

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,
Petitioner,

IPR2025-00836
Patent No. 11,816,249

**PETITION FOR *INTER PARTES* REVIEW
UNDER 35 U.S.C. § 312 AND 37 C.F.R. § 42.104**

TABLE OF CONTENTS

Table of Contents	2
Petitioner’s Exhibit List	5
I. Introduction.....	7
II. Grounds for Standing.....	7
III. Note.....	7
IV. Technology background.....	8
V. Summary of the ’249 patent	8
VI. Prosecution history	9
VII. Effective priority date	10
VIII. Level of ordinary skill in the art	10
IX. Claim construction.....	11
X. Relief requested and reasons therefore	12
XI. Identification of how the claims are unpatentable.....	12
A. Challenged Claims	12
XII. Statutory grounds.....	12
A. Statement of Material Facts.....	13
B. Detailed claim analysis (Grounds 1 & 2).....	15
1. Burns	15
2. Yang	17
3. Reasons to combine Burns and Yang.....	18
4. Guruswamy (Ground 2)	18
5. Reasons to combine Burns, Yang, and Guruswamy (Ground 2).....	19

6.	Claim 1	20
7.	Claim 2	37
8.	Claim 3	40
9.	Claim 4	41
10.	Claim 5	51
11.	Claim 6	52
12.	Claim 7	52
13.	Claim 8	54
14.	Claim 9	55
15.	Claim 10	56
16.	Claim 11	56
17.	Claim 12	59
18.	Claim 13	59
19.	Claim 14	59
20.	Claim 15	60
21.	Claim 16	60
22.	Claim 17	60
23.	Claim 18	60
24.	Claim 19	61
25.	Claims 20-29	65
26.	Claim 30	66
XIII.	Discretionary denial is inappropriate	68
A.	No §325(d) denial	68
B.	No <i>Fintiv</i> denial	69
1.	No evidence regarding a stay	69

2.	Parallel proceeding trial date	69
3.	Investment in the parallel proceeding	70
4.	Overlapping issues with the parallel proceeding	70
5.	Identity of parties	71
6.	Other circumstances	71
C.	No General Plastic denial	72
XIV.	Conclusion	72
	Claims appendix	73
XV.	Mandatory notices	80
A.	Real party-in-interest	80
B.	Related matters	80
C.	Lead and back-up counsel and service information	80
	Certificate of Word Count	82
	Certificate of Service	83

PETITIONER’S EXHIBIT LIST

Ex.1001	U.S. 11,816,249
Ex.1002	Prosecution History of U.S. 11,816,249
Ex.1003	Declaration of Nader Mir, Ph.D. under 37 C.F.R. §1.68
Ex.1004	<i>Curriculum Vitae</i> of Nader Mir, Ph.D.
Ex.1005	U.S. Patent No. 8,341,724 to Burns et al.
Ex.1006	U.S. Patent No. 8,291,495 to Burns et al. (“Yang”)
Ex.1007	U.S. Patent No. 8,209,756 to Guruswamy et al.
Ex.1008	Complaint, <i>QPrivacy USA LLC v. Cisco Systems, Inc.</i> , No. 2:24-cv-00855 (E.D. Tex. Oct. 21, 2024).
Ex.1009	Docket Control Order, <i>QPrivacy USA LLC v. Cisco Systems, Inc.</i> , No. 2:24-cv-00855 (E.D. Tex. Feb. 6, 2025).
Ex.1010	United States District Courts - National Judicial Caseload Profile (December 2024)
Ex.1011	Transmission Control Protocol, RFC 793 (Sept. 1981).
Ex.1012	Andrew Tanenbaum, “Computer Networks,” 3rd ed. (1996).
Ex.1013	U.S. Patent Publication No. 2006/0215695 to Olderdisen
Ex.1014	Perlman, Radia, “Interconnections: Bridges, Routers, Switches, and Internetworking Protocols,” 2d ed. (2000).
Ex.1015	<i>Reserved</i>
Ex.1016	Kozierok, Charles M., “The TCP/IP Guide: A Comprehensive, Illustrated Internet Protocols Reference,” 1st ed. (2005).
Ex.1017	Nayak, Umesha; Hodeghatta Rao, Umesh, “The InfoSec Handbook: An Introduction to Information Security,” 1st ed. (2014).
Ex.1018	U.S. Patent No. 7,406,522 to Riddle
Ex.1019	U.S. Patent Publication No. 2011/0106710 to Reed et al.
Ex.1020	U.S. Patent Publication No. 2011/0022835 to Schibuk et al.

