Decl. ¶ 144.) "Available targets" in *Edwards* are "services for which the object gateway has created service interceptors," but they do not correspond to a "secure server," as *Edwards* explicitly teaches that an administrator can remove authentication and authorization controls from an available target:

When a target is made available, authentication and authorization are enabled for that target; this is indicated by the 'AA' before the name in the "Available Targets" list. *Once the target is available, the administrator can adjust the access control as required.*

(*Edwards* 933, emphasis added; Keromytis Decl. ¶ 144.) In fact, the Request points to this *same* passage to show an alleged example of a *nonsecure* computer in *Edwards* for an element of claim 1. (Cisco Req. Ex. E-6 at 15-16, citing to the above passage in *Edwards* and asserting that by adjusting the access control, an administrator can make a target *not* correspond to a secure server.) Thus, the mere existence or nonexistence of an object in a list of available targets has nothing to do with whether or not a particular requesting client is authorized to access that target. (Keromytis Decl. ¶ 144.) Accordingly, *Edwards* does not disclose returning an error message of any kind, let alone a host unknown error message, *when the client is not authorized to access the secure server*, as recited in claims 3, 9, and 15.

Despite the fact that *Wesinger* and *Edwards* do not disclose a host unknown error message, the Request concludes, without any reasoning, that it would have been obvious to "translate the idea of *Edwards*' 'object not found' error into a 'host unknown error' in the system of *Wesinger*." (Cisco Req. Ex. E-6 at 23.) However, even if *Wesinger* and *Edwards* were combined in this manner, the combination still does not disclose returning a host unknown error when the client is not authorized to access the secure server, as recited in claim 3, 9, and 15.

For at least the reasons provided above, *Wesinger* and *Edwards* do not render claims 3, 9, and 15 obvious. Accordingly, Patent Owner requests that the rejection of claims 3, 9, and 15 be withdrawn and the claims be confirmed.

5. Dependent Claims 6 and 12

Dependent claims 6 and 12 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, claims 6 and 12 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 6 and 12 also distinguish over *Wesinger* and *Edwards* for additional reasons. For example, dependent claims 6 and 12 recite that “automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.” *Wesinger* and *Edwards*, alone or in combination, do not disclose or suggest this feature.

As explained above with respect to the rejection under 35 U.S.C. § 102(b) based on *Wesinger*, *Wesinger* does not disclose that automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client. And for the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*, the Cisco Request repeats its assertions regarding *Wesinger* for this claim feature. (*Compare* Cisco Req. Ex. E-2 at 18-19 *with* Cisco Req. Ex. E-6 at 25-26.) Thus, Patent Owner’s arguments above regarding *Wesinger* apply equally to the rejection under 35 U.S.C. § 103(a) based on *Wesinger* and *Edwards*.

Edwards does not make up for the deficiencies of *Wesinger*. The Request cites a portion of *Edwards* that allegedly discloses that services are available to clients only through the name service interceptor and proxy, and asserts that this “suggest[s] that their ‘true’ identification is not provided to the client.” (Cisco Req. Ex. E-6 at 26.) This does not disclose or suggest the recited feature for at least two reasons. First, *Edwards* does not disclose automatically initiating an encrypted channel between a client and a secure server. Nor do the Request and the Office Action assert that it does. (*See, e.g., id.* at 18-20, relying only on *Wesinger*.) Thus, *Edwards* cannot disclose that automatically initiating the encrypted channel avoids sending a true IP address of a secure server to a client, if *Edwards* does not disclose or suggest automatically initiating the encrypted channel in the first place. Second, *Edwards* does not disclose that the object references to the services are IP addresses.

In view of the above, *Wesinger* and *Edwards* do not disclose or suggest the features of claims 6 and 12. However, the Cisco Request asserts that in view of the teachings of *Wesinger* and *Edwards*, “it would be obvious to ‘avoid[] sending a true IP address of the secure server to the client.’” (*Id.* at 26.) This analysis is improper for at least three reasons. First, the Request only

addresses part of the claimed feature and does not assert that “*automatically initiating the encrypted channel between the client and the secure server* avoids sending a true IP address of the secure server to the client” (emphasis added). Merely stating that not sending a true IP address would be obvious does not address the claimed feature as a whole. M.P.E.P. § 2141.02 (“The claimed invention as a whole must be considered.”) Second, considering what “would *be* obvious” improperly fails to determine what would *have been* obvious “at the time of the invention.” M.P.E.P. §§ 2141.01, 2141.02. Third, citing to references that do not disclose or suggest a recited feature and merely stating that the combined teachings would render the recited feature obvious contravenes the requirement that “analysis supporting a rejection under 35 U.S.C. [§] 103 should be made explicit.” M.P.E.P. § 2142.

For at least the reasons provided above, *Wesinger* and *Edwards* do not render claims 6 and 12 obvious. Accordingly, Patent Owner requests that the rejection of claims 6 and 12 be withdrawn and the claims be confirmed.

6. Remaining Dependent Claims 4, 10, and 16

Dependent claims 4, 10, and 16 depend from one or more of allowable claims 1, 3, 7, 9, 13, and 15, and thus include all of the features of the claims from which they depend. The combination of *Wesinger* and *Edwards* does not render claims 4, 10, and 16 obvious, and the rejections of claims 4, 10, and 16 should be withdrawn and these claims should be confirmed at least for the reasons discussed above in connection with claims 1, 3, 7, 9, 13, and 15.

Accordingly, Patent Owner respectfully requests that the rejection of claims 1-4, 6-10, and 12-16 under 35 U.S.C. § 103(a) based on *Wesinger* in view of *Edwards* be withdrawn, and the patentability of these claims be confirmed.

7. The Rejection of Claims 5 and 11 Under 35 U.S.C. § 103(a) Based on *Wesinger* in View of *Edwards* and *Martin* Should Be Withdrawn (Issue 17)

The Office Action rejects claims 5 and 11 under § 103(a) based on *Wesinger* in view of *Edwards* and further in view of *Martin*. (OA at 32.) Claim 5 depends from independent claim 1, and claim 11 depends from independent claim 7. As explained above, *Wesinger* and *Edwards*, alone or in combination, do not disclose or suggest the features of claims 1 and 7, and thus do not support the rejection of those claims. The above-listed rejection of claims 5 and 11 should also be withdrawn and the claims should be confirmed because *Martin* does not remedy the deficiencies of the primary references discussed above with respect to independent claims 1 and 7. Nor do the Cisco Request and the Office Action assert that *Martin* does. Accordingly, the rejection of claims 5 and 11 under

§ 103 based on *Wesinger* in view of *Edwards* and further in view of *Martin* should be withdrawn and the claims should be confirmed.

Accordingly, Patent Owner respectfully requests that the rejection of claims 5 and 11 under 35 U.S.C. § 103(a) based on *Wesinger* in view of *Edwards* and in further view of *Martin* be withdrawn, and the patentability of these claims be confirmed.

H. The Rejection of Claims 1, 7, and 13 Under 35 U.S.C. § 102(e) Based on *Blum* Should Be Withdrawn (Issue 11)

The Office Action rejects claims 1, 7, and 13 under § 102(e) based on U.S. Patent No. 6,182,141 (“*Blum*”). (OA at 4.) For the reasons discussed below, this rejection should be withdrawn, and the claims should be confirmed.

1. Overview of *Blum*

Blum is generally related to a proxy server. (*Blum* 1:6-7.) In particular, *Blum* discloses “[a] layered service provider [that] intercepts a communications request from a client application in the native protocol of the communications request. If the communications request requests communication with a remote server, the layered service provider packages and forwards the communications request to a predetermined well-known port.” (*Id.* at 2:26-32.)

In *Blum*, “[a] transparent proxy application listening on the predetermined well-known port receives the communications request in the native protocol of the request and establishes communication with the remote server, such that communication between the client application and the remote server is tunneled bi-directionally through the transparent proxy.” (*Id.* at 2:32-37.) The layered service provider “directs communications requests from client applications to the transparent proxy such that the client programs themselves do not need to be configured to know about the transparent proxy in order to use the transparent proxy.” (*Id.* at 3:50-55.)

In contrast to determining whether a DNS request corresponds to a *secure* server, the process in *Blum* determines whether the communications request is directed to a *remote* server or a *local* communications service. (*Id.* at Fig. 1, block 105.) The client application of *Blum* is configured such that it “is not required to include proxy mode configuration capabilities, and the request for communication remains in its native protocol without being encapsulated or otherwise altered.” (*See, e.g., id.* at 3:56-59.) The process of *Blum* is thus concerned with routing service requests either locally or remotely, without regard to the security status of the destination server.

2. Independent Claim 1

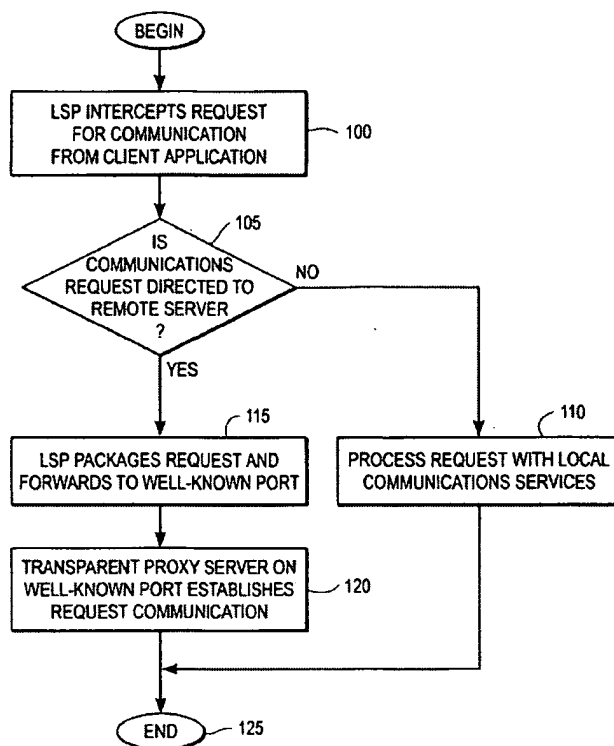
Independent claim 1 is directed to a data processing device storing a DNS proxy module. *Blum* fails to disclose the combination of features recited in this claim for at least the reasons discussed below.

a. *Blum* Fails to Disclose “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

Independent claim 1 recites, among other things, “determining whether the intercepted DNS request corresponds to a *secure* server” (emphasis added). *Blum* fails to disclose this feature, because as discussed below, *Blum* merely discloses determining whether a server is *local* or *remote*, and does not address whether the server is *secure*.

The Cisco Request and the Office Action assert that *Blum* discloses determining whether the intercepted DNS request corresponds to a secure server because *Blum* discloses communication requests that are sent to remote servers outside of the local area network and because *Blum* discloses establishing communications with the remote servers “such that communication between the client application and the remote server is tunneled bi-directionally through the transparent proxy.” (Cisco Req. Ex. E-3 at 4, citing *Blum* 2:32-37, 5:23-27.) Thus, the Request asserts that the “remote servers” are “secure servers” because they “require specific tunneling to reach.” (*Id.*) This assertion is incorrect for at least the two reasons discussed below.

First, *Blum* simply does not disclose making any determination as to the security status of the remote server. (Keromytis Decl. ¶ 149.) Instead, *Blum* only determines whether the server is remote or local, and then proceeds with processing the requests. In particular, according to *Blum*, “[i]n step 105, the LSP determines whether the communications request is directed to a *remote* server or to a *server on the local area network* (LAN) to which a computer system is hosting the client application is connected.” (*Blum* 3:32-23, emphasis added; Keromytis Decl. ¶ 149.) The flowchart in Fig. 1 of *Blum*, reproduced below, shows that the only potential determination made in the process of *Blum* is whether or not the communication is directed to a *remote* server.



Blum, therefore, does not determine whether the intercepted DNS request corresponds to a *secure* server. (Keromytis Decl. ¶¶ 149-150.)

Indeed, *Blum* does not address security of the remote servers at all. The only reference related to security in *Blum* is found in a passage in the section titled “Description of Related Art,” which states:

Currently available proxy servers have another issue in that specific code must be included in the proxy server to recognize and interpret each protocol that may be used by a client program. Commonly used protocols include Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Telnet, and *Secure Sockets Layer (SSL)*, for example.

(*Blum* 1:43-50, emphasis added; Keromytis Decl. ¶ 150.) Notably, *Blum* is describing “Related Art” (see, e.g., *id.* at 1:9), describing a list of “commonly used protocols” used in “currently available proxy servers” (*id.* at 1:43; Keromytis Decl. ¶ 150.). The passage makes mention of Secure Sockets Layer (SSL) as one of several protocols commonly used, to explain the need for the proxy server disclosed in *Blum*, stating, “what is needed is a proxy server application which does not require additional code or significant code revisions in order to support new or revised protocols. . . .” (*Blum* 2:15-21; Keromytis Decl. ¶ 150.) *Blum* indicates here that SSL is one of several common protocols used in client programs. (Keromytis Decl. ¶ 150.) However, *Blum* does not disclose using SSL or

any other security protocol in the disclosed communications processes with the remote servers. (*Id.*) Because *Blum* never addresses the topic of security in connection with the remote servers, they cannot be the recited secure servers.

Second, contrary to the Request's assertions, *Blum* does not disclose that remote servers "require specific tunneling to reach . . ." or that "[t]he DNS request corresponds to a secure server if it can *only* be accessed through the transparent proxy." (Cisco Req. Ex. E-3 at 4, 6, emphases added.) The Request cites different portions of *Blum* that indicate that the transparent proxy may bidirectionally tunnel communications between a client application and a remote server, and that connections may be established to a DNS service over a remote network. (*Id.* at 4, 6, citing *Blum* 2:32-37, 8:42-48, 8:57-64.) But these passages do not support the Request's position that the remote servers are secure because they *require* specific tunneling to reach and can *only* be accessed through a transparent proxy. (Keromytis Decl. ¶ 151.) Indeed, these passages simply do not disclose that a remote server may only be accessed through a transparent proxy. (*Id.*) Moreover, *Blum* teaches that the transparent proxy facilitates an *unsecured* socket-to-socket connection. For example, according to *Blum*, the tunneling function of the proxy is accomplished through the API tunneling LSP 425 such that a socket-to-socket connection can be established. (*Blum* 6:35-38.) "The API tunneling LSP 425 checks to see if the connection request *is directed to a local IP address* (i.e. an address on the LAN 310). If so, the connection request is passed through and handled directly by the TCP/IP transport service provider 340." (*Id.* at 9:15-17, emphasis added.) But *Blum* does not disclose that the transparent proxy provides any level of security to the remote server. (Keromytis Decl. ¶ 151.)

Therefore, for at least the reasons provided above, *Blum* fails to disclose or suggest "determining whether the intercepted DNS request corresponds to a secure server," as recited in claim 1.

b. *Blum* Fails to Disclose or Suggest "When the Intercepted DNS Request Does Not Correspond to a Secure Server, Forwarding the DNS Request to a DNS Function that Returns an IP Address of a Nonsecure Computer"

For the same reason that *Blum* fails to disclose or suggest "determining whether the intercepted DNS request corresponds to a secure server," *Blum* also fails to disclose or suggest determining "when the intercepted DNS request *does not* correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer . . ." (emphasis added). *Blum*'s silence as to security precludes determining the security status of any one or more servers and/or computers. As explained above, *Blum* makes no reference to the security of

the alleged secure servers. Therefore, *Blum* cannot disclose or suggest doing anything when the DNS request *does not* correspond to a secure server, at least because no determination is made in *Blum* as to the security of the alleged secure servers.

Thus, for at least the above reasons, *Blum* fails to disclose or suggest “when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer,” as recited in claim 1.

c. ***Blum* Fails to Disclose or Suggest “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”**

The Cisco Request and the Office Action assert that “*Blum* teaches automatically initiating an encrypted channel between the client and secure server,” and purport that “transparently establishing the connection to the remote server shows automatically establishing the channel between the client and the secure server.” (Cisco Req. Ex. E-3 at 6-7.) *Blum* fails to disclose these elements for at least the three reasons discussed below.

First, for the same reason that *Blum* fails to disclose or suggest “determining whether the intercepted DNS request corresponds to a secure server,” *Blum* also fails to disclose or suggest determining *when the intercepted DNS request corresponds to a secure server*, automatically initiating an encrypted channel. *Blum*’s silence as to security precludes determining the security status of any one or more servers and/or computers. As explained above, *Blum* makes no reference to the security of the alleged secure servers. Therefore, *Blum* cannot disclose or suggest doing anything when the DNS request corresponds to a secure server, at least because no determination is made in *Blum* as to the security of the alleged secure servers.

Second, *Blum* does not disclose that the alleged encrypted channel actually uses encryption. (Keromytis Decl. ¶ 153.) As discussed, *Blum* mentions the existence of Secure Socket Layer (SSL) in a list of common protocols *in the related art*, but does not disclose that SSL is used for the remote socket connections in the disclosed system. (*Id.*) Instead, *Blum* refers repeatedly to establishing socket connections, which could be interpreted by those skilled in the art as “raw IP” sockets, because the transport layer is bypassed, and the packet headers are made accessible to the application. (See, e.g., *Blum* 6:24-26, 6:28, 6:32, 6:34, 6:38, 8:15, 8:22, 8:43, 9:12, 9:31, 9:45; Keromytis Decl. ¶ 153.) Raw IP sockets are not encrypted. (Keromytis Decl. ¶ 153.) *Blum* makes multiple references to establishing socket connections, but never discloses that these connections are

encrypted. (*Blum* 1:43-50; Keromytis Decl. ¶ 153.) Therefore, *Blum* is silent as to initiating an encrypted channel.

Third, contrary to the Request's and the Office Action's assertions, *Blum* does not disclose using a "protocol filter" to encrypt the channel. The Request and the Office Action purport that "the channel is encrypted . . . through the use of a protocol filter." (Cisco Req. Ex. E-3 at 6-7.) This is incorrect. First, as discussed above, *Blum* does not disclose that the alleged channel is encrypted. Second, *Blum* does not disclose that the alleged channel is established with a protocol filter. The protocol filter in *Blum* merely limits certain types of communications and *interprets* protocols. (Keromytis Decl. ¶ 154.) Indeed, *Blum* states that "the transparent proxy application 355 checks to see if there is a protocol filter 520 associated with the native protocol of the connection request or with a port indicated in the connection request." (*Blum* 8:65-9:2.) *Blum* also discloses that

[p]rotocol filters 520, may also be stored on the server 305 in some embodiments to provide specific capabilities or functions in response to communications requests utilizing particular protocol(s) or ports . . . *However, for some protocols, a minimum amount of data interpretation may be necessary, and thus, a protocol filter 520 may need to be applied to provide the required interpretation.*

(*Id.* at 7:35-45, emphasis added.) An example of the interpretive function of protocol filters given in *Blum* is translating the browser in proxy mode to handle a request such as FTP encapsulated within HTTP by an encapsulation routine. (See, e.g., *id.* at 1:58-67; Keromytis Decl. ¶ 154.) Thus, *Blum* simply does not disclose using a protocol filter to encrypt anything, let alone to initiate an encrypted channel.

Moreover, even assuming arguendo that *Blum* discloses encrypting a channel with a protocol filter (which it does not), the Request and the Office Action improperly point to two different connections as allegedly disclosing "automatically initiating an encrypted channel." For example, the Request points to the connection between the transparent proxy application and *the DNS service* as allegedly teaching "automatically initiating," but point to a different connection between the transparent proxy application and *the remote server identified in the communications request* as allegedly teaching "encryption" using a protocol filter. (Compare Cisco Req. Ex. E-3 at 6, citing *Blum* 8:42-48, 8:57-64 with Cisco Req. Ex. E-3 at 7, citing *Blum* 9:33-46⁶.) This second connection relied on by the Request is made after the DNS request has been resolved by the DNS service. Thus, even under the Request's and the Office Action's incorrect interpretation of a protocol filter, *Blum*

⁶ The Cisco Request erroneously cites the quotation in *Blum* as being located at 8:65-9:2. The proper citation for the quoted part of *Blum* is 9:33-46.

still does not disclose automatically initiating an encrypted channel between the client and secure server.

For at least the reasons provided above, *Blum* does not anticipate all of the elements of claim 1, and the rejection should be withdrawn and the claim should be confirmed.

3. Independent Claims 7 and 13

Independent claim 7 recites features similar to those described above for claim 1. For example, claim 7's recited feature of "determining whether the intercepted DNS request corresponds to a secure server" is similar to the "determining" step of claim 1, discussed above. Also, claim 7's recited feature of "when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer," is similar to the "forwarding" feature of claim 1. Furthermore, claim 7's recited feature of "when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server," is similar to the "automatically initiating" feature of claim 1, also discussed above. Accordingly, *Blum* does not disclose these features of claim 7 for similar reasons as those discussed above with respect to claim 1.

Independent claim 13 also recites features similar to those described above for claim 1. For example, claim 13's recited "determining whether a DNS request sent by a client corresponds to a secure server" is similar to the "determining" step of claim 1, discussed above. Also, claim 13's recited feature of "when the DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer," is similar to the "forwarding" feature of claim 1. Independent claim 13 also recites features that differ from the features recited in claim 1. For example, claim 13 recites "when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel . . ." The arguments presented in the Cisco Request and the Office Action fail to appropriately address these differences, merely asserting without support that arguments "essentially identical" to those made with respect to claim 1 suffice. (*See* Cisco Req. Ex. E-3 at 10-11.) Thus, the rejection of claim 13 based on *Blum* is improper for failing to consider all of the words in the claim. M.P.E.P. § 2131; *id.* at § 2143.03 ("All words in a claim must be considered in judging the patentability of that claim against the prior art.") (emphasis added) (internal citations omitted). Moreover, to the extent the Requester and the Office later assert that the features recited in claim 13 are similar to the features recited in claim 1, Patent Owner asserts that *Blum* does not disclose or suggest these features of claim 13 for similar reasons as those discussed above with respect to claim 1.

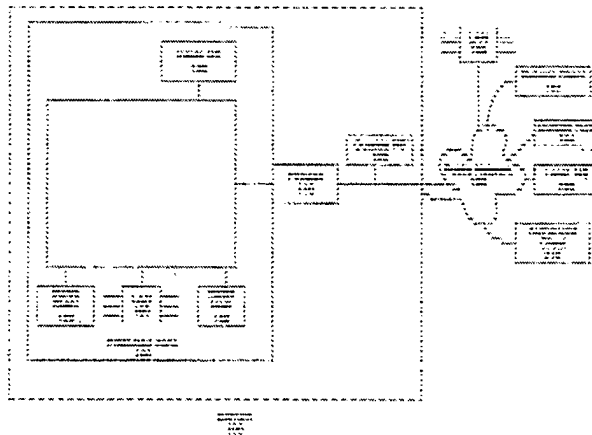
Thus, for these reasons, Patent Owner requests that the rejection of claims 7 and 13 under 35 U.S.C. § 102(e) be withdrawn, and their patentability confirmed.

**I. The Rejection of Claims 1-4, 6-10, and 12-16 Under 35 U.S.C. § 103(a)
Based on *Aziz* in View of *Edwards* Should Be Withdrawn (Issue 12)**

The Office Action rejects claims 1-4, 6-10, and 12-16 under § 103(a) based on U.S. Patent No. 6,119,234 ("*Aziz*") in view of *Edwards*. (OA at 30.) For the reasons discussed below, this rejection should be withdrawn and the claims should be confirmed.

1. Overview of *Aziz*

Aziz discloses a system "for dynamically configuring authorized clients with the address of a protected host and the key and address of an intermediate device (e.g., encrypting firewall, encrypting router, secure gateway) which is protecting a number of hosts on a private network" behind the intermediate device. (*Aziz*, 4:3-9.) Fig. 1 of *Aziz*, reproduced below, illustrates the system:



Aziz explains that "outside NS" 120 may receive a query for a host address located within domain 100 and may check its database for an SX record and the requested host name. (*Id.* at 9:49-53.) An SX record is a resource record that "contains the identifier (e.g., name or address) of a 'secure exchanger,'" such as firewall 110. (*Id.* at 6:23-40.) If an SX record exists, then outside NS 120 may include the SX record in the response to the requester, which may also include the requested host address, if available. (*Id.* at 9:54-10:5.)

Aziz also discloses a resolver 225, which is included in the "authorized client" 210 (*id.* at 8:5-50, Figs. 2A-2C), and receives a response to the query for a host address (*id.* at 10:39-41). If the response includes an SX record and the requested host address, then resolver 225 creates a tunnel map entry that provides the information "authorized client" 210 needs to encrypt messages to "inside

host” 140. (*Id.* at 11:13-60.) Resolver 225 then returns the requested host address to an application 215, also located in “inside host” 210. (*Id.* at 11:55-60.) According to *Aziz*, “[t]his completes the execution” of the configuration process. (*Id.* at 11:60-62.)

2. Independent Claim 1

Independent claim 1 is directed to a data processing device storing a domain name server (DNS) proxy module. *Aziz* and *Edwards*, alone or in combination, fail to disclose or suggest the combination of features recited in this claim for at least the reasons discussed below.

a. **The Combination of *Aziz* and *Edwards* Fails to Disclose or Suggest a “Data Processing Device . . . Storing a Domain Name Server (DNS) Proxy Module” that Performs All of the Recited Features**

Independent claim 1 recites a data processing device storing a DNS proxy module that performs all of the recited features of claim 1. The rejection of claim 1 should be withdrawn because it does not point to a data processing device that performs all of these features, but instead mixes and matches among different components in *Aziz* without providing any reason why it would have been obvious to combine these different components into a data processing device.

The Cisco Request initially asserts that the “Domain Name Server” of *Aziz* and, particularly, local NS 250 of *Aziz* is the DNS proxy module. (*See* Cisco Req. Ex. E-4 at 4-5.) The Office Action, however, rejects this position and instead asserts that the “resolver [225] of *Aziz* represents a DNS proxy module.” (OA at 30-31.) The Office Action then adopts and incorporates by reference the remainder of the proposed rejection in the Request. (*Id.* at 30.) Thus, the Office Action initially asserts that resolver 225 is the DNS proxy module, but the remainder of the rejection cites to *outside NS 120, and not resolver 225 or local NS 250*, as allegedly performing every recited feature in the remainder of the claim, except for “automatically initiating an encrypted channel between the client and the secure server,” which the adopted portion of the Request asserts is performed by resolver 225. (*See, e.g.*, Cisco Req. Ex. E-4 at 8, 10-12, 14-15, citing to outside NS 120 for all other recited features.)

Moreover, the Request and the Office Action do not explain how it would have been obvious to combine the two separate elements relied on by the Office Action (outside NS 120 and resolver 225) into a data processing device including a DNS proxy module, as recited in claim 1. Indeed, in all three embodiments shown in Figs. 3A-3C of *Aziz*, resolver 225 is included in authorized client 210, which is separate from outside NS 120. (*See Aziz* Figs. 3A-3C; *see also id.* at Fig. 1, showing authorized client 210 and outside NS 120 as separate components.)

Any proposed modification of *Aziz* to include resolver 225 in NS 120 and thus outside of authorized client 210 would be improper because doing so would render *Aziz*'s system inoperable and unsatisfactory for its intended purpose. M.P.E.P. § 2143.01(V). Resolver 225 is responsible for forwarding the query from authorized client 210 to local NS 250, receiving the response from outside NS 120, and for updating tunnel map entry 500 at the authorized client 210 based on the response. (See *Aziz* 8:22-24, 10:39-41, 11:16-18; Keromytis Decl. ¶ 159.) If resolver 225 were located at outside NS 120 instead of at authorized client 210, resolver 225 would be unable to forward a query from authorized client 210, receive a response from outside NS 120, or update tunnel map entry 500 at the authorized client 210 based on the response. (Keromytis Decl. ¶ 159.) Thus, moving resolver 225 to be located at "outside NS" 120 would render *Aziz* inoperable. (*Id.*)

Aziz does not include any teaching to the contrary. Although *Aziz* indicates that resolver 225's functionality can be customized "regardless of how many components are used to implement such functionality, or where those components may be located," *Aziz* does not in fact disclose that resolver 225 can be located separate from authorized client 210. (*Aziz* 8:7-11; Keromytis Decl. ¶ 160.) Rather, *Aziz* contemplates that resolver 225 can be implemented using different arrangements of multiple components *all within* authorized client 210. (Keromytis Decl. ¶ 160.) Indeed, in the context of the remainder of the "Resolver Location" section (of which the quoted *Aziz* 8:7-11 is a part), it is clear that the different arrangements of the components making up resolver 225 all include resolver 225 located within authorized client 210. (Keromytis Decl. ¶ 160.) Specifically, the "Resolver Location" describes the three embodiments discussed above where resolver 225 includes various numbers of components located in different configurations, with all of them locating resolver 225 entirely within "authorized client" 210. (See *Aziz* 8:12-50, Figs. 2A-2C; Keromytis Decl. ¶ 160.) Nowhere in this description does *Aziz* disclose that resolver 225 may be located anywhere other than at "authorized client" 210. (Keromytis Decl. ¶ 160.) Furthermore, *Aziz* does not disclose that resolver 225 can be located *at outside NS 120*, the component that the Request relies on as allegedly disclosing features of claim 1. And, as discussed above, doing so would render *Aziz* inoperable. Thus, *Aziz*, when read as a whole, makes clear that resolver 225 and outside NS 120 are different components. (*Id.*)

The Request and the Office Action also rely on *Edwards* as allegedly disclosing a DNS proxy module (the "object gateway" in *Edwards*). (Cisco Req. Ex. E-4 at 5.) However, the Request and the Office Action do not assert, and *Edwards* does not disclose or suggest, that the object gateway performs all of the features recited in claim 1. (See, e.g., *id.* at 14-15, 16-18, not relying on *Edwards*

to allegedly disclose returning an IP address or automatically initiating an encrypted channel.) Nor do the Request and the Office Action explain how it would have been obvious to combine *Edwards*'s object gateway with *Aziz*'s outside NS 120 and resolver 225. And for at least the reasons discussed above with regard to *Aziz*, doing so would render the resulting system inoperable.

Thus, *Aziz* and *Edwards*, alone or in combination, do not disclose or suggest a single data processing device storing a DNS proxy module that performs all of the recited features. As such, the combination of *Aziz* and *Edwards* does not render obvious claim 1.

b. The Combination of *Aziz* and *Edwards* Fails to Disclose or Suggest “a Domain Name Server (DNS) Proxy Module that Intercepts DNS Requests Sent by a Client”

Independent claim 1 recites “a domain name server (DNS) proxy module that intercepts DNS requests sent by a client.” The combination of *Aziz* and *Edwards* does not disclose or suggest this feature. In particular, neither reference discloses or suggests intercepting DNS requests.