Ex.1021	U.S. Patent No. 6,944,762 to Garrison
Ex.1022	U.S. Patent Publication No. 2005/0281192 to Nadeau et al.
Ex.1023	U.S. Patent Publication No. 2011/0110328 to Pradeep et al.
Ex.1024	U.S. Patent Publication No. 2013/0329578 to Groves et al.
Ex.1025	U.S. Patent Publication No. 2002/0046264 to Dillon et al.
Ex.1026	U.S. Patent Publication No. 2002/0059435 to Border et al.
Ex.1027	U.S. Patent No. 6,826,620 to Mawhinney et al.
Ex.1028	U.S. Patent Publication No. 2017/0230065 to Saraswathyama et al.
Ex.1029	U.S. Patent Publication No. 2014/0351106 to Furr et al.
Ex.1030	U.S. Patent Publication No. 2012/0230208 to Pyatkovskiy
Ex.1031	U.S. Patent Publication No. 2006/0248582 to Panjwani et al.
Ex.1032	U.S. Patent Publication No. 2004/0013112 to Goldberg et al.
Ex.1033	Transmission Control Protocol, RFC 1700 (1994).
Ex.1034	Transmission Control Protocol, RFC 791 (1981).
Ex.1035	Transmission Control Protocol, RFC 790 (1981).
Ex.1036	Proctor, Paul E., “The Practical Intrusion Detection Handbook,” (2001).
Ex.1037	U.S. Patent Publication No. 2008/0159146 to Claudatos et al.
Ex.1038	U.S. Patent Publication No. 2007/0088845 to Memon et al.
Ex.1039	U.S. Patent Publication No. 2005/0047449 to Adolph et al.
Ex.1040	Transmission Control Protocol, RFC 5246 (2008).
Ex.1041	U.S. Patent Publication No. 2003/0038791 to Chou
Ex.1042	U.S. Patent No. 5,289,589 to Bingham et al.
Ex.1043	U.S. Patent Publication No. 2006/0040650 to Schepers et al.
Ex.1044	U.S. Patent Publication No. 2003/0014624 to Maturana et al.
Ex.1045	U.S. Patent Publication No. 2006/0209858 to Blum
Ex.1046	Transmission Control Protocol, RFC 4253 (2006).

I. Introduction

Pursuant to 35 U.S.C. §§ 311, 314(a), and 37 C.F.R. § 42.100, Cisco Systems, Inc. (“Petitioner”) respectfully requests that the Board review and cancel as unpatentable under (pre-AIA) 35 U.S.C. §103(a) claims 1-30 (hereinafter, the “Challenged Claims”) of U.S. 11,816,249 (the “’249 patent,” Ex.1001).

The ’249 patent describes managing encrypted or private data communicated between a remote server and a user’s device. Ex.1001, Abstract, Claim 1. The patent purports to “secur[e] private elements of a user’s device” by having a remote device modify data packets based on configurable preferences. Ex.1001, 1:32-35, Abstract, Claim 1. As this petition demonstrates, such ideas were known before the ’249 patent’s earliest priority date.

Because the Challenged Claims merely recite an obvious combination of known concepts, Petitioner asks the Board to institute trial and find the claims unpatentable.

II. Grounds for Standing

Petitioner certifies that the ’249 patent is eligible for IPR and that Petitioner is not barred or estopped from requesting IPR of the Challenged Claims.

III. Note

Petitioner cites to exhibits’ original page numbers. **Emphasis** in quoted material has been added. Claim terms are presented in *italics*.

IV. Technology background

Dr. Mir's Declaration (Ex.1003) provides a technology background, where he explains networking concepts that would have been background knowledge, such as the use of Transmission Control Protocol/Internet Protocol (TCP/IP), (Ex.1003, ¶¶26-33; Ex.1011; Ex.1012; Ex.1013; Ex.1014; Ex.1016), packet-based networking (Ex.1003, ¶¶34-37; Ex.1012; Ex.1014; Ex.1016), data encryption (Ex.1003, ¶¶38-39; Ex.1016; Ex.1017) and intrusion detection systems (Ex.1003, ¶¶40-44; Ex.1017).

V. Summary of the '249 patent

The '249 patent relates to “dynamic management of private data during communication between a remote server and a user's device.” Ex.1001, Abstract. There, the user's device receives a request for retrieval of at least one data packet from the user's device. Ex.1001, 2:10-12. The content of these data packets is encrypted. Ex.1001, 12:45. The user's device is “configured to provide a response corresponding to the received request.” Ex.1001, 2:13-15. Characteristics of a data packet are used to identify a data packet's “data type.” Ex.1001, 9:13-17. Similarly, dynamic features of a data packet are used to identify a data packet's “data pattern.” Ex.1001, 9:65-10:9.

The user specifies their privacy preferences regarding the handling of data packets via a list of data types or data patterns that are either allowed or forbidden.

Ex.1001, 2:18-19, 32-33. A remote device applies these privacy preferences to data packets. Ex.1001, 11:15-19. If the privacy preferences dictate that a certain “data packet is forbidden for sharing” due to the data type and/or data pattern being incompatible with the privacy preferences, the content of the data packet can be modified. Ex.1001, 10:52-56. Modifying the packet includes blocking the packet. Ex.1001, 2:48-54. Once a packet has been modified, the packet is shared so as to maintain communications between the remote server and the user’s device. Ex.1001, 15:55-58.

The ’249 patent specification focuses on restricting access to a user’s “private data,” which “refer to any data and/or portion of data representing user’s behavior, habits, characteristics, opinions, preferences, tendencies, relations, actions, status parameters (e.g., social, health, economic/wealth, legal, dependencies, being part of group and the like) and the like.” Ex.1001, 7:52-58.

VI. Prosecution history

The ’249 patent was filed on January 10, 2023, under U.S. Patent Application No. 16/603,252. Ex.1001. The ’249 patent claims priority via a chain of continuing applications back to Israeli Patent No. 251,683, filed on April 9, 2017. Ex.1001, code (30). In a first Office Action, claims 1, 12, and 19 were rejected on the ground of non-statutory double patenting as being unpatentable over claims 1, 9, and 17 of U.S. Patent No. 11,106,824 (the ’824 patent) in view of

U.S. Patent No. 6,751,728 (Gunter et al.). Ex.1002, 134. The applicants responded by filing a Terminal Disclaimer to disclaim the term of the '249 patent beyond the term of the '824 patent. Ex.1002, 101.