Aziz does not disclose this feature because *Aziz* merely discloses receiving, but not intercepting, DNS requests. (Keromytis Decl. ¶ 162.) The Request and the Office Action admit this deficiency by asserting that the alleged DNS proxy server module NS 120 merely “receives DNS requests.” (Cisco Req. Ex. E-4 at 7, emphasis added.) Instead, the Request asserts that “[i]t would have been obvious to modify the name server software of *Aziz* to additionally intercept name service requests, as taught by *Edwards*.” (*Id.*)

Edwards does not make up for the deficiencies of *Aziz* because *Edwards* also does not disclose intercepting *DNS requests*. In fact, the Request appears to admit that *Edwards* does not intercept *DNS requests*, but instead asserts that *Edwards* “intercepts name service requests.” (*Id.* at 6.) But *Edwards* does not disclose that its name service requests are DNS requests. (Keromytis Decl. ¶ 162.) The name service requests are requests for “services which are accessible from the back-end of the web server.” (*Edwards* 932.) Instead of receiving a network address or IP address in response to a name service request, the requesting device receives “a reference to a service interceptor,” which it later invokes. (*Id.*; Keromytis Decl. ¶ 162.) In fact, the Request does not assert that *Edward*'s name service request is a DNS request. Thus, *Edwards* does not make up for these deficiencies of *Aziz*.

Moreover, it would not have been obvious to modify *Aziz* in the way suggested by the Request, such that the outside NS 120 “additionally intercept[s] name service requests” (Cisco Req. Ex. E-4 at 7), because there would have been no reason for doing so. (Keromytis Decl. ¶ 163.) For example, the Request asserts that “combining the transparent encryption of *Aziz* with the interception

of name service requests as taught by *Edwards* would allow the *Aziz* network to provide its transparent encryption services with little or no client configuration required.” (Cisco Req. Ex. E-4 at 2.) Not only does the Request fail to explain how using *Edwards*’s interception techniques would reduce the amount of configuration required, *Aziz* already purports to achieve this goal:

Network administrators need a way to configure authorized clients with the addresses of protected hosts that *does not require human intervention to modify the configuration files on every authorized client*. The solutions provided by various embodiments of the invention will [solve this problem].

(*Aziz* 3:3-12, emphasis added; Keromytis Decl. ¶ 163.) Thus, one of ordinary skill in the art would not have found it obvious to invest the time, effort, and resources to make the proposed modifications to *Aziz* when doing so would not result in any improvement in the overall system. (Keromytis Decl. ¶ 163.) Moreover, to the extent the Request is suggesting that a device other than NS 120 would intercept the DNS requests, this modification of *Aziz* would render *Aziz* inoperable and unsatisfactory for its intended purpose because NS 120 would not receive the DNS request and thus would be unable to perform the process explained in Fig. 3 of *Aziz*. (*Id.*)

Thus, *Aziz* and *Edwards*, alone or in combination, do not disclose or suggest a DNS proxy module that intercepts DNS requests sent by a client. As such, the combination of *Aziz* and *Edwards* does not render obvious claim 1.

c. The Combination of *Aziz* and *Edwards* Fails to Disclose or Suggest “Determining Whether the Intercepted DNS Request Corresponds to a Secure Server”

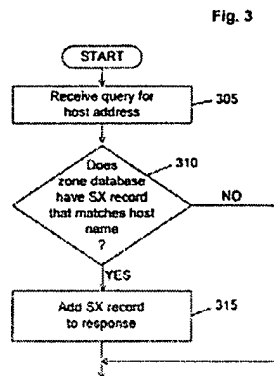
Independent claim 1 recites “determining whether the intercepted DNS request corresponds to a secure server.” The combination of *Aziz* and *Edwards* does not disclose or suggest this feature.

Aziz does not disclose or suggest determining whether the intercepted DNS request corresponds to a secure server. The Office Action and the Cisco Request assert that *Aziz* discloses this feature “by checking whether the request has an associated ‘secure exchanger’ or ‘SX’ record.” (OA at 30-31; Cisco Req. Ex. E-4 at 10.) But merely checking for the existence of an SX record is not the same as determining whether a DNS request corresponds to a secure server. (Keromytis Decl. ¶ 165.)

Aziz explains that SX records are just another type of resource record. (*Aziz* 6:25-28.) And NS 120 behaves like any other name server—when it receives a request, it checks to see what types of records it has for the name in the request, such as address records (“A records”) or SX records. (Keromytis Decl. ¶ 165.) If it has an SX record, it simply adds it to the response it will send to the client, much like it does with A records:

At step 310, outside NS 120 *checks if its zone database has an SX record* with an owner name that matches the requested host name. If the database does not have such a record, execution jumps to step 320. If the database does, at step 315, outside NS 120 adds the SX record identifying the secure exchanger for the requested host to the response.

(*Aziz* 9:49-55, emphasis added; Keromytis Decl. ¶ 165; *see also Aziz* Fig. 3, reproduced in part below, where NS 120 merely checks for the existence of an SX record in step 310, but does not determine whether a website is secure.)



Thus, the portions of *Aziz* relied on by the Request and the Office Action merely disclose checking for the existence of an SX record for a particular host name, but *Aziz* does not make any determination as to whether a DNS request corresponds to a secure server. (Keromytis Decl. ¶ 165.)

The Request and the Office Action assert that “[i]f the domain name has an associated SX record, then it corresponds to a secure server.” (Cisco Req. Ex. E-4 at 10.) Thus, the Request continues, “checking to see if there is an SX record shows determining whether the intercepted DNS request corresponds to a secure server.” (*Id.*) These statements are incorrect because *Aziz* does not disclose that the existence of an SX record dictates that the matching host name corresponds to a secure server. (Keromytis Decl. ¶ 166.) Just because an SX record may be used for secure communications does not mean that all host names with SX records correspond to secure servers and host names without SX records do not correspond to secure servers. (*Id.*)

In fact, *Aziz* is silent regarding the Request’s and the Office Action’s assertion that existence of an SX record in connection with a host name means that the host name corresponds to a secure server. (*Id.* at ¶ 167.) Instead, *Aziz* explains that the SX record is merely a resource record that stores the name and address of a secure exchanger, such as a firewall. (*Aziz* 6:25-28; Keromytis Decl. ¶ 167.) *Aziz* does not disclose that the SX record indicates whether a corresponding server is secure. (Keromytis Decl. ¶ 167.) Thus, *Aziz*’s checking whether a zone database has an SX record does not disclose or suggest determining whether a DNS request corresponds to a secure server.

Edwards also does not disclose or suggest determining whether the intercepted DNS request corresponds to a secure server. (Keromytis Decl. ¶ 168.) Instead, *Edwards* discloses determining whether an intercepted name request specifies an *available target*. (*Edwards* 933; *see also* Cisco Req. Ex. E-4 at 10-11; Keromytis Decl. ¶ 168.) But an “available target” in *Edwards* does not correspond to a “secure server.” (Keromytis Decl. ¶ 168.) The Request’s assertions to the contrary are incorrect because they are directly contradicted by both *Edwards* and other assertions in the Request itself. For example, the Request asserts that “*Edwards* describes the list of available targets as being *only those services* that have authentication and authorization enabled,” and that “a target with authentication and authorization enabled corresponds to a ‘secure server.’” (Cisco Req. Ex. E-4 at 11, emphasis added.) The Request is incorrect, however, that available targets are only those services that have authentication and authorization enabled because *Edwards* explicitly teaches that an administrator can remove these controls from an available target:

When a target is made available, authentication and authorization are enabled for that target; this is indicated by the ‘AA’ before the name in the “Available Targets” list. *Once the target is available, the administrator can adjust the access control as required.*

(*Edwards* 933, emphasis added; Keromytis Decl. ¶ 168.) In fact, the Request points to this *identical* passage to show an alleged example of a *nonsecure* computer in *Edwards* for a different element of claim 1. (Cisco Req. Ex. E-4 at 12-13, citing to the above passage in *Edwards* and asserting that by adjusting the access control, an administrator can make a target *not* correspond to a secure server.) Thus, merely determining whether a name request specifies an available target does not disclose determining whether a DNS request corresponds to a secure server.

Accordingly, *Aziz* and *Edwards*, alone or in combination, do not disclose or suggest determining whether a DNS request corresponds to a secure server. As such, the combination of *Aziz* and *Edwards* does not render obvious claim 1.

d. The Combination of *Aziz* and *Edwards* Fails to Disclose or Suggest “When the Intercepted DNS Request Corresponds to a Secure Server, Automatically Initiating an Encrypted Channel Between the Client and the Secure Server”

Independent claim 1 recites “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server.” The Request and the Office Action assert that *Aziz* discloses automatically initiating an encrypted channel by “creat[ing] a tunnel map entry . . . which is used by crypto-processor 230 to encrypt messages to inside host 140.” (Req. Ex. E-4 at 17, quoting *Aziz* 11:16-20.) This is incorrect because *Aziz*’s creating a tunnel map entry does not automatically initiate an encrypted channel. (Keromytis Decl. ¶ 169.)

Aziz only discloses adding an SX record to a tunnel map, and explicitly states that the process ends after “authorized client” 210 updates its tunnel map. (See *Aziz* 11:20-62; Keromytis Decl. ¶ 170.) *Aziz* does not disclose that any encrypted channel is initiated. Tunnel map entry 500 is merely a listing of fields from A records and/or SX records that are stored at “authorized client” 210:

[T]o construct a tunnel map entry 500, resolver 225 uses the data in the A record for inside host 140 as the destination address in field1 510. Resolver 225 fills in field2 520 and field3 530 using the data in the A and KEY records for the secure exchanger identified in the SX record (i.e., firewall 110), respectively. . . . [F]ield4 540 is used to indicate the scope of coverage of the secure exchanger identified in the SX record.

(*Aziz* 11:20-28.) Merely storing resource records in a memory does not initiate an encrypted channel. (Keromytis Decl. ¶ 170.)

The Request and the Office Action assert that creating a tunnel map entry 500 is initiating a VPN because “application 215 can now communicate securely with inside host 140.” (Req. Ex. E-4 at 17, quoting *Aziz* 11:54-60.) However, the sentence cited by the Requester and the Examiner must be read in light of *Aziz* as a whole and, more particularly, in light of the very next sentence in *Aziz*. (Keromytis Decl. ¶ 171.) This sentence makes clear that the process in *Aziz* ends once tunnel map entry 500 is created, and before any alleged encrypted channel is ever initiated:

[A]pplication 215 can now communicate securely with inside host 140 because the tunnel map entry 500 provides all the information that crypto-processor 230 needs to encrypt messages to inside host 140. *This completes the execution in an embodiment where one name server is used and the network topology is not hidden.*

(*Aziz* 11:54-62, emphasis added; Keromytis Decl. ¶ 171.)

Thus, *Aziz* discloses stopping execution of the process after storing the SX record in the tunnel map entry 500. (Keromytis Decl. ¶ 172.) *Aziz* does not disclose *automatically* initiating an

encrypted channel after storing the SX record. (*Id.*) Merely having the *capability* to send an encrypted message and automatically initiating an encrypted channel are not the same. (*Id.*) *Aziz* only discloses a *capability* for sending an encrypted message, but does not disclose *automatically initiating an encrypted channel*. (*Id.*)

Moreover, *Aziz* does not disclose or suggest *automatically* initiating an encrypted channel when the intercepted DNS request corresponds to a secure server, because, even when it is allegedly determined that the request corresponds to a secure server (i.e., when it is determined that an SX record exists, according to the Request), *Aziz* discloses situations when an encrypted channel may not be automatically initiated. For example, if an SX record exists, but A and NS records do not exist (*Aziz* Fig. 4B, step 440, NO), then *Aziz* discloses that the process ends without even generating a tunnel map entry (*id.* at 12:15-22; Keromytis Decl. ¶ 173). Additionally, even after a tunnel map entry is created in *Aziz*, firewall 110 may prevent communication to inside host 140, thus preventing any alleged encrypted channel from being initiated. (Keromytis Decl. ¶ 173.) Thus, *Aziz* does not disclose or suggest automatically initiating an encrypted channel when the intercepted DNS request corresponds to a secure server.

Edwards does not make up for the deficiencies of *Aziz*. Nor do the Request and the Office Action assert that it does. (*See* Cisco Req. Ex. E-4 at 17, relying only on *Aziz* as disclosing this feature.) Accordingly, *Aziz* and *Edwards*, alone or in combination, do not disclose or suggest when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server. As such, the combination of *Aziz* and *Edwards* does not render obvious claim 1.

For at least the reasons provided above, *Aziz* and *Edwards* do not render claim 1 obvious. Accordingly, Patent Owner requests that the rejection of claim 1 be withdrawn and the claim be confirmed.

3. Independent Claims 7 and 13

Independent claim 7 recites features similar to those described above for claim 1. For example, claim 7's recited feature of "a computer readable medium storing a domain name server (DNS) proxy module . . . intercepting a DNS request sent by a client" is similar to the "domain name server (DNS) proxy module that intercepts DNS requests sent by a client" feature of claim 1. Also, claim 7's recited feature of "determining whether the intercepted DNS request corresponds to a secure server" is similar to the "determining" step of claim 1, discussed above. Additionally, claim 7's recited feature of "when the intercepted DNS request does not correspond to a secure

server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer,” is similar to the “forwarding” feature of claim 1. Furthermore, claim 7’s recited feature of “when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,” is similar to the “automatically initiating” feature of claim 1, also discussed above. Accordingly, *Aziz* and *Edwards* do not disclose or suggest these features of claim 7 for similar reasons as those discussed above with respect to claim 1.

Independent claim 13 also recites features similar to those described above for claim 1. For example, claim 13’s recited “computer readable medium storing a domain name server (DNS) module” feature is similar to the “data processing device . . . storing a domain name server (DNS) proxy module” feature of claim 1. Also, claim 13’s recited “determining whether a DNS request sent by a client corresponds to a secure server” is similar to the “determining” step of claim 1, discussed above. Additionally, claim 13’s recited feature of “when the DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer,” is similar to the “forwarding” feature of claim 1. Independent claim 13 also recites features that differ from the features recited in claim 1. For example, claim 13 recites “when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel” But given the arguments presented in the Cisco Request and the Office Action (Cisco Req. Ex. E-4 at 31-32), however, Patent Owner asserts that *Aziz* and *Edwards* do not disclose these features of claim 13 for similar reasons as those discussed above with respect to claim 1.

Thus, for these reasons, Patent Owner requests that the rejection of claims 7 and 13 under 35 U.S.C. § 103(a) be withdrawn, and their patentability confirmed.

4. Dependent Claims 2, 8, and 14

Dependent claims 2, 8, and 14 depend from independent claims 1, 7, and 13, respectively, and include all of their features. Thus, claims 2, 8, and 14 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 2, 8, and 14 also distinguish over *Aziz* and *Edwards* for additional reasons. For example, dependent claims 2, 8, and 14 recite “sending a request to the secure server to establish an encrypted channel between the secure server and the client.” *Aziz* and *Edwards*, alone or in combination, do not disclose or suggest this feature.

Aziz does not disclose sending a request to the secure server to establish an encrypted channel. The Cisco Request takes the position that *Aziz*’s inside host 140 is the recited “secure server.” (See, e.g., *id.* at 16-18, asserting that *Aziz* discloses initiating an encrypted channel between

the client and the secure server by disclosing creating a tunnel map entry to allow authorized client 210 to “communicate securely *with inside host 140*,” emphasis added.) The Request then asserts that *Aziz* discloses sending a request *to the secure server* to establish an encrypted channel by disclosing “making an additional query for [an] address [of a second name server].” (*Id.* at 20, citing *Aziz* claims 28, 30.) This is incorrect for two reasons. First, *Aziz* does not disclose that the query for an address is a request to establish an encrypted channel. (Keromytis Decl. ¶ 174.) Second, *Aziz* does not disclose that the request is sent *to the alleged secure server*, inside host 140. (*Id.*) In fact, because *Aziz* describes the query as being for an address, this suggests that the query would be sent to a name server, and not to inside host 140. (*Id.*) The Request also asserts that this query is sent “to the secure exchanger.” (Cisco Req. Ex. E-4 at 20.) Even if this assertion were supported by *Aziz* (which it is not), the “secure exchanger” in *Aziz* is synonymous with firewall 110, which is also not inside host 140. (*Aziz* 6:33-40, Fig. 1.) Thus, *Aziz* does not disclose or suggest sending a request to the secure server to establish an encrypted channel.

Edwards does not make up for the deficiencies of *Aziz*, because *Edwards* also does not disclose or suggest this feature. Nor do the Request and the Office Action assert that it does. (*See, e.g.*, Cisco Req. Ex. E-4 at 19-20, relying only on *Aziz* as allegedly disclosing this feature.)

For at least the reasons provided above, *Aziz* and *Edwards* do not render claims 2, 8, and 14 obvious. Accordingly, Patent Owner requests that the rejection of claims 2, 8, and 14 be withdrawn and the claims be confirmed.

5. Dependent Claims 3, 9, and 15

Dependent claims 3, 9, and 15 each depend from one or more of claims 1, 2, 7, 8, 13, and 14, and include all of their features. Thus, claims 3, 9, and 15 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 3, 9, and 15 also distinguish over *Aziz* and *Edwards* for additional reasons. For example, dependent claims 3, 9, and 15 recite “when the client is not authorized to access the secure server, returning a host unknown error message to the client.” *Aziz* and *Edwards*, alone or in combination, do not disclose or suggest this feature. In particular, *Aziz* and *Edwards* do not disclose or suggest returning an error of any kind, let alone a host unknown error, *when the client is not authorized to access the secure server*.

The Cisco Request points to a portion of *Aziz* disclosing that “an error has occurred” if a client is not able to find an authoritative name server. (Cisco Req. Ex. E-4 at 21, citing *Aziz* 12:17-22.) Specifically, *Aziz* discloses that an error occurs when an NS record for the name server does not exist, not *when a client is not authorized to access a secure server*. (*Aziz* 12:17-22;

Keromytis Decl. ¶ 176.) In fact, the cited portion and surrounding disclosure of *Aziz* discusses requests made by application 215 and resolver 225, which are a part of already authorized client 210. (Keromytis Decl. ¶ 176; *Aziz* 12:3-22; *see also id.* at Figs. 2A-2C, showing application 215 and resolver 225 as part of *authorized* client 210.) Thus, *Aziz* does not disclose returning an error message of any kind, let alone a host unknown error message, *when the client is not authorized to access the secure server*, as recited in claims 3, 9, and 15.

Edwards does not make up for the deficiencies of *Aziz*. *Edwards* discloses that an “object not found” error will occur when a requested name “is not in the list of available targets.” (*Edwards* 933.) But this does not disclose returning an error when a client is not authorized to access a secure server. (Keromytis Decl. ¶ 177.) “Available targets” in *Edwards* are “services for which the object gateway has created service interceptors,” but they do not correspond to a “secure server,” as *Edwards* explicitly teaches that an administrator can remove authentication and authorization controls from an available target:

When a target is made available, authentication and authorization are enabled for that target; this is indicated by the ‘AA’ before the name in the “Available Targets” list. *Once the target is available, the administrator can adjust the access control as required.*

(*Edwards* 933, emphasis added; Keromytis Decl. ¶ 177.) In fact, the Request points to this *identical* passage to show an alleged example of a *nonsecure* computer in *Edwards* for an element of claim 1. (Cisco Req. Ex. E-4 at 12-13, citing to the above passage in *Edwards* and asserting that by adjusting the access control, an administrator can make a target *not* correspond to a secure server.) Thus, the mere existence or nonexistence of an object in a list of available targets has nothing to do with whether or not a particular requesting client is authorized to access that target. (Keromytis Decl. ¶ 177.) Accordingly, *Edwards* does not disclose returning an error message of any kind, let alone a host unknown error message, *when the client is not authorized to access the secure server*, as recited in claims 3, 9, and 15.

Despite the fact that *Aziz* and *Edwards* do not disclose a host unknown error message, the Request includes several paragraphs asserting why it would have been obvious to “translate the idea of *Edwards*’ ‘object not found’ error into a ‘host unknown error’ in the system of *Aziz*.” (Cisco Req. Ex. E-4 at 21-22.) However, even if *Aziz* and *Edwards* are combined in this manner, the combination still does not disclose returning a host unknown error *when the client is not authorized to access the secure server*, as recited in claims 3, 9, and 15.

For at least the reasons provided above, *Aziz* and *Edwards* do not render claims 3, 9, and 15 obvious. Accordingly, Patent Owner requests that the rejection of claims 3, 9, and 15 be withdrawn and the claims be confirmed.

6. Dependent Claims 6 and 12

Dependent claims 6 and 12 depend from independent claims 1 and 7, respectively, and include all of their features. Thus, claims 6 and 12 should be confirmed for at least the reasons discussed above with respect to those claims. Claims 6 and 12 also distinguish over *Aziz* and *Edwards* for additional reasons. For example, dependent claims 6 and 12 recite that “automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.” *Aziz* and *Edwards*, alone or in combination, do not disclose or suggest this feature.

Aziz does not disclose or suggest that automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client. The Cisco Request points to an embodiment of *Aziz* that teaches that the “network topology is hidden” and that the domain database of outside NS 120 “would not include an A record for inside host 140” as allegedly disclosing or suggesting this feature. (*Id.* at 25, citing *Aziz* 11:64-12:1.) This is incorrect, because the A record of inside host 140 is still sent in this embodiment in order to establish the alleged secure communication link. (Keromytis Decl. ¶ 179.) Specifically, *Aziz* discloses that in this embodiment, the A record of inside host 140 “would [instead] be in the zone database used by inside NS 130.” (*Aziz* 12:1-2.) And *Aziz* discloses that inside NS 130 returns the A record of inside host 140 to resolver 225 in this embodiment. (*See, e.g., id.* at 12:47-56; *see also id.* at Fig. 6C, showing the A record of “<inside host 140>” included in the response from inside NS 130.) Thus, *Aziz* does not disclose or suggest that automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

Edwards does not make up for the deficiencies of *Aziz*. The Request cites a portion of *Edwards* that allegedly discloses that services are available to clients only through the name service interceptor and proxy, and asserts that this “suggest[s] that their ‘true’ identification is not provided to the client.” (Cisco Req. Ex. E-4 at 24.) This does not disclose or suggest the recited feature for at least two reasons. First, *Edwards* does not disclose automatically initiating an encrypted channel between a client and a secure server. Nor do the Request and the Office Action assert that it does. (*See, e.g., id.* at 16-18, relying only on *Aziz* for this feature.) Thus, *Edwards* cannot disclose that

automatically initiating the encrypted channel avoids sending a true IP address of a secure server to a client, if *Edwards* does not disclose or suggest automatically initiating the encrypted channel in the first place. Second, *Edwards* does not disclose that the object references to the services are IP addresses. (Keromytis Decl. ¶180.)

In view of the above, *Aziz* and *Edwards* do not disclose or suggest the features of claims 6 and 12. However, the Request asserts that in view of the teachings of *Aziz* and *Edwards*, “it would be obvious to ‘avoid[] sending a true IP address of the secure server to the client.’” (Cisco Req. Ex. E-4 at 25.) This analysis is improper for at least three reasons. First, the Request only addresses part of the claimed feature and does not assert that “*automatically initiating the encrypted channel between the client and the secure server* avoids sending a true IP address of the secure server to the client” (emphasis added). Thus, merely stating that not sending a true IP address would be obvious does not address the claimed feature as a whole. M.P.E.P. § 2141.02 (“The claimed invention as a whole must be considered.”). Second, considering what “would *be* obvious” today improperly fails to determine what would *have been* obvious “at the time of the invention.” M.P.E.P. §§ 2141.01, 2141.02. Third, citing to references that do not disclose or suggest a recited feature and merely stating that the combined teachings would render the recited feature obvious contravenes the requirement that “analysis supporting a rejection under 35 U.S.C. [§] 103 should be made explicit.” M.P.E.P. § 2142.

For at least the reasons provided above, *Aziz* and *Edwards* do not render claims 6 and 12 obvious. Accordingly, Patent Owner requests that the rejection of claims 6 and 12 be withdrawn and the claims be confirmed.

7. Dependent Claims 4, 10, and 16

Dependent claims 4, 10, and 16 depend from one or more of allowable claims 1, 3, 7, 9, 13, and 15, and thus include all of the features of the claims from which they depend. The combination of *Aziz* and *Edwards* does not render claims 4, 10, and 16 obvious, and the rejections of claims 4, 10, and 16 should be withdrawn and these claims should be confirmed at least for the reasons discussed above in connection with claims 1, 3, 7, 9, 13, and 15.

J. The Rejection of Claims 5 and 11 Under 35 U.S.C. § 103(a) Based on *Aziz* in View of *Edwards* and *Martin* Should Be Withdrawn (Issue 13)

The Office Action rejects claims 5 and 11 under § 103(a) based on *Aziz* in view of *Edwards* and further in view of *Martin*. (OA at 31.) Claim 5 depends from independent claim 1, and claim 11 depends from independent claim 7. As explained above, *Aziz* and *Edwards*, alone or in combination, do not disclose or suggest the features of claims 1 and 7, and thus do not support the rejection of

those claims. The above-listed rejection of claims 5 and 11 should also be withdrawn and the claims should be confirmed because *Martin* does not remedy the deficiencies of the primary references discussed above with respect to independent claims 1 and 7. Nor do the Request and the Office Action assert that *Martin* does. Accordingly, the rejection of claims 5 and 11 under § 103 based on *Aziz* in view of *Edwards* and further in view of *Martin* should be withdrawn and the claims should be confirmed.

Accordingly, Patent Owner respectfully requests that the rejection of claims 1-4, 6-10, and 12-16 under 35 U.S.C. § 103(a) based on *Aziz* in view of *Edwards* and the rejection of claims 5 and 11 under 35 U.S.C. § 103(a) based on *Aziz* in view of *Edwards* and in further view of *Martin* be withdrawn, and the patentability of these claims be confirmed.

K. Secondary Considerations Demonstrate Nonobviousness

Even if the Office had established a prima facie case of obviousness regarding any of claims 1-16 (which it has not), there is substantial evidence to rebut any finding of obviousness. As provided in M.P.E.P. § 2145, “[o]ffice personnel should consider all rebuttal arguments and evidence presented by applicants,” including evidence relating to the secondary considerations as set forth in *Graham v. John Deere Co.*, 383 U.S. 1 (1966), which can support the nonobviousness of the claimed inventions. Those secondary considerations include commercial success, acceptance by others in the field, long-felt need, failure of others, and praise by others. M.P.E.P. § 2145. Here, evidence related to secondary considerations rebuts any finding of obviousness of the claimed inventions.

Generally, the computer and Internet-security industries have long sought ways to conveniently establish VPNs. Around the time of the effective filing date of the ’151 patent, it was widely recognized that providing secure remote access to a LAN or WAN was extremely difficult for IT support desks. (Short Decl. ¶¶ 3, 8, 11.) Specifically, remote access was “a nightmare” for support desks. (*Id.* at ¶ 8.) Staffers never knew what combination of CPU, modem, operating system, and software configuration they were going to have to support, and adding the commercially available VPN software only made matters worse. (*Id.*) The computer and Internet-security industries were forced to choose between ease of use and security, but they could not have both. (*Id.* at ¶ 9.) The inventions claimed in the ’151 patent, which provide systems and methods of automatically initiating an encrypted channel between a client and a secure server, combine both ease of use and security aspects without sacrificing one or the other. (*Id.*)

Prior to the features claimed in the ’151 patent, there was a long-felt need for a system that could establish secure communications, such as an encrypted channel, in a simple and

straightforward manner because “a solution that was difficult for an end-user to employ would likely have lead [sic] to a lack of use or incorrect use.” (*Id.* at ¶ 3.) As one example of the manifestation of the long-felt need, the Defense Advanced Research Projects Agency (“DARPA”) funded various research programs to further the science and technology of information assurance and survivability. (*Id.* at ¶¶ 4-5.) One such program, “Next Generation Internet,” received approximately \$130 million in funding between 1998 and 2000. (*Id.* at ¶ 4.)

Recognizing this long-felt need for these inventions, both In-Q-Tel, a venture capital firm that invests in companies developing cutting-edge technology, and SAIC (the original owner of the ’151 patent) also spent significant resources on their development. (*Id.* at ¶¶ 6-7.) In fact, in the year the inventions claimed in the ’151 patent were developed, SAIC spent approximately 85% of its entire research and development budget for that year on developing these and other similar inventions. (*Id.* at ¶ 7.)

Other attempts to provide an easy-to-use solution were unsuccessful. For example, the DARPA-funded research programs discussed above fell far short of the claimed inventions of the ’151 patent. (*Id.* at ¶¶ 4-5, 10.) One such program, “Dynamic Coalitions,” was specifically created to address the ability of the Department of Defense to quickly and easily set up secure communications over the Internet. (*Id.* at ¶¶ 4-5.) More than fifteen prestigious organizations took part in the “Dynamic Coalitions” research program, but none of them came up with a solution, in the relevant time frame, that was even close to the solutions provided in the claimed inventions of the ’151 patent. (*Id.*) That is, they did not develop a solution that automatically initiated an encrypted channel between a client and a secure server when an intercepted DNS request sent by the client corresponds to a secure server. (*Id.*) By providing systems and methods of automatically initiating an encrypted channel between a client and a secure server, the inventions of the ’151 patent succeeded where others failed. (*Id.* at ¶ 11.)