VII. Effective priority date

The '249 patent has an earliest claimed international priority date of April 9, 2017. Determining whether any of the Challenged Claims are entitled this date is unnecessary in this proceeding, as all the cited and relied-upon references predate April 2017. Accordingly, Petitioner has not undertaken a priority date analysis. Petitioner does not waive any right or opportunity it may have to dispute the priority date of the '249 patent in this or another forum where the issue becomes relevant.

VIII. Level of ordinary skill in the art

The '249 patent “relates to data management,” in the context of packet-based “communication between a remote server and a user’s device.” Ex.1001, 1:19, Abstract. A Person of Ordinary Skill in The Art (“POSITA”) in April 2017 would have had a working knowledge of the data communications art that is pertinent to the '249 patent, including packet-based computer networking. *See* Ex.1001, 6:24-27. The '249 patent refers to a variety of internet networking terms and protocols acronyms without explanation, indicating that a POSITA would be familiar with internet networking topics such as TCP/IP. *See* Ex.1001, 9:3-13, 10:4;

Ex.1003, ¶¶26-33.

A POSITA would have had a bachelor’s degree in computer science, computer engineering, electrical engineering, or an equivalent, and three years of professional experience relating to packet-based network communications. Lack of professional experience can be remedied by additional education, and vice versa.

Ex.1003, ¶20.

The ordinary level of skill is also reflected in the prior art itself. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355, (Fed. Cir. 2001). For example, Burns refers to “throttl[ing] down the communication session with fully encrypted packets to reduce the bandwidth usage of that communication session,” but does not describe how to achieve such bandwidth throttling. Ex.1005, 6:54-56. This reflects that techniques for bandwidth throttling would have been part of a POSITA’s background knowledge.

IX. Claim construction

Claim terms in IPR are construed according to their “ordinary and customary meaning” to those of skill in the art. 37 C.F.R. § 42.100(b). Claims need be construed “only to the extent necessary.” *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017). For the purposes of this proceeding and the grounds presented herein, Petitioner submits that no express constructions are necessary.

X. Relief requested and reasons therefore

Petitioner asks that the Board institute a trial for *inter partes* review and cancel the Challenged Claims in view of the analysis below.

XI. Identification of how the claims are unpatentable

A. Challenged Claims

Petitioner challenges claims 1-30, which are all claims of the '249 patent. A finding that the Challenged Claims are unpatentable in this proceeding will resolve the parties' dispute in the co-pending litigation and obviate any need for a trial regarding the '249 patent.

XII. Statutory grounds

Grounds	Claims	Basis
#1	1-30	35 U.S.C. § 103 over Burns and Yang
#2	4, 7, 13, 15, 22, 24, 30	35 U.S.C. § 103 over Burns, Yang, and Guruswamy

U.S. 8,341,724 to Burns and Sukhanov (Ex.1005, “**Burns**”) issued on December 25, 2012.

U.S. 8,291,495 to Burns, Yang, and Sobrier (Ex.1006, “**Yang**”) issued on October 16, 2012.

U.S. 8,209,756 to Guruswamy and Leung (Ex.1007, “**Guruswamy**”) issued on June 26, 2012.

Burns, Yang, and Guruswamy are prior art under 35 U.S.C. § 102(a)(1).

Petitioner’s § 103 obviousness grounds rely on the combined teachings of the references and not on a physical incorporation of elements. *See In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012).

Petitioner also cites below to additional prior art as evidence of the background knowledge of a POSITA and to provide contemporaneous context to support Petitioner’s assertions regarding what a POSITA would have understood from the prior art in the grounds. *See Yeda Research v. Mylan Pharm. Inc.*, 906 F.3d 1031, 1041-1042 (Fed. Cir. 2018) (affirming the use of “supporting evidence relied upon to support the challenge”); 37 C.F.R. § 42.104(b); *see also K/S HIMPP v. Hear-Wear Techs., LLC*, 751 F.3d 1362, 1365-66 (Fed. Cir. 2014); *Arendi S.A.R.L. v. Apple Inc.*, 832 F.3d 1355, 1363 (Fed. Cir. 2016).

A. Statement of Material Facts

1. A POSITA would have been familiar with the format of TCP and IP packet headers. Ex.1003, ¶¶26-33; Ex.1044, [0003]; Ex.1045, [0015]; Ex.1018, 1:59-60; Ex.1014, 4; Ex.1012, 19; Ex.1016, 19.

2. Decrypting an encrypted communication requires access to a corresponding decryption key, which only the endpoints to the communication would typically have. Ex.1003, ¶¶38-39; Ex.1017, 163-167, 243-247; Ex.1019, [0067] (“As should be understood, because the PIN is encrypted by the terminal using a key that is not know[n] by the merchant or acquirer, the merchant or

acquirer is unable to decrypt the PIN.”); Ex.1020, [0056] (“if an unauthorized client device 110 receives this message, it is unable to decrypt message 178 because it does not have possession of decryption key 112.”); Ex.1021, 1:25-36.

3. The forwarding plane of a network device is responsible for forwarding packets toward their destination with minimal delay. Ex.1003, ¶37; Ex.1022, [0038] (“excessive delay” is a “forwarding plane failure”); Ex.1023, [0030] (“achieving lower latency” is a forwarding plane advantage); Ex.1024, [0029] (describing forwarding plane architectures “to reduce latency and other delays”).

4. A POSITA also would have been familiar with techniques for throttling a communication session, such as by manipulating the TCP congestion window size. Ex.1003, ¶44; Ex.1025, [0079] (“by regulating the advertized window size of each system user, each user's bandwidth can be controlled”); Ex.1026, [0123] (“flow control can be applied... by shrinking the TCP windows being advertised”); Ex.1027, 11:30-45 (an exemplary rate control technique where a “FRAU [frame relay access unit] controls... the TCP window size within the acknowledge packets.”).