The claimed inventions have also experienced commercial success. In particular, SafeNet, a leading provider of Internet-security technology that is the de facto standard in the VPN industry, entered into a portfolio license in July 2002 with the original owner of the ’151 patent. (*Id.* at ¶ 12.) SafeNet licensed the patents because of features disclosed and claimed in the patents, including those in the ’151 patent. (*Id.*) In addition, Microsoft has entered into a similar license that includes the ’151 patent. (*Id.*) Indeed, as noted, Microsoft was found to willfully infringe the ’151 patent and another patent in the Munger patent family, leading to a damages award of over one hundred million dollars. (*Id.*) And on May 3, 2012, Aastra USA, Inc. entered into a license with VirnetX that

includes the '151 patent. (*Id.*) Likewise, on July 11, 2012, Mitel Networks Corporation entered into a license with VirnetX that also includes the '151 patent. (*Id.*)

The claimed inventions of the '151 patent were also contrary to the accepted wisdom at the time of the inventions. (*Id.* at ¶ 13.) For example, there was a general understanding that reliable security could only be achieved through difficult-to-provision VPNs and that easy-to-set-up connections could not be secure. (*Id.*)

The technology of the '151 patent was also met with skepticism by those skilled in the art who learned of the patented inventions. (*Id.* at ¶ 15.) For example, a DARPA program manager informed one of the coinventors of the '151 patent that the technology disclosed in the '151 patent would never be adopted. (*Id.*) Moreover, the IT offices of many large companies and institutions expressed skepticism that secure connections could ever be enabled easily by regular computer users. (*Id.*)

Several events also demonstrate praise for the inventions in the '151 patent by those in the field. As discussed above, SAIC invested a disproportionately large percentage of its internal resources in the technology. (*Id.* at ¶ 17.) SafeNet, Microsoft, Aastra, and Mitel have all licensed the technology. (*Id.*) A study done by CSMG praised the inventions. (*Id.*) Jim Rutt at Network Solutions, which was eventually acquired by Verisign, praised and expressed significant interest in the technology and would have invested but for a change in circumstances at his company. (*Id.*) This evidence showing that the claimed inventions met a long-felt need, succeeded where others have failed, have been commercially successful, were contrary to the accepted wisdom at the time of the invention, were met by skepticism by those skilled in the art, and were praised by others in the field, rebuts any finding that the claimed inventions would have been obvious.

IV. Conclusion

For at least these reasons, VirnetX requests reconsideration and withdrawal of the rejections in the Office Action and confirmation of the patentability of all of the claims of the '151 patent.

VirnetX notes that the Requests, Orders, and Office Action contain a number of assertions and allegations concerning the disclosure, claims, and cited references. VirnetX does not subscribe to any assertion or allegation in the Requests, Orders, and Office Action regardless of whether it is addressed specifically herein.

Please grant any extension of time and charge any required fees to Deposit Account No. 06-0916.

Respectfully submitted,

FINNEGAN, HENDERSON, FARABOW,
GARRETT & DUNNER, L.L.P.

Dated: July 20, 2012

By: /Joseph E. Palys/
Joseph E. Palys
Reg. No. 46,508



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.
95/001,697	07/25/2011	Edward Colby Munger	41484-80130	2161
23630	7590	04/20/2012	EXAMINER	
McDermott Will & Emery 600 13th Street, NW Washington, DC 20005-3096			YIGDALL, MICHAEL J	
			ART UNIT	PAPER NUMBER
			3992	
			MAIL DATE	DELIVERY MODE
			04/20/2012	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.



DO NOT USE IN PALM PRINTER

(THIRD PARTY REQUESTER'S CORRESPONDENCE ADDRESS)

SIDLEY AUSTIN LLP
717 NORTH HARWOOD
SUITE 3400
DALLAS, TX 75201

MAILED

APR 20 2012

CENTRAL REEXAMINATION UNIT

**Transmittal of Communication to Third Party Requester
Inter Partes Reexamination**

REEXAMINATION CONTROL NUMBER 95/001,697 + 95/001,714

PATENT NUMBER 7,490,151.

TECHNOLOGY CENTER 3999.

ART UNIT 3992.

Enclosed is a copy of the latest communication from the United States Patent and Trademark Office in the above-identified reexamination proceeding. 37 CFR 1.903.

Prior to the filing of a Notice of Appeal, each time the patent owner responds to this communication, the third party requester of the *inter partes* reexamination may once file written comments within a period of 30 days from the date of service of the patent owner's response. This 30-day time period is statutory (35 U.S.C. 314(b)(2)), and, as such, it cannot be extended. See also 37 CFR 1.947.

If an *ex parte* reexamination has been merged with the *inter partes* reexamination, no responsive submission by any *ex parte* third party requester is permitted.

All correspondence relating to this *inter partes* reexamination proceeding should be directed to the **Central Reexamination Unit** at the mail, FAX, or hand-carry addresses given at the end of the communication enclosed with this transmittal.

OFFICE ACTION IN INTER PARTES REEXAMINATION	Control No.	Patent Under Reexamination
	95/001,697 + 95/001,714	MUNGER ET AL.
	Examiner	Art Unit
	Michael J. Yigdall	3992

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address. --

Responsive to the communication(s) filed by:

Patent Owner on _____

Third Party(ies) on _____

RESPONSE TIMES ARE SET TO EXPIRE AS FOLLOWS:

For Patent Owner's Response:

2 MONTH(S) from the mailing date of this action. 37 CFR 1.945. EXTENSIONS OF TIME ARE GOVERNED BY 37 CFR 1.956.

For Third Party Requester's Comments on the Patent Owner Response:

30 DAYS from the date of service of any patent owner's response. 37 CFR 1.947. NO EXTENSIONS OF TIME ARE PERMITTED. 35 U.S.C. 314(b)(2).

All correspondence relating to this inter partes reexamination proceeding should be directed to the **Central Reexamination Unit** at the mail, FAX, or hand-carry addresses given at the end of this Office action.

This action is not an Action Closing Prosecution under 37 CFR 1.949, nor is it a Right of Appeal Notice under 37 CFR 1.953.

PART I. THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. Notice of References Cited by Examiner, PTO-892
2. Information Disclosure Citation, PTO/SB/08
3. _____

PART II. SUMMARY OF ACTION:

- 1a. Claims 1-16 are subject to reexamination.
- 1b. Claims _____ are not subject to reexamination.
2. Claims _____ have been canceled.
3. Claims _____ are confirmed. [Unamended patent claims]
4. Claims _____ are patentable. [Amended or new claims]
5. Claims 1-16 are rejected.
6. Claims _____ are objected to.
7. The drawings filed on _____ are acceptable are not acceptable.
8. The drawing correction request filed on _____ is: approved. disapproved.
9. Acknowledgment is made of the claim for priority under 35 U.S.C. 119 (a)-(d). The certified copy has:
 - been received. not been received. been filed in Application/Control No _____.
10. Other _____

DETAILED ACTION

1. A first request for *inter partes* reexamination of claims 1-16 of U.S. Patent No. 7,490,151 (“the ‘151 patent”) was filed on July 25, 2011 and assigned Control No. 95/001,697 (“the ‘1697 proceeding”). An order granting the request was mailed on October 21, 2011.

A second request for *inter partes* reexamination of claims 1-16 of the ‘151 patent was filed on August 16, 2011 and assigned Control No. 95/001,714 (“the ‘1714 proceeding”). An order granting the request was mailed on October 31, 2011.

A decision merging the ‘1697 and ‘1714 proceedings was mailed on March 15, 2012.

Prior Art Cited in the Merged Proceedings

2. The following patents and printed publications were cited in the ‘1697 and ‘1714 proceedings:

Aventail Connect v3.1/v2.6 Administrator's Guide, 1999 (“Aventail Connect v3.1”).

Aventail Connect v3.01/v2.51 Administrator's Guide, 1999 (“Aventail Connect v3.01”).

Aventail AutoSOCKS v2.1 Administration and User's Guide, 1997 (“Aventail AutoSOCKS”).

Wang, “Core Network Architecture Recommendations for Access to Legacy Data Networks over ADSL,” Broadband Forum Technical Report TR-025, September 1999 (“Wang”).

U.S. Patent No. 6,496,867 to Beser et al. (“Beser”).

Kent et al., “Security Architecture for the Internet Protocol,” Network Working Group RFC 2401, November 1998 (“Kent”).

BinGO! User's Guide: Installation and Configuration and Extended Feature Reference,
March 1999 ("BinGO").

Kiuchi, Takahiro and Shigekoto Kaihara, "C-HTTP – The Development of a Secure,
Closed HTTP-based Network on the Internet," Proceedings of the SNDSS, 1996 ("Kiuchi").

U.S. Patent No. 5,898,830 to Wesinger, Jr. et al. ("Wesinger").

U.S. Patent No. 6,182,141 to Blum et al. ("Blum").

U.S. Patent No. 6,119,234 to Aziz et al. ("Aziz").

Edwards, Nigel and Owen Rees, "High Security Web Servers and Gateways," Computer
Networks and ISDN Systems 29, September 1997, pages 927-938 ("Edwards").

Martin, David M., "A Framework for Local Anonymity in the Internet," Technical
Report, Boston University, 21 February 1998 ("Martin").

Rejections Proposed in the Requests

3. The following rejections of the claims were proposed in the '1697 and '1714 requests for
inter partes reexamination:

Issue 1: Claims 1-16 are rejected as anticipated under 35 U.S.C. § 102(b) based on
Aventail Connect v3.01 (see the '1697 request, pages 21-50 and Ex. C1).

Issue 2: Claims 1-16 are rejected as anticipated under 35 U.S.C. § 102(b) based on
Aventail AutoSOCKS (see the '1697 request, pages 51-81 and Ex. C2).

Issue 3: Claims 1-16 are rejected as anticipated under 35 U.S.C. § 102(a) based on
BinGO (see the '1697 request, pages 82-117 and Ex. C3).

Issue 4: Claims 1-16 are rejected as obvious under 35 U.S.C. § 103(a) based on Beser in
view of Kent (see the '1697 request, pages 118-150 and Ex. C4).

Issue 5: Claims 1-5, 7-11 and 13-16 are rejected as anticipated under 35 U.S.C. § 102(a) based on Wang (see the '1697 request, pages 151-183 and Ex. C5).

Issue 6: Claims 6 and 12 are rejected as obvious under 35 U.S.C. § 103(a) based on Wang in view of Beser (see the '1697 request, pages 183-184 and Ex. C5).

Issue 7: Claims 1-4, 6-10 and 12-16 are rejected as anticipated under 35 U.S.C. § 102(b) based on Kiuchi (see the '1714 request, page 19 and Ex. E-1).

Issue 8: Claims 5 and 11 are rejected as obvious under 35 U.S.C. § 103(a) based on Kiuchi in view of Martin (see the '1714 request, page 19 and Ex. E-1).

Issue 9: Claims 1-4, 6-10 and 12-16 are rejected as anticipated under 35 U.S.C. § 102(e) based on Wesinger (see the '1714 request, page 19 and Ex. E-2).

Issue 10: Claims 5 and 11 are rejected as obvious under 35 U.S.C. § 103(a) based on Wesinger in view of Martin (see the '1714 request, page 19 and Ex. E-2).

Issue 11: Claims 1, 7 and 13 are rejected as anticipated under 35 U.S.C. § 102(e) based on Blum (see the '1714 request, page 20 and Ex. E-3).

Issue 12: Claims 1-4, 6-10 and 12-16 are rejected as obvious under 35 U.S.C. § 103(a) based on Aziz in view of Edwards (see the '1714 request, page 20 and Ex. E-4).

Issue 13: Claims 5 and 11 are rejected as obvious under 35 U.S.C. § 103(a) based on Aziz in view of Edwards and Martin (see the '1714 request, page 20 and Ex. E-4).

Issue 14: Claims 1-4, 6-10 and 12-16 are rejected as obvious under 35 U.S.C. § 103(a) based on Kiuchi in view of Edwards (see the '1714 request, page 20 and Ex. E-5).

Issue 15: Claims 5 and 11 are rejected as obvious under 35 U.S.C. § 103(a) based on Kiuchi in view of Edwards and Martin (see the '1714 request, page 20 and Ex. E-5).

Art Unit: 3992

Issue 16: Claims 1-4, 6-10 and 12-16 are rejected as obvious under 35 U.S.C. § 103(a) based on Wesinger in view of Edwards (see the '1714 request, page 21 and Ex. E-6).

Issue 17: Claims 5 and 11 are rejected as obvious under 35 U.S.C. § 103(a) based on Wesinger in view of Edwards and Martin (see the '1714 request, page 21 and Ex. E-6).

Statutory Basis for Rejections

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. § 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent.

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. 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

Art Unit: 3992

invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Rejections Adopted

6. Issue 1: The rejection of claims 1-16 as anticipated under 35 U.S.C. § 102(b) based on Aventail Connect v3.01 is ADOPTED essentially as proposed in the request (see the '1697 request, pages 21-50 and Ex. C1) and is set forth below.

Claim 1

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

Aventail Connect v3.01 teaches a computer system comprising a proxy module that intercepts network traffic to and from a client application (see, e.g., page 7, "Aventail Connect is the client component of the Aventail ExtraNet Center. ... You can use Aventail Connect as a simple proxy client for managed outbound access, and for secure inbound access. ... When you run Aventail Connect on your system, it automatically routes appropriate network traffic from a WinSock application to an extranet (SOCKS) server, or through successive servers. ... Aventail Connect is designed to run transparently on each workstation, without adding overhead to the user's desktop."). The intercepted network traffic includes DNS requests sent from the client application (see, e.g., page 11, "The application does a DNS lookup to convert the hostname to an IP address. If the application already knows the IP address, this entire step is skipped. Otherwise, Aventail Connect does the following:").

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

Aventail Connect v3.01 teaches determining whether to redirect and/or encrypt a connection (see, e.g., page 10, "When the Aventail Connect LSP receives a connection request, it determines whether or not the connection needs to be redirected (to an Aventail ExtraNet Server) and/or encrypted (in SSL)."). The determination is based on rules in a configuration file (see, e.g., page 9, "Aventail Connect can change data (compressing it or

Art Unit: 3992

encrypting it, for example) before routing it to the TCP/IP stack for transport over the network. The routing is determined by the rules described in the configuration file.”). Aventail Connect v3.01 further teaches determining whether the DNS request corresponds to a rule for a secure server (see, e.g., pages 11-12, “If the destination hostname matches a redirection rule domain name (i.e., the host is part of a domain we are proxying traffic to) then Aventail Connect creates a false DNS entry (HOSTENT) that it can recognize during the connection request. Aventail Connect will forward the hostname to the extranet (SOCKS) server in step 2 and the SOCKS server performs the hostname resolution.”).

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

Aventail Connect v3.01 teaches forwarding the intercepted DNS request to a standard DNS function when the query does not correspond to a rule for a secure server (see, e.g., page 11, “If the hostname matches a local domain string or does not match a redirection rule, Aventail Connect passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack performs the lookup as if Aventail Connect were not running.”).

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

Aventail Connect v3.01 teaches automatically establishing an encrypted tunnel (see, e.g., page 7, “Aventail Connect can establish an encrypted tunnel automatically.”), and further teaches establishing an encrypted connection when the intercepted DNS request corresponds to a rule for a secure server (see, e.g., page 12, “If the request contains a false DNS entry (from step 1), it will be proxied. ... When the SOCKS negotiation is completed, Aventail Connect notifies the application. From the application’s point of view, the entire SOCKS negotiation, including the authentication negotiation, is merely the TCP handshaking. ... If an encryption module is enabled and selected by the SOCKS server, Aventail Connect encrypts the data on its way to the server on behalf of the application. If data is being returned, Aventail Connect decrypts it so that the application sees cleartext data.”).

Art Unit: 3992

Claim 2

The data processing device of claim 1, wherein step (iii) comprises the steps of:

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

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

Aventail Connect v3.01 teaches determining whether the client application is authorized to access the secure server (see, e.g., page 12, "When the connection is completed, Aventail Connect begins the SOCKS negotiation. It sends the list of authentication methods enabled in the configuration file. Once the server selects an authentication method, Aventail Connect executes the specified authentication processing," and see, e.g., page 73, "User authentication and encryption on the Aventail ExtraNet Server require all users to use Aventail Connect to authenticate and encrypt their sessions before any connection to the internal private network(s). For this example, the Aventail ExtraNet Server encrypts all sessions with SSL.").