5. A TCP port number was commonly considered to be “metadata” associated with and extracted from TCP packets. Ex.1003, ¶35; Ex.1028, [0030] (“**traffic metadata, such as** MAC address, IP address, **TCP port**, UDP port,

etc.”); Ex.1029, [0033], (“a port number (e.g., a Transmission Control Protocol (TCP) port number) may also be included in the metadata record”); Ex.1030, [0057] (“metadata extracted from each outgoing TCP packet may include source and destination IP addresses, source and destination TCP ports”).

6. The closing of a TCP session involves setting a FIN flag value in the TCP packet header. Ex.1003, ¶32; Ex.1016, 752, 760-761, 763, Fig. 47-5; Ex.1031, [0095] (“[A] TCP endpoint that desires to initiate session closing operations sends a FIN command to the other endpoint. A FIN is represented by a TCP header flag.”); Ex.1032, [0109] (“A TCP session is closed by both sides sending packets comprising the FIN flag indicating that the sender has no more data to send.”); Ex.1011, 12 (“The clearing of a connection also involves the exchange of segments, in this case carrying the FIN control flag.”), 23 (Fig. 6 showing “CLOSE” actions accomplished by “s[e]nd FIN”), 38 (connection closing is initiated “by sending a FIN control signal”), 75 (if a packet received where “the FIN bit is set, signal the user ‘connection closing’”).

B. Detailed claim analysis (Grounds 1 & 2)

1. Burns

Burns describes identifying and responding to encrypted communication sessions that are not associated with identifiable network applications. The primary focus is on intrusion detection and prevention systems (IDS) that can detect

encrypted packets and determine if they are linked to unwanted applications or network attacks. The system includes an application identification module to identify the application associated with a packet, an encryption detection module to determine if the packet is encrypted, and an attack detection module to assess if the packet is part of a network attack. If the application cannot be identified and the packet is found to be encrypted, the system assumes the packet is associated with a network attack and takes appropriate action, such as terminating the communication session.

The IDS employs various techniques to detect encryption, including analyzing the randomness of the packet's payload. If the payload exhibits a high degree of randomness¹, it is likely encrypted. The system also considers whether a proper key exchange was detected for the communication session. Legitimate applications typically perform a key exchange before encrypting data, whereas malicious applications may avoid this step to evade detection.

Burns and the '249 patent are both related to the same field of dynamically managing network communications. Ex.1001, 1:19-21, 17:34-35; Ex.1005, 2:47-

¹ As Dr. Mir explains, randomness strengthens cryptographic encryption, making it more difficult for third parties to decipher encrypted data. Ex.1003, ¶57; Ex.1017, 172-180.

49, 5:53-6:8, 6:46-56; Ex.1003, ¶¶56-61. Burns is also directed to addressing the same problem of controlling the transmission of a user's private data, including by restricting the use of encrypted communications. Ex.1001, 1:25-32, 2:42-54; Ex.1005, 1:21-35, 2:21-26.

2. Yang

Yang describes an advanced IDS designed to enhance the accuracy of detecting network attacks by analyzing client-to-server and server-to-client packet flows. The IDS performs an initial identification of the type of software application and communication protocol associated with the incoming packet flow from a client. It then applies a set of patterns to determine if the packet flow represents a network attack. By correlating traffic in both directions and applying compound attack definitions, the IDS can identify sophisticated attack behaviors more quickly and accurately. This approach not only enhances security by reducing the likelihood of undetected attacks but also minimizes the occurrence of false alarms.

Yang and the '249 patent are both related to the same field of dynamically managing network communications. Ex.1001, 1:19-21, 17:34-35; Ex.1006, 1:6-42, 2:47-49, 5:54-6:8, 6:46-56; Ex.1003, ¶¶62-66. Yang is also directed to addressing the same problem of controlling transmission of private data, such as by restricting communications to and from a private computing devices in a private enterprise computing network. Ex.1001, 1:25-32, 2:42-54; Ex.1006, 4:17-45; 5:53-65.

3. Reasons to combine Burns and Yang

Burns expressly refers to Yang by its application number (11/835,923) and incorporates Yang's contents by reference. Ex.1003, ¶¶67-68; Ex.1005, 4:39-45. It would have been obvious to a POSITA considering Burns to also refer to and consider Yang because Burns expressly directs and encourages this.

Burns specifically refers to Yang's disclosure of "techniques for identifying specific applications and protocols." Ex.1005, 4:39-40. A POSITA would have found Yang's techniques instructive regarding the operation of Burns's intrusion detection system, which "attempts to identify applications and protocols for each communication session." Ex.1005, 4:36-39; Ex.1003, ¶¶64-65.

4. Guruswamy (Ground 2)

Guruswamy describes an Intrusion Detection and Prevention (IDP) system like the IDS's described in Burns and Yang. Guruswamy describes a user interface through which an administrator creates a compound network attack definition. The user interface allows the administrator to specify a variety of details for the compound network attack definition, including a "particular protocol," "whether the attack is session based... or transaction based," "patterns... in the form of regular expressions," and "whether the constituent members must be detected in the order presented" or "in any order." Ex.1007, 7:37-8:5.