Aventail Connect v3.01 teaches establishing the encrypted connection if the client application is authorized to access the secure server (see, e.g., pages 72-73, "The mobile user workstations connected to the public Internet are the client workstations, onto which, Aventail Connect will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail ExtraNet Server unless they are running Aventail Connect. Depending on the security policy and the Aventail ExtraNet Server configuration, Aventail Connect will automatically proxy their allowed application traffic into the private network. In this situation, Aventail Connect will forward traffic destined for the private internal network to the Aventail ExtraNet Server. Then, based on the security policy, the Aventail ExtraNet Server will proxy mobile user traffic into the private network but only to those resources allowed.").

Claim 3

The data processing device of claim 2, wherein step (iii) further comprises the step of:

(c) when the client is not authorized

Aventail Connect v3.01 implicitly teaches returning a host unknown error message when the client application is not authorized to access the secure server. Specifically, Aventail Connect v3.01 describes the use of the SOCKS version 5 and DNS protocols (see, e.g.,

Art Unit: 3992

to access the secure server, returning a host unknown error message to the client.

page 7, "Aventail Connect automates the 'socksification' of Transmission Control Protocol/Internet Protocol (TCP/IP) client applications, making it simple for workstations to take advantage of the SOCKS v5 protocol," and see, e.g., page 11, "The application does a DNS lookup to convert the hostname to an IP address. If the application already knows the IP address, this entire step is skipped. Otherwise, Aventail Connect does the following:"). Returning a host unknown error message when the client application is not authorized to access the secure server is inherent to these protocols (see the '1697 request, pages 28-29 and Ex. C1 at pages 10-13).

Claim 4

The data processing device of claim 3, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

Aventail Connect v3.01 teaches that the client application sending the DNS request comprises a Web browser (see, e.g., page 65, "When users need to access Web pages behind an Aventail ExtraNet Server, you must properly configure the Web browser. ... There are two approaches to configuring Aventail Connect for use with a Web browser," and see, e.g., page 8, "Windows TCP/IP networking applications (such as telnet, e-mail, Web browsers, and ftp) use WinSock (Windows Sockets) to gain access to networks or the Internet. ... The application executes a Domain Name System (DNS) lookup to convert the hostname into an Internet Protocol (IP) address.").

Claim 5

The data processing device of claim 1, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

Aventail Connect v3.01 teaches establishing IP address hopping schemes between the client application and the secure server in the form of Aventail MultiProxy and proxy chaining (see, e.g., page 59, "The Aventail MultiProxy feature allows Aventail Connect to traverse multiple firewalls by making connections through successive proxy servers. Aventail Connect makes a connection with each proxy server individually. Each proxy server forms a link in a chain that connects Aventail Connect to the final destination," and see, e.g.,

Art Unit: 3992

page 63; "Proxy chaining is an Aventail ExtraNet Server feature. With proxy chaining, Aventail ExtraNet Servers forward connections for certain destinations to other proxy servers.").

Claim 6

The data processing device of claim 1, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

Aventail Connect v3.01 implicitly teaches avoiding sending a true IP address of the secure server to the client application because the encrypted connection is routed through a proxy (see, e.g., page 72, "Therefore, no direct network connections between the public LAN and the private LAN can be created without being securely proxied through the Aventail ExtraNet Server," and see the '1697 request, pages 31-32 and Ex. C1 at pages 15-16).

Claim 7

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

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

Aventail Connect v3.01 teaches a computer system comprising a proxy module that intercepts network traffic to and from a client application (see, e.g., page 7, "Aventail Connect is the client component of the Aventail ExtraNet Center. ... You can use Aventail Connect as a simple proxy client for managed outbound access, and for secure inbound access. ... When you run Aventail Connect on your system, it automatically routes appropriate network traffic from a WinSock application to an extranet (SOCKS) server, or through successive servers. ... Aventail Connect is designed to run transparently on each workstation, without adding overhead to the user's desktop."). The intercepted network traffic includes DNS requests sent from the client application (see, e.g., page 11, "The application does a DNS lookup to convert the hostname to an IP address. If the application already knows the IP address, this entire step is skipped. Otherwise, Aventail Connect does the following:").

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

Aventail Connect v3.01 teaches determining whether to redirect and/or encrypt a connection (see, e.g., page 10, "When the Aventail Connect LSP receives a connection

Art Unit: 3992

request, it determines whether or not the connection needs to be redirected (to an Aventail ExtraNet Server) and/or encrypted (in SSL).”). The determination is based on rules in a configuration file (see, e.g., page 9, “Aventail Connect can change data (compressing it or encrypting it, for example) before routing it to the TCP/IP stack for transport over the network. The routing is determined by the rules described in the configuration file.”). Aventail Connect v3.01 further teaches determining whether the DNS request corresponds to a rule for a secure server (see, e.g., pages 11-12, “If the destination hostname matches a redirection rule domain name (i.e., the host is part of a domain we are proxying traffic to) then Aventail Connect creates a false DNS entry (HOSTENT) that it can recognize during the connection request. Aventail Connect will forward the hostname to the extranet (SOCKS) server in step 2 and the SOCKS server performs the hostname resolution.”).

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

Aventail Connect v3.01 teaches forwarding the intercepted DNS request to a standard DNS function when the query does not correspond to a rule for a secure server (see, e.g., page 11, “If the hostname matches a local domain string or does not match a redirection rule, Aventail Connect passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack performs the lookup as if Aventail Connect were not running.”).

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

Aventail Connect v3.01 teaches automatically establishing an encrypted tunnel (see, e.g., page 7, “Aventail Connect can establish an encrypted tunnel automatically.”), and further teaches establishing an encrypted connection when the intercepted DNS request corresponds to a rule for a secure server (see, e.g., page 12, “If the request contains a false DNS entry (from step 1), it will be proxied. ... When the SOCKS negotiation is completed, Aventail Connect notifies the application. From the application’s point of view, the entire SOCKS negotiation, including the authentication negotiation, is merely the TCP handshaking. ... If an encryption module is enabled and selected by the SOCKS server, Aventail Connect encrypts the data on its way to the

Art Unit: 3992

server on behalf of the application. If data is being returned, Aventail Connect decrypts it so that the application sees cleartext data.”).

Claim 8

The computer readable medium of claim 7, wherein step (iv) comprises the steps of

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

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

Aventail Connect v3.01 teaches determining whether the client application is authorized to access the secure server (see, e.g., page 12, “When the connection is completed, Aventail Connect begins the SOCKS negotiation. It sends the list of authentication methods enabled in the configuration file. Once the server selects an authentication method, Aventail Connect executes the specified authentication processing,” and see, e.g., page 73, “User authentication and encryption on the Aventail ExtraNet Server require all users to use Aventail Connect to authenticate and encrypt their sessions before any connection to the internal private network(s). For this example, the Aventail ExtraNet Server encrypts all sessions with SSL.”).

Aventail Connect v3.01 teaches establishing the encrypted connection if the client application is authorized to access the secure server (see, e.g., pages 72-73, “The mobile user workstations connected to the public Internet are the client workstations, onto which, Aventail Connect will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail ExtraNet Server unless they are running Aventail Connect. Depending on the security policy and the Aventail ExtraNet Server configuration, Aventail Connect will automatically proxy their allowed application traffic into the private network. In this situation, Aventail Connect will forward traffic destined for the private internal network to the Aventail ExtraNet Server. Then, based on the security policy, the Aventail ExtraNet Server will proxy mobile user traffic into the private network but only to those resources allowed.”).

Art Unit: 3992

Claim 9

The computer readable medium of claim 8, wherein step (iv) further comprises the step of:

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

Aventail Connect v3.01 implicitly teaches returning a host unknown error message when the client application is not authorized to access the secure server.

Specifically, Aventail Connect v3.01 describes the use of the SOCKS version 5 and DNS protocols (see, e.g., page 7, "Aventail Connect automates the 'socksification' of Transmission Control Protocol/Internet Protocol (TCP/IP) client applications, making it simple for workstations to take advantage of the SOCKS v5 protocol," and see, e.g., page 11, "The application does a DNS lookup to convert the hostname to an IP address. If the application already knows the IP address, this entire step is skipped. Otherwise, Aventail Connect does the following:"). Returning a host unknown error message when the client application is not authorized to access the secure server is inherent to these protocols (see the '1697 request, pages 38-40 and Ex. C1 at pages 26-29).

Claim 10

The computer readable medium of claim 9, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

Aventail Connect v3.01 teaches that the client application sending the DNS request comprises a Web browser (see, e.g., page 65, "When users need to access Web pages behind an Aventail ExtraNet Server, you must properly configure the Web browser. ... There are two approaches to configuring Aventail Connect for use with a Web browser," and see, e.g., page 8, "Windows TCP/IP networking applications (such as telnet, e-mail, Web browsers, and ftp) use WinSock (Windows Sockets) to gain access to networks or the Internet. ... The application executes a Domain Name System (DNS) lookup to convert the hostname into an Internet Protocol (IP) address.").

Claim 11

The computer readable medium of claim 7, wherein automatically initiating the encrypted channel

Aventail Connect v3.01 teaches establishing IP address hopping schemes between the client application and the secure server in the form of Aventail MultiProxy and

Art Unit: 3992

between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

proxy chaining (see, e.g., page 59, "The Aventail MultiProxy feature allows Aventail Connect to traverse multiple firewalls by making connections through successive proxy servers. Aventail Connect makes a connection with each proxy server individually. Each proxy server forms a link in a chain that connects Aventail Connect to the final destination," and see, e.g., page 63, "Proxy chaining is an Aventail ExtraNet Server feature. With proxy chaining, Aventail ExtraNet Servers forward connections for certain destinations to other proxy servers.").

Claim 12

The computer readable medium of claim 7, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

Aventail Connect v3.01 implicitly teaches avoiding sending a true IP address of the secure server to the client application because the encrypted connection is routed through a proxy (see, e.g., page 72, "Therefore, no direct network connections between the public LAN and the private LAN can be created without being securely proxied through the Aventail ExtraNet Server," and see the '1697 request, pages 41-42 and Ex. C1 at page 32).

Claim 13

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

Aventail Connect v3.01 teaches a computer system comprising a proxy module that intercepts network traffic to and from a client application (see, e.g., page 7, "Aventail Connect is the client component of the Aventail ExtraNet Center. ... You can use Aventail Connect as a simple proxy client for managed outbound access, and for secure inbound access. ... When you run Aventail Connect on your system, it automatically routes appropriate network traffic from a WinSock application to an extranet (SOCKS) server, or through successive servers. ... Aventail Connect is designed to run transparently on each workstation, without adding overhead to the user's desktop."). The intercepted network traffic includes DNS requests sent from the client application (see, e.g., page 11, "The application does a DNS lookup to convert the hostname to an IP

Art Unit: 3992

address. If the application already knows the IP address, this entire step is skipped. Otherwise, Aventail Connect does the following:”).

(i) determining whether a DNS request sent by a client corresponds to a secure server;

Aventail Connect v3.01 teaches determining whether to redirect and/or encrypt a connection (see, e.g., page 10, “When the Aventail Connect LSP receives a connection request, it determines whether or not the connection needs to be redirected (to an Aventail ExtraNet Server) and/or encrypted (in SSL).”). The determination is based on rules in a configuration file (see, e.g., page 9, “Aventail Connect can change data (compressing it or encrypting it, for example) before routing it to the TCP/IP stack for transport over the network. The routing is determined by the rules described in the configuration file.”). Aventail Connect v3.01 further teaches determining whether the DNS request corresponds to a rule for a secure server (see, e.g., pages 11-12, “If the destination hostname matches a redirection rule domain name (i.e., the host is part of a domain we are proxying traffic to) then Aventail Connect creates a false DNS entry (HOSTENT) that it can recognize during the connection request. Aventail Connect will forward the hostname to the extranet (SOCKS) server in step 2 and the SOCKS server performs the hostname resolution.”).

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

Aventail Connect v3.01 teaches forwarding the intercepted DNS request to a standard DNS function when the query does not correspond to a rule for a secure server (see, e.g., page 11, “If the hostname matches a local domain string or does not match a redirection rule, Aventail Connect passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack performs the lookup as if Aventail Connect were not running.”).

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

Aventail Connect v3.01 teaches automatically establishing an encrypted tunnel (see, e.g., page 7, “Aventail Connect can establish an encrypted tunnel automatically.”), and further teaches establishing an encrypted connection when the intercepted DNS request corresponds to a rule for a secure server (see, e.g., page 12, “If the request contains a false DNS entry (from step

Art Unit: 3992

1), it will be proxied. ... When the SOCKS negotiation is completed, Aventail Connect notifies the application. From the application's point of view, the entire SOCKS negotiation, including the authentication negotiation, is merely the TCP handshaking. ... If an encryption module is enabled and selected by the SOCKS server, Aventail Connect encrypts the data on its way to the server on behalf of the application. If data is being returned, Aventail Connect decrypts it so that the application sees cleartext data.”).

Claim 14

The computer readable medium of claim 13, wherein step (iii) comprises the steps of

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

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

Aventail Connect v3.01 teaches determining whether the client application is authorized to access the secure server (see, e.g., page 12, “When the connection is completed, Aventail Connect begins the SOCKS negotiation. It sends the list of authentication methods enabled in the configuration file. Once the server selects an authentication method, Aventail Connect executes the specified authentication processing,” and see, e.g., page 73, “User authentication and encryption on the Aventail ExtraNet Server require all users to use Aventail Connect to authenticate and encrypt their sessions before any connection to the internal private network(s). For this example, the Aventail ExtraNet Server encrypts all sessions with SSL.”).

Aventail Connect v3.01 teaches establishing the encrypted connection if the client application is authorized to access the secure server (see, e.g., pages 72-73, “The mobile user workstations connected to the public Internet are the client workstations, onto which, Aventail Connect will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail ExtraNet Server unless they are running Aventail Connect. Depending on the security policy and the Aventail ExtraNet Server configuration, Aventail Connect will automatically proxy their allowed application traffic into the private network. In this situation, Aventail Connect will forward traffic destined for the private internal network to the Aventail ExtraNet Server. Then, based on the

Art Unit: 3992

security policy, the Aventail ExtraNet Server will proxy mobile user traffic into the private network but only to those resources allowed.”).

Claim 15

The computer readable medium of claim 14, wherein step (iii) further comprises the step of:

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

Aventail Connect v3.01 implicitly teaches returning a host unknown error message when the client application is not authorized to access the secure server.

Specifically, Aventail Connect v3.01 describes the use of the SOCKS version 5 and DNS protocols (see, e.g., page 7, “Aventail Connect automates the ‘socksification’ of Transmission Control Protocol/Internet Protocol (TCP/IP) client applications, making it simple for workstations to take advantage of the SOCKS v5 protocol,” and see, e.g., page 11, “The application does a DNS lookup to convert the hostname to an IP address. If the application already knows the IP address, this entire step is skipped. Otherwise, Aventail Connect does the following: ...”). Returning a host unknown error message when the client application is not authorized to access the secure server is inherent to these protocols (see the ‘1697 request, pages 48-49 and Ex. C1 at pages 41-44).

Claim 16

The computer readable medium of claim 15, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

Aventail Connect v3.01 teaches that the client application sending the DNS request comprises a Web browser (see, e.g., page 65, “When users need to access Web pages behind an Aventail ExtraNet Server, you must properly configure the Web browser. ... There are two approaches to configuring Aventail Connect for use with a Web browser,” and see, e.g., page 8, “Windows TCP/IP networking applications (such as telnet, e-mail, Web browsers, and ftp) use WinSock (Windows Sockets) to gain access to networks or the Internet. ... The application executes a Domain Name System (DNS) lookup to convert the hostname into an Internet Protocol (IP) address.”).