Guruswamy and the '249 patent are both related to the same field of

dynamically managing network communications. Ex.1001, 1:19-21, 17:34-35; Ex.1007, 1:36-52, 5:23-40, 6:7-38, 6:54-7:22 (e.g., selecting applicable compound attack definitions based on a detected protocol); Ex.1003, ¶¶69-71. Guruswamy is also directed to addressing the same problem of controlling the transmission of private data, such as by restricting communications to and from a private computing devices in a private enterprise computing network. Ex.1001, 1:25-32, 2:42-54; Ex.1007, 2:27-35. And like the '249 patent, Guruswamy contemplates providing “limited access” to “semi-private” resources. Ex.1007, 2:36-42; Ex.1001, 1:62-66 (discussing “partial” blocking of data collection).

5. Reasons to combine Burns, Yang, and Guruswamy (Ground 2)

As discussed above, Burns expressly refers to and incorporates disclosure from Yang. Yang similarly refers to Guruswamy by its application number (11/045,572) and incorporates Guruswamy by reference. Ex.1006, 4:52-58. It would have been obvious to a POSITA considering Burns and Yang to also refer to and consider Guruswamy because Yang expressly directs and encourages it.

Yang specifically refers to Guruswamy’s disclosure of “details on application of attack definitions, e.g. compound attack definitions.” Ex.1006, 4:52-58. A POSITA would have found Guruswamy’s details instructive regarding the operation of Burns’s IDS, where an administrator “may specify attack definitions,” including “compound attack definitions.” Ex.1005, 6:40-45; Ex.1003, ¶¶72-73.

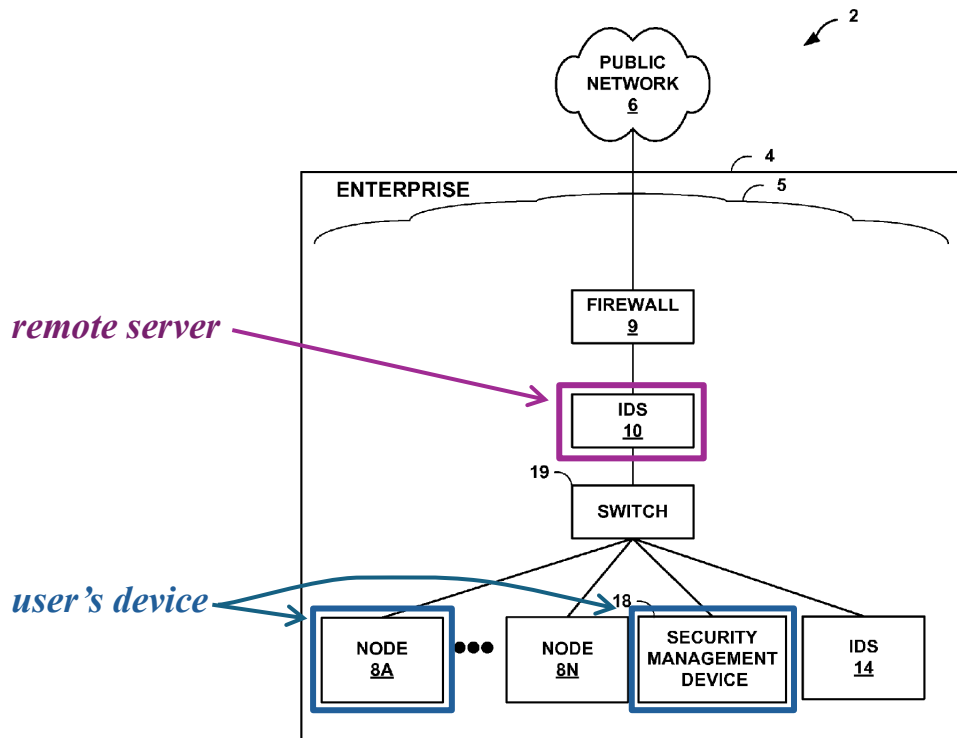
6. Claim 1

[1.0] A method of dynamic management of encrypted data during communication between a remote server and a user's device, the method comprising:

Burns describes dynamically managing encrypted data. For example, Burns describes “blocking unidentified encrypted communication sessions.” Ex.1005, Abstract.

Burns describes “*method*” embodiments, such as the “flowchart illustrating an exemplary operation of an IDS [intrusion detection system]” in Fig. 4. Ex.1005, 9:60-63. In the Fig. 4 method, Burns’s IDS provides “*dynamic management of encrypted data*” by managing encrypted communication sessions, for example identifying and terminating unpermitted sessions. Ex.1003, ¶¶74-76; Ex.1005, 11:30-38.

Burns illustrates the network context of its disclosed IDS in Fig. 1. The IDS “attempts to identify applications and protocols for each communication session between computing nodes 8 and other computing devices in public network 6.” Ex.1005, 4:36-39.



Ex.1003, ¶¶77-78; Ex.1005, Fig. 1(annotated).

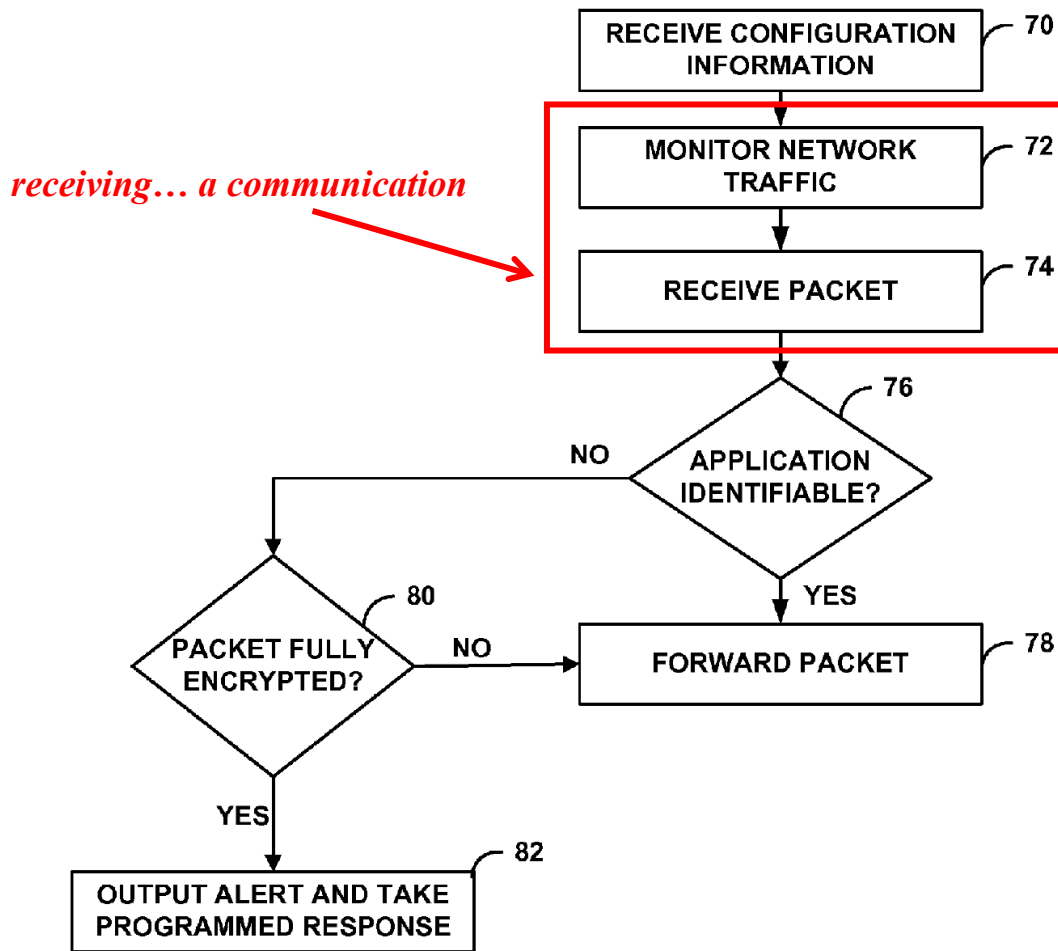
The computing nodes 8 “represent any private computing device with an enterprise network 5, including workstations.” Ex.1005, 4:32-34. It would have been obvious for a computer in an enterprise network, such as a workstation, to be used by a user. Ex.1003, ¶77. Thus, node 8A corresponds to “a user’s device.” Alternatively, Burns describes configuring the IDS via security management device 18. Ex.1005, 6:11-13. It would have been obvious for security management device 18 to also be a workstation computer similar to node 8A that is used by an administrative user. *See* Ex.1005, 6:5-8, 6:37-38; Ex.1003, ¶77. Thus, security management device 18 also corresponds to “a user’s device.”

Figure 1 shows that IDS 10 is separated from node 8A and security

management device 18 by switch 19. Thus, Burns's IDS corresponds to "*a remote server.*" See Ex.1017, 238-242. Since the IDS is on the path between node 8A and public network 6, and since Burns describes communication sessions between computing nodes 8 and devices on public network 6, it would have been obvious for there to be "*communication between*" node 8A and the IDS. Ex.1005, 4:36-39; Ex.1003, ¶79. Alternatively, since Burns contemplates the IDS presenting a user interface to an administrative user at security management device 18 (Ex.1005, 6:5-8, 6:11-13, 6:37-38), it would have been obvious for there to be "*communication between*" security management device 18 and IDS. Ex.1003, ¶79.

[1.1] receiving, by the remote server, a communication comprising encrypted data packets;

Burns illustrates in Fig. 4 the steps of "monitor[ing] network traffic" and "receiv[ing] all packets of a particular communications session." Ex.1005, 10:18-26. These steps are performed by "stateful inspection engine 28 of forwarding plane 22," which is part of the IDS ("*the remote server*"). Ex.1005, 6:29-36, Fig.2. Thus, Burns discloses "*receiving, by the remote server, a communication.*" Ex.1003, ¶¶80-81.



Ex.1003, ¶82; Ex.1005, Fig. 4 (annotated).

Burns’s IDS identifies encrypted packets in communication sessions:

“Encryption detection module 58 attempts to determine whether the packets of the communication session are fully encrypted.” Ex.1005, 9:9-25, 11:3-18, Fig. 4 (step 80). Thus, it would have been obvious for some of the communications received by the IDS to “*compris[e] encrypted data packets.*” Ex.1003, ¶82.

[1.2] determining, by the remote server, a content of at least one data packet of the communication in accordance with characteristics of the at least one data packet, and

As discussed at [1.0], Burns’s IDS corresponds to “*the remote server.*”

The IDS analyzes packets to determine an application or protocol for each communication session:

In one embodiment, IDS 10 attempts to identify applications and protocols for each communication session between computing nodes 8 and other computing devices in public network 6.

Ex.1005, 4:36-39; Ex.1003, ¶83.

Burns refers to Yang for “[e]xemplary techniques for identifying specific applications and protocols.” Ex.1003, ¶84; Ex.1005, 4:39-45.