Art Unit: 3992

7. Issue 2: The rejection of claims 1-16 as anticipated under 35 U.S.C. § 102(b) based on Aventail AutoSOCKS is ADOPTED essentially as proposed in the request (see the '1697 request, pages 51-81 and Ex. C2) and is set forth below.

Claim 1

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

Aventail AutoSOCKS teaches a computer system comprising a proxy module that intercepts network traffic to and from a client application (see, e.g., page 1, "AutoSOCKS transparently intercepts WinSock communication requests issued by TCP/IP applications and processes them based upon a set of routing directives (rules) assigned when AutoSOCKS is configured," and see, e.g., page 6, "When you run AutoSOCKS on your system, it automatically routes appropriate network traffic from a WinSock application to the SOCKS server."). The intercepted network traffic includes DNS requests sent from the client application (see, e.g., page 8, "The application does a DNS lookup to convert the hostname to an IP address. However, if the application already knows the IP address, this entire step is skipped. Otherwise, AutoSOCKS does the following:").

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

Aventail AutoSOCKS teaches routing requests based on rules in a configuration file (see, e.g., page 7, "In simple terms, AutoSOCKS redirects WinSock calls (both parameters and data) and reroutes them through a SOCKS-based server when required. The routing is determined by the rules described in the configuration file created when AutoSOCKS is installed."), and further teaches determining whether the DNS request corresponds to a rule for a secure server (see, e.g., page 8, "If the destination hostname matches a redirection rule domain name (i.e. the host is part of a domain we are proxying traffic to) then AutoSOCKS creates a false DNS entry (HOSTENT) that it can recognize during the connection request. AutoSOCKS will forward the hostname to the SOCKS server in step 2 and the SOCKS server performs the hostname resolution.").

Art Unit: 3992

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

Aventail AutoSOCKS teaches forwarding the intercepted DNS request to a standard DNS function when the query does not correspond to a rule for a secure server (see, e.g., page 8, "If the hostname matches a local domain string or does not match a redirection rule, AutoSOCKS passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack then performs the lookup as if AutoSOCKS is not running.").

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

Aventail AutoSOCKS teaches establishing an encrypted connection when the intercepted DNS request corresponds to a rule for a secure server (see, e.g., pages 8-9, "If the request contains a false DNS entry (from step 1) it will be proxied. ... When the SOCKS negotiation is completed, AutoSOCKS notifies the application. From the application's point of view, the entire SOCKS negotiation including the authentication negotiation, is merely the TCP handshaking. ... If an encryption module is enabled and selected by the SOCKS server, AutoSOCKS encrypts the data on its way to the server on behalf of the application. If data is being returned, AutoSOCKS decrypts it so that the application sees clear text data.").

Claim 2

The data processing device of claim 1, wherein step (iii) comprises the steps of:

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

Aventail AutoSOCKS teaches determining whether the client application is authorized to access the secure server (see, e.g., page 8, "When the connection is completed, AutoSOCKS begins the SOCKS negotiation. It sends the list of authentication methods enabled in the configuration file. Once the server chooses an authentication method, AutoSOCKS executes the specified authentication processing," and see, e.g., page 39, "End user authentication and encryption has been enabled on the Aventail VPN Server, which will require all end users to use AutoSOCKS to enable authentication and encryption of their sessions before being allowed to have any connectivity to the internal private network(s). For this example, the Aventail VPN Server is configured to use SSL for encryption of all sessions.").

Art Unit: 3992

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

Aventail AutoSOCKS teaches establishing the encrypted connection if the client application is authorized to access the secure server (see, e.g., pages 38-39, "The mobile user workstations connected to the public Internet are the client workstations, onto which, AutoSOCKS will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail VPN Server unless they are running AutoSOCKS. Depending on the security policy and the Aventail VPN Server configuration, AutoSOCKS will automatically proxy their allowed application traffic into the private network. In this is situation, AutoSOCKS will forward traffic destined for the private internal network to the Aventail VPN Server. Then, based on the security policy, the Aventail VPN Server will proxy mobile end user traffic into the private network but only to those resources allowed.").

Claim 3

The data processing device of claim 2, wherein step (iii) further comprises the step of:

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

Aventail AutoSOCKS implicitly teaches returning a host unknown error message when the client application is not authorized to access the secure server. Specifically, Aventail AutoSOCKS describes the use of the SOCKS version 5 and DNS protocols (see, e.g., page 6, "AutoSOCKS automates the 'socksification' of client applications, making it simple for workstations to take advantage of the SOCKS v5 protocol," and see, e.g., page 8, "The application does a DNS lookup to convert the hostname to an IP address. However, if the application already knows the IP address, this entire step is skipped. Otherwise, AutoSOCKS does the following:"). Returning a host unknown error message when the client application is not authorized to access the secure server is inherent to these protocols (see the '1697 request, pages 58-60 and Ex. C2 at pages 10-14).

Claim 4

The data processing device of claim

Aventail AutoSOCKS teaches that the client application

Art Unit: 3992

3, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

sending the DNS request comprises a Web browser (see, e.g., page 55, "AutoSOCKS automatically routes appropriate network traffic from a WinSock-compatible TCP/IP application such as an e-mail program or a web browser to a SOCKS-based server.").

Claim 5

The data processing device of claim 1, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP address hopping scheme between the client and the secure server.

Aventail AutoSOCKS implicitly teaches establishing an IP address hopping scheme between the client application and the secure server. Aventail AutoSOCKS describes network routing according to the TCP/IP protocol (see, e.g., page 6, "When you run AutoSOCKS on your system, it automatically routes appropriate network traffic from a WinSock application to the SOCKS server. (WinSock is a Windows component that connects a Windows PC to the Internet using Transmission Control Protocol/Internet Protocol—TCP/IP.) The SOCKS server then sends the traffic to the Internet or the external network."). An IP address hopping scheme is inherent to the TCP/IP protocol (see the '1697 request, pages 60-61 and Ex. C2 at page 15).

Claim 6

The data processing device of claim 1, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

Aventail AutoSOCKS implicitly teaches avoiding sending a true IP address of the secure server to the client application because the encrypted connection is routed through a proxy (see, e.g., page 38, "Therefore, no direct network connections between the public LAN and the private LAN can be created without being securely proxied through the VPN server," and see the '1697 request, pages 61-62 and Ex. C2 at pages 15-16).

Claim 7

A computer readable medium storing a domain name server (DNS) proxy module comprised of computer readable instructions that, when executed, cause a data processing

Aventail AutoSOCKS teaches a computer system comprising a proxy module that intercepts network traffic to and from a client application (see, e.g., page 1, "AutoSOCKS transparently intercepts WinSock communication requests issued by TCP/IP applications

Art Unit: 3992

device to perform the steps of:

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

and processes them based upon a set of routing directives (rules) assigned when AutoSOCKS is configured," and see, e.g., page 6, "When you run AutoSOCKS on your system, it automatically routes appropriate network traffic from a WinSock application to the SOCKS server."). The intercepted network traffic includes DNS requests sent from the client application (see, e.g., page 8, "The application does a DNS lookup to convert the hostname to an IP address. However, if the application already knows the IP address, this entire step is skipped. Otherwise, AutoSOCKS does the following:").

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

Aventail AutoSOCKS teaches routing requests based on rules in a configuration file (see, e.g., page 7, "In simple terms, AutoSOCKS redirects WinSock calls (both parameters and data) and reroutes them through a SOCKS-based server when required. The routing is determined by the rules described in the configuration file created when AutoSOCKS is installed."), and further teaches determining whether the DNS request corresponds to a rule for a secure server (see, e.g., page 8, "If the destination hostname matches a redirection rule domain name (i.e. the host is part of a domain we are proxying traffic to) then AutoSOCKS creates a false DNS entry (HOSTENT) that it can recognize during the connection request. AutoSOCKS will forward the hostname to the SOCKS server in step 2 and the SOCKS server performs the hostname resolution.").

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

Aventail AutoSOCKS teaches forwarding the intercepted DNS request to a standard DNS function when the query does not correspond to a rule for a secure server (see, e.g., page 8, "If the hostname matches a local domain string or does not match a redirection rule, AutoSOCKS passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack then performs the lookup as if AutoSOCKS is not running.").

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

Aventail AutoSOCKS teaches establishing an encrypted connection when the intercepted DNS request corresponds to a rule for a secure server (see, e.g., pages 8-9, "If the request contains a false DNS entry (from

Art Unit: 3992

and the secure server.

step 1) it will be proxied. ... When the SOCKS negotiation is completed, AutoSOCKS notifies the application. From the application's point of view, the entire SOCKS negotiation including the authentication negotiation, is merely the TCP handshaking. ... If an encryption module is enabled and selected by the SOCKS server, AutoSOCKS encrypts the data on its way to the server on behalf of the application. If data is being returned, AutoSOCKS decrypts it so that the application sees clear text data.").

Claim 8

The computer readable medium of claim 7, wherein step (iv) comprises the steps of

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

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

Aventail AutoSOCKS teaches determining whether the client application is authorized to access the secure server (see, e.g., page 8, "When the connection is completed, AutoSOCKS begins the SOCKS negotiation. It sends the list of authentication methods enabled in the configuration file. Once the server chooses an authentication method, AutoSOCKS executes the specified authentication processing," and see, e.g., page 39, "End user authentication and encryption has been enabled on the Aventail VPN Server, which will require all end users to use AutoSOCKS to enable authentication and encryption of their sessions before being allowed to have any connectivity to the internal private network(s). For this example, the Aventail VPN Server is configured to use SSL for encryption of all sessions.").

Aventail AutoSOCKS teaches establishing the encrypted connection if the client application is authorized to access the secure server (see, e.g., pages 38-39, "The mobile user workstations connected to the public Internet are the client workstations, onto which, AutoSOCKS will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail VPN Server unless they are running AutoSOCKS. Depending on the security policy and the Aventail VPN Server configuration, AutoSOCKS will automatically proxy their allowed application traffic into the private network. In this situation, AutoSOCKS will forward traffic

Art Unit: 3992

destined for the private internal network to the Aventail VPN Server. Then, based on the security policy, the Aventail VPN Server will proxy mobile end user traffic into the private network but only to those resources allowed.”).

Claim 9

The computer readable medium of claim 8, wherein step (iv) further comprises the step of:

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

Aventail AutoSOCKS implicitly teaches returning a host unknown error message when the client application is not authorized to access the secure server. Specifically, Aventail AutoSOCKS describes the use of the SOCKS version 5 and DNS protocols (see, e.g., page 6, “AutoSOCKS automates the ‘socksification’ of client applications, making it simple for workstations to take advantage of the SOCKS v5 protocol,” and see, e.g., page 8, “The application does a DNS lookup to convert the hostname to an IP address. However, if the application already knows the IP address, this entire step is skipped. Otherwise, AutoSOCKS does the following:”). Returning a host unknown error message when the client application is not authorized to access the secure server is inherent to these protocols (see the ‘1697 request, pages 68-70 and Ex. C2 at pages 25-30).

Claim 10

The computer readable medium of claim 9, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

Aventail AutoSOCKS teaches that the client application sending the DNS request comprises a Web browser (see, e.g., page 55, “AutoSOCKS automatically routes appropriate network traffic from a WinSock-compatible TCP/IP application such as an e-mail program or a web browser to a SOCKS-based server.”).

Claim 11

The computer readable medium of claim 7, wherein automatically initiating the encrypted channel between the client and the secure server comprises establishing an IP

Aventail AutoSOCKS implicitly teaches establishing an IP address hopping scheme between the client application and the secure server. Aventail AutoSOCKS describes network routing according to the TCP/IP protocol (see, e.g., page 6, “When you run AutoSOCKS

Art Unit: 3992

address hopping scheme between the client and the secure server.

on your system, it automatically routes appropriate network traffic from a WinSock application to the SOCKS server. (WinSock is a Windows component that connects a Windows PC to the Internet using Transmission Control Protocol/Internet Protocol—TCP/IP.) The SOCKS server then sends the traffic to the Internet or the external network.”). An IP address hopping scheme is inherent to the TCP/IP protocol (see the ‘1697 request, page 71 and Ex. C2 at page 30).

Claim 12

The computer readable medium of claim 7, wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.

Aventail AutoSOCKS implicitly teaches avoiding sending a true IP address of the secure server to the client application because the encrypted connection is routed through a proxy (see, e.g., page 38, “Therefore, no direct network connections between the public LAN and the private LAN can be created without being securely proxied through the VPN server,” and see the ‘1697 request, page 72 and Ex. C2 at page 31).

Claim 13

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

Aventail AutoSOCKS teaches a computer system comprising a proxy module that intercepts network traffic to and from a client application (see, e.g., page 1, “AutoSOCKS transparently intercepts WinSock communication requests issued by TCP/IP applications and processes them based upon a set of routing directives (rules) assigned when AutoSOCKS is configured,” and see, e.g., page 6, “When you run AutoSOCKS on your system, it automatically routes appropriate network traffic from a WinSock application to the SOCKS server.”). The intercepted network traffic includes DNS requests sent from the client application (see, e.g., page 8, “The application does a DNS lookup to convert the hostname to an IP address. However, if the application already knows the IP address, this entire step is skipped. Otherwise, AutoSOCKS does the following:”).

(i) determining whether a DNS

Aventail AutoSOCKS teaches routing requests based on

Art Unit: 3992

request sent by a client corresponds to a secure server;

rules in a configuration file (see, e.g., page 7, "In simple terms, AutoSOCKS redirects WinSock calls (both parameters and data) and reroutes them through a SOCKS-based server when required. The routing is determined by the rules described in the configuration file created when AutoSOCKS is installed."), and further teaches determining whether the DNS request corresponds to a rule for a secure server (see, e.g., page 8, "If the destination hostname matches a redirection rule domain name (i.e. the host is part of a domain we are proxying traffic to) then AutoSOCKS creates a false DNS entry (HOSTENT) that it can recognize during the connection request. AutoSOCKS will forward the hostname to the SOCKS server in step 2 and the SOCKS server performs the hostname resolution.").

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

Aventail AutoSOCKS teaches forwarding the intercepted DNS request to a standard DNS function when the query does not correspond to a rule for a secure server (see, e.g., page 8, "If the hostname matches a local domain string or does not match a redirection rule, AutoSOCKS passes the name resolution query through to the TCP/IP stack on the local workstation. The TCP/IP stack then performs the lookup as if AutoSOCKS is not running.").

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

Aventail AutoSOCKS teaches establishing an encrypted connection when the intercepted DNS request corresponds to a rule for a secure server (see, e.g., pages 8-9, "If the request contains a false DNS entry (from step 1) it will be proxied. ... When the SOCKS negotiation is completed, AutoSOCKS notifies the application. From the application's point of view, the entire SOCKS negotiation including the authentication negotiation, is merely the TCP handshaking. ... If an encryption module is enabled and selected by the SOCKS server, AutoSOCKS encrypts the data on its way to the server on behalf of the application. If data is being returned, AutoSOCKS decrypts it so that the application sees clear text data.").

Art Unit: 3992

Claim 14

The computer readable medium of claim 13, wherein step (iii) comprises the steps of

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

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

Aventail AutoSOCKS teaches determining whether the client application is authorized to access the secure server (see, e.g., page 8, "When the connection is completed, AutoSOCKS begins the SOCKS negotiation. It sends the list of authentication methods enabled in the configuration file. Once the server chooses an authentication method, AutoSOCKS executes the specified authentication processing," and see, e.g., page 39, "End user authentication and encryption has been enabled on the Aventail VPN Server, which will require all end users to use AutoSOCKS to enable authentication and encryption of their sessions before being allowed to have any connectivity to the internal private network(s). For this example, the Aventail VPN Server is configured to use SSL for encryption of all sessions.").

Aventail AutoSOCKS teaches establishing the encrypted connection if the client application is authorized to access the secure server (see, e.g., pages 38-39, "The mobile user workstations connected to the public Internet are the client workstations, onto which, AutoSOCKS will be deployed. Due to the routing restrictions described above, these clients will have no network access beyond the Aventail VPN Server unless they are running AutoSOCKS. Depending on the security policy and the Aventail VPN Server configuration, AutoSOCKS will automatically proxy their allowed application traffic into the private network. In this is situation, AutoSOCKS will forward traffic destined for the private internal network to the Aventail VPN Server. Then, based on the security policy, the Aventail VPN Server will proxy mobile end user traffic into the private network but only to those resources allowed.").

Claim 15

The computer readable medium of claim 14, wherein step (iii) further comprises the step of:

Aventail AutoSOCKS implicitly teaches returning a host unknown error message when the client application is not authorized to access the secure server. Specifically,

Art Unit: 3992

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

Aventail AutoSOCKS describes the use of the SOCKS version 5 and DNS protocols (see, e.g., page 6, "AutoSOCKS automates the 'socksification' of client applications, making it simple for workstations to take advantage of the SOCKS v5 protocol," and see, e.g., page 8, "The application does a DNS lookup to convert the hostname to an IP address. However, if the application already knows the IP address, this entire step is skipped. Otherwise, AutoSOCKS does the following:"). Returning a host unknown error message when the client application is not authorized to access the secure server is inherent to these protocols (see the '1697 request, pages 78-81 and Ex. C2 at pages 41-45).

Claim 16

The computer readable medium of claim 15, wherein the client comprises a web browser into which a user enters a URL resulting in the DNS request.

Aventail AutoSOCKS teaches that the client application sending the DNS request comprises a Web browser (see, e.g., page 55, "AutoSOCKS automatically routes appropriate network traffic from a WinSock-compatible TCP/IP application such as an e-mail program or a web browser to a SOCKS-based server.").

8. Issue 3: The rejection of claims 1-16 as anticipated under 35 U.S.C. § 102(a) based on BinGO is ADOPTED as proposed in the request (see the '1697 request, pages 82-117 and Ex. C3) and is incorporated by reference.

9. Issue 4: The rejection of claims 1, 2, 4-8, 10-14 and 16 as obvious under 35 U.S.C. § 103(a) based on Beser in view of Kent is ADOPTED essentially as proposed in the request (see the '1697 request, pages 118-150 and Ex. C4) and is incorporated by reference. The proposed rejection of claims 3, 9 and 15 is NOT ADOPTED (see below in this Office action).

With respect to the rejection of claims 1, 2, 4-8, 10-14 and 16 adopted essentially as proposed in the request, the examiner does not adopt the statement that "an edge router or a

Art Unit: 3992

network device behind an edge router that can communicate through an authenticated and encrypted channel is a 'secure web site'" (see the '1697 request, pages 129, 137 and 145, and Ex. C4 at pages 3, 13 and 25). Instead, the examiner submits that an edge router or a network device behind an edge router that communicates through an authenticated and encrypted channel is reasonably construed as a "secure server."

10. Issue 7: The rejection of claims 1-4, 6-10 and 12-16 as anticipated under 35 U.S.C. § 102(b) based on Kiuchi is ADOPTED as proposed in the request (see the '1714 request, page 19 and Ex. E-1) and is incorporated by reference.

11. Issue 8: The rejection of claims 5 and 11 as obvious under 35 U.S.C. § 103(a) based on Kiuchi in view of Martin is ADOPTED as proposed in the request (see the '1714 request, page 19 and Ex. E-1) and is incorporated by reference.

The examiner notes that in Martin, the term "lanon" refers to a locally anonymous network (see Martin, page 8, "We call a locally anonymous network a *lanon* (think LANon). A lanon will usually consist of a possibly proper subset of nodes in a LAN.").

12. Issue 9: The rejection of claims 1-4, 6-10 and 12-16 as anticipated under 35 U.S.C. § 102(e) based on Wesinger is ADOPTED as proposed in the request (see the '1714 request, page 19 and Ex. E-2) and is incorporated by reference.

13. Issue 10: The rejection of claims 5 and 11 as obvious under 35 U.S.C. § 103(a) based on Wesinger in view of Martin is ADOPTED as proposed in the request (see the '1714 request, page 19 and Ex. E-2) and is incorporated by reference.

Art Unit: 3992

As the examiner noted above, the term “lanon” in Martin refers to a locally anonymous network (see Martin, page 8, “We call a locally anonymous network a *lanon* (think LANon). A lanon will usually consist of a possibly proper subset of nodes in a LAN.”).

14. Issue 11: The rejection of claims 1, 7 and 13 as anticipated under 35 U.S.C. § 102(e) based on Blum is ADOPTED as proposed in the request (see the ‘1714 request, page 20 and Ex. E-3) and is incorporated by reference.

15. Issue 12: The rejection of claims 1-4, 6-10 and 12-16 as obvious under 35 U.S.C. § 103(a) based on Aziz in view of Edwards is ADOPTED essentially as proposed in the request (see the ‘1714 request, page 20 and Ex. E-4) and is incorporated by reference.

The examiner does not adopt the proposed reasoning that “Aziz teaches that the Domain Name Server is a proxy server by teaching that it can supports recursive queries,” and further that “it would have been obvious to one of skill in the art that a DNS proxy server module is a server that implements ‘recursive’ queries, since a recursive query could involve making multiple iterative queries to other DNS servers” (see the ‘1714 request, Ex. E-4 at page 5). The request cites Aziz at column 10, lines 36-42:

At step 405, resolver 225 receives the query from application 215. At step 410, resolver 225 could follow the referral chain to the name server for the domain of inside host 140 or could pass the query on to local NS 250 if the local server supports recursive service. In any case, resolver 225 subsequently receives back a response to the query, at step 415.

The examiner submits that Aziz teaches a “domain name server (DNS) proxy module” not merely because it supports recursive service, but because it “could follow the referral chain

Art Unit: 3992

to the name server for the domain of inside host 140 or could pass the query on to local NS 250.”

Aziz further describes (see column 6, line 62 to column 7, line 7):

A resolver is a program that acts as an intermediary between a name server and an application program running on a client. Resolvers receive queries for information from application programs, direct the queries to an appropriate name server, and then return the responses, if any, to the requesting application. The types of queries include host address for a given host name, host name for a given host address, and general lookups for information stored in the name server database. Resolvers generally perform four steps in handling queries: (1) return the answer to the query if it is available locally; otherwise, (2) find the best servers to ask for the answer; (3) send queries to the servers until one responds; and (4) process the response.

Thus, it would have been obvious to those of ordinary skill in the art that the resolver of Aziz represents a DNS proxy module.

16. Issue 13: The rejection of claims 5 and 11 as obvious under 35 U.S.C. § 103(a) based on Aziz in view of Edwards and Martin is ADOPTED as proposed in the request (see the ‘1714 request, page 20 and Ex. E-4) and is incorporated by reference.

As the examiner noted above, the term “lanon” in Martin refers to a locally anonymous network (see Martin, page 8, “We call a locally anonymous network a *lanon* (think LANon). A lanon will usually consist of a possibly proper subset of nodes in a LAN.”).

17. Issue 14: The rejection of claims 1-4, 6-10 and 12-16 as obvious under 35 U.S.C. § 103(a) based on Kiuchi in view of Edwards is ADOPTED as proposed in the request (see the ‘1714 request, page 20 and Ex. E-5) and is incorporated by reference.

Art Unit: 3992

18. Issue 15: The rejection of claims 5 and 11 as obvious under 35 U.S.C. § 103(a) based on Kiuchi in view of Edwards and Martin is ADOPTED as proposed in the request (see the '1714 request, page 20 and Ex. E-5) and is incorporated by reference.

As the examiner noted above, the term "lanon" in Martin refers to a locally anonymous network (see Martin, page 8, "We call a locally anonymous network a *lanon* (think LANon). A lanon will usually consist of a possibly proper subset of nodes in a LAN.").

19. Issue 16: The rejection of claims 1-4, 6-10 and 12-16 as obvious under 35 U.S.C. § 103(a) based on Wesinger in view of Edwards is ADOPTED as proposed in the request (see the '1714 request, page 21 and Ex. E-6) and is incorporated by reference.

20. Issue 17: The rejection of claims 5 and 11 as obvious under 35 U.S.C. § 103(a) based on Wesinger in view of Edwards and Martin is ADOPTED as proposed in the request (see the '1714 request, page 21 and Ex. E-6) and is incorporated by reference.

As the examiner noted above, the term "lanon" in Martin refers to a locally anonymous network (see Martin, page 8, "We call a locally anonymous network a *lanon* (think LANon). A lanon will usually consist of a possibly proper subset of nodes in a LAN.").

Rejections Not Adopted

21. Issue 4: As noted above, the proposed rejection of claims 3, 9 and 15 as obvious under 35 U.S.C. § 103(a) based on Beser in view of Kent (see the '1697 request, pages 118-150 and Ex. C4) is NOT ADOPTED.

The request states that in Beser, "a failure of authentication will result in no establishment of the encrypted channel (IP tunnel) between the first and second network

Art Unit: 3992

devices,” and concludes that Beser “thus teaches a process whereby a domain name will not be resolved if it is determined that the client has not successfully authenticated with the trusted third network device” (see the ‘1697 request, pages 134, 142 and 150, and Ex. C4 at pages 10, 21 and 32). However, the request does not provide factual support for the conclusion that the network device of Beser will necessarily *not* resolve the domain name without authentication, and therefore does not show that the claimed step of “returning a host unknown error message to the client” would have been obvious to those of ordinary skill in the art.

22. Issues 5 and 6: The proposed rejection of claims 1-5, 7-11 and 13-16 as anticipated under 35 U.S.C. § 102(a) based on Wang (see the ‘1697 request, pages 151-183 and Ex. C5) and the proposed rejection of claims 6 and 12 as obvious under 35 U.S.C. § 103(a) based on Wang in view of Beser (see the ‘1697 request, pages 183-184 and Ex. C5) are NOT ADOPTED.

Each of independent claims 1, 7 and 13 recites a step of “when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer.” At page 18, Wang describes:

The BAS extracts the domain string portion of the user-name and sends off a query to NSP to authenticate and obtain address information (e.g., DNS server’s address). In the case of IP network, the NSP replies with an IP address and IP configuration information (e.g., DNS server’s address). This information is passed along to the user during the NCP phase for configuring IP transport (based on IPCP). The BAS maps a user identifier (e.g. port, session identifier, etc.) to the outgoing NSP port.

Based on these teachings of Wang, the request states (see the ‘1697 request, pages 155, 166 and 177, and Ex. C5 at pages 6-7, 25-26 and 44):

In other words, Wang shows the domain name being sent to a DNS server which returns an IP address associated with the domain name. DNS servers are well-known in the art to perform a standard function of returning an IP

Art Unit: 3992

address associated with a domain name. In the example on page 18 [of Wang], the DNS resolution step will be performed and an IP address will be returned regardless of whether the user is requesting access to a domain that is associated with a corporate network or a non-secure destination on the Internet. Thus, only when the DNS request presented by the client computer in this example is specifying a secure domain will the routing to the NSP occur. If the DNS request does not specify a secure domain, normal DNS name resolution will occur.

However, the request does not show that Wang returns an IP address as a result of forwarding an intercepted DNS request to a DNS function. Specifically, Wang describes that the NSP returns “an IP address and IP configuration information (e.g., DNS server’s address)” (see page 18). Here, the “DNS server’s address” is the address of a DNS server, not an address returned from the DNS server as a result of domain name resolution. The other “IP address” is not clearly identified in Wang. The request seems to imply that the IP address is associated with the destination or domain name to which the user is requesting access. However, based on the description in Wang, the IP address seems to be an IP address associated with the user, rather than an IP address associated with a domain name (see page 16; emphasis added):

The BAS by definition is a network layer device and may be required to provide network and higher layer services. The BAS interacts with both the user and the NSPs ‘AAA’ infrastructure to provide functions such as IP address configuration and user authentication, user authorization and NSP accounting using the PPP suite of protocols and proxy transactions to the NSP. In the case of an IP network as shown in Figure 6, an IP address and other configuration information for the user are also obtained from the NSP during this query. ... When the user wants to terminate the connection to the NSP, he terminates the PPP session to the BAS. The BAS deletes the user-NSP mapping in its routing tables and returns the IP address to the NSP.

Thus, the request does not show that Wang teaches the claimed step of “when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer.”

Conclusion

23. In order to ensure full consideration of any amendments, affidavits or declarations, or other documents as evidence of patentability, such documents must be submitted in response to this Office action. Submissions after the next Office action, which is intended to be an Action Closing Prosecution (ACP), will be governed by 37 CFR 1.116(b) and (d), which will be strictly enforced.

24. Extensions of time under 37 CFR 1.136(a) will not be permitted in *inter partes* reexamination proceedings because the provisions of 37 CFR 1.136 apply only to “an applicant” and not to the patent owner in a reexamination proceeding. Additionally, 35 U.S.C. § 314(c) requires that *inter partes* reexamination proceedings “will be conducted with special dispatch” (37 CFR 1.937). Patent owner extensions of time in *inter partes* reexamination proceedings are provided for in 37 CFR 1.956. Extensions of time are not available for third party requester comments, because a comment period of thirty (30) days from service of the patent owner’s response is set by statute. 35 U.S.C. § 314(b)(3).

The patent owner is reminded of the continuing responsibility under 37 CFR 1.985(a) to apprise the Office of any litigation activity, or other prior or concurrent proceeding, involving the ‘151 patent throughout the course of this reexamination proceeding. The third party requesters are also reminded of the ability to similarly apprise the Office of any such activity or proceeding throughout the course of this reexamination proceeding. See MPEP §§ 2686 and 2686.04.

Art Unit: 3992

25. All correspondence relating to this *inter partes* reexamination proceeding should be directed:

By mail to: Mail Stop *Inter Partes* Reexam
Attn: Central Reexamination Unit
Commissioner for Patents
United States Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

By fax to: (571) 273-9900
Central Reexamination Unit

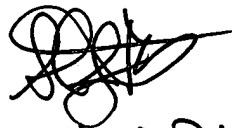
By hand: Customer Service Window
Randolph Building
401 Dulany Street
Alexandria, VA 22314

By EFS: Registered users may submit correspondence via the EFS-Web electronic filing system at <https://efs.uspto.gov/efile/myportal/efs-registered>.

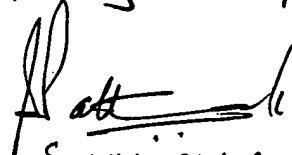
Any inquiry concerning this communication should be directed to the Central Reexamination Unit at telephone number (571) 272-7705.

/Michael J. Yigdall/
Primary Examiner, Art Unit 3992

Conferees:



Stephen J. Roberts
Primary Examiner, Art Unit 3992



SUBHANSHU C. PATIL
SPRS, 3992