Yang describes identifying an application or protocol based on a TCP port in a packet header. Ex.1006, 9:46-10:29. This technique employs a “static port mapping” that associates exemplary TCP port numbers with exemplary applications and protocols:

TABLE I

PORT	APPLICATION
20	FTP
22	SSH
23	Telnet
25	SMTP
43	WHOIS
53	DNS
67	BOOTP or DHCP
70	Gopher
79	Finger
80	HTTP
109	POP
110	POP3
113	ident/IRC
118	SQL
119	NNTP
194	IRC
443	HTTPS
445	SMB
564	RTSP

Ex.1006, 10:5-24.

It would have been obvious to a POSITA for Yang’s exemplary applications and protocols to be associated with particular kinds of “*content*.” Ex.1003, ¶¶85-86. For example, the listed application HTTP refers to Hyper-Text Transfer Protocol, which was a well-known protocol associated with communicating web page data. In contrast, the application HTTPS refers to “secure” HTTP, which uses encryption to transfer web page data. The POP and POP3 protocols were well-known to be associated with e-mail data, and the IRC protocol was well-known to be associated with instant messaging data. Ex.1003, ¶86; *see* Ex.1016, 709, Table

43-1; Ex.1033.

A POSITA would have been familiar with the format of TCP and IP packet headers and would have recognized that identifying an application or protocol with Yang's technique would use multiple packet header values ("*characteristics*"). Ex.1003, ¶87; Material Fact #1. First, Yang's technique requires first identifying that an IP packet contains a TCP packet via the IP packet header's "protocol" field. Ex.1006, 5:16-18, 6:24-27; *see* Ex.1034, 14 ("Protocol:... This field indicates the next level protocol used in the data portion of the internet datagram."); Ex.1035, 6 (showing TCP assigned Internet Protocol Number 6). Second, Yang's technique identifies an application or protocol via the source and destination port numbers in the TCP packer's header. Ex.1006, 6:54-58, 17:3-5; *see* Ex.1011, 15 (showing "Source Port" and "Destination Port" in the "TCP Header Format"); Ex.1003, ¶87.

Since Burns's IDS ("*remote server*") identifies an application or protocol ("*content*") based on packet header values ("*characteristics*"), Burns and Yang render obvious "*determining, by the remote server, a content of at least one data packet of the communication in accordance with characteristics of the at least one data packet.*" Ex.1003, ¶88.

[1.3] wherein the content of the at least one data packet is not decrypted by the remote server, and

It was well-known in the art that network devices such as Burns's IDS ("*the remote server*") would not have the key information needed to decrypt encrypted

packets passing through them. Material Fact #2. As Burns explains, packets are “encrypted... by one party and decrypted by the other party to a communication session.” Ex.1005, 2:1-3. Since Burns’s IDS is not a “party” to the communication session itself, it would have been obvious to a POSITA for the IDS not to decrypt packets passing through it. Ex.1003, ¶89.

Consistent with this understanding, Yang’s technique of identifying an application or protocol based on TCP header values does not require decrypting encrypted packets. *See supra*, [1.2]. TCP header values would not have been encrypted. Ex.1003, ¶86.

[1.4] the determination of the content is performed by the remote server in real time during a communication session between the remote server and the user's device;

As discussed at [1.0], Burns’s IDS corresponds to the “remote server.” It would have been obvious for the IDS to operate “in real time” for many reasons. Ex.1003, ¶¶90-93; *see* Ex.1005, 5:61-63; Ex.1017, 228-229.

First, Burns’s IDS employs Yang’s network analysis techniques (*supra*, [1.2]), which Yang explains allows an IDS to “monitor[] network traffic” and “analyze the traffic **in real-time** prior to forwarding.” Ex.1006, 11:63-12:3. Thus, the combination renders obvious for Burns’s IDS to perform the “*determination of the content... in real time.*” Ex.1003, ¶90.

Second, Burns explains that the IDS “**transparently** monitors... and

forwards” traffic:

IDS 20 includes a forwarding plane 22 that **transparently monitors inbound network traffic 24 and forwards the network traffic** as outbound network traffic 26. In the example illustrated by FIG. 2, forwarding plane 22 includes flow analysis module 25, stateful inspection engine 28, protocol decoders 30, forwarding component 31 and security management module 44.

Ex.1005, 6:30-36.

For the IDS to monitor traffic “transparently,” it would have been obvious to a POSITA for the IDS to perform its content determinations “*in real time*” because delaying traffic would interfere with communication sessions. Ex.1003, ¶91. In other words, for the IDS to operate “transparently,” it would need to make content determinations “*in real time*.” Ex.1003, ¶91.

Third, Burns and Yang describe identifying an application associated with a packet with an application identification module 51 within stateful inspection engine 28. Ex.1005, 8:11-23, Fig. 3; Ex.1006, 9:27-49. Thus, application identification module 51 is part of the IDS’s “forwarding plane” that operates transparently. As part of the forwarding plane, it would have been obvious for the application identification module 51 to operate “*in real time*” as claimed. Ex.1003, ¶93; Material Fact #3.

Fourth, it would have been obvious for Burns’s IDS to operate “*in real time*”

because “[m]ost commercial network intrusion detection systems run in real-time.”

Ex.1036, 38; Ex.1003, ¶93; *Randall Mfg. v. Rea*, 733 F.3d 1355, 1363 (Fed. Cir.

2013) (confirming obviousness of using the “prevalent, perhaps even predominant method” in the art).

“...during a communication session...”

As discussed at [1.0], Burns’s IDS monitors traffic flowing between enterprise computers such as node 8 (“*user’s computer*”) and a public network.

Ex.1005, Fig. 1. Since the IDS is on that data path, the data traffic and communication sessions are also “*between*” node 8 and the IDS (“*remote server*”).

Ex.1003, ¶94.

In summary, Burns and Yang describe an IDS that performs real-time analysis of communications passing through the IDS, which renders this limitation obvious. Ex.1003, ¶¶90-94.

[1.5] storing, by the remote server, a preference list;

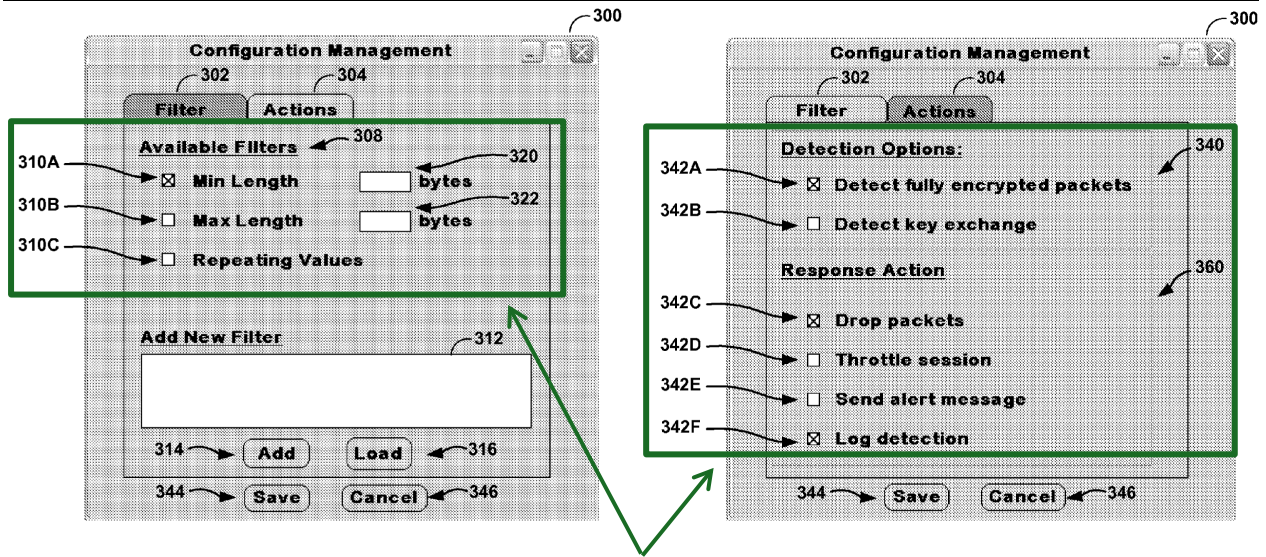
Burns’s IDS is configurable, allowing its operation to be customized to match the preferences of an administrator. Ex.1005, 1:21-35. For example, the administrator may specify patterns, attack definitions, and desired responses:

Initially, security management module 44 receives configuration information from administrator 42 and, in response, configures IDS 20 to monitor a network or portions thereof (subnets) of interest (70). During this

process, configuration manager 44 presents a user interface by which **administrator 42 specifies patterns or other attack definitions** 33. For example, administrator 42 may **configure IDS 20 to block all packets deemed to be fully encrypted** and to log information about the communication session associated with the blocked packets. Administrator 42 may also **configure IDS 20 to throttle down (i.e., bandwidth limit) the communication session** associated with the packets to minimize the bandwidth used by the communication session.

Ex.1005, 9:64-10:9; *see also id.*, 5:29-34 (“a specified list of permitted software applications”).

Burns also illustrates in Figs. 7A-7B (below) various options for an administrator to selectively enable or disable, such as whether to detect repeating values or an encryption key exchange. Ex.1005, 17:58-20:67; *see infra*, claim 4.



Configurable preference list items

Ex.1003, ¶96; Ex.1005, Figs. 7A-7B (annotated).

Burns’s administrator-provided configuration information corresponds to the claimed “*preference list*.”

Burns explains that configuration information is sent to a security management module 44 within the IDS, which then controls other IDS components to operate as configured:

Initially, security management module 44 receives configuration information from administrator 42 and, in response, configures IDS 20 to monitor a network or portions thereof (subnets) of interest (70).

Ex.1005, 9:64-67; *see also id.* 19:7-17, 19:22-33, 6:37-59.

The administrator transmits configuration to the security management module 44 by clicking the “Save” button shown in Figs. 7A-7B. *See, e.g.,* Ex.1005,

19:7-17. Burns expressly discloses that security management module 44 can store information in “a database, a text file, or any appropriate data structure.” Ex.1005, 20:65-67. Thus, it would have been obvious for the configuration information to be saved (“*stor[ed]*”) by security management module 44 to facilitate instructing other IDS components to operate in accordance with the administrator-provided configuration information. Ex.1003, ¶¶95-100.

[1.6] determining, by the remote server, based on a comparison of the determined content, whether to modify the at least one data packet, and

“...a comparison of the determined content...”

As discussed at [1.2], the “*determined content*” of a packet corresponds to its application or protocol.

Burns’s IDS compares the application or protocol of a packet to its configuration information to determine how to handle the packet. As discussed at [1.5], an administrator may configure the IDS to take specified actions, such as selectively allowing or preventing packets based on their applications:

For example, a system administrator may configure IDS 10 to explicitly **allow all identifiable applications**, allow all applications except for a specified list of identifiable applications, or **prevent all communications except for communications from a specified list** of permitted software applications.

Ex.1005, 5:29-34.

To perform such tasks, it would have been obvious for the IDS to compare the application identified for a packet (“*the determined content*”) with the configuration provided by the administrator (“*the preference list*”). Ex.1003, ¶¶101-02. For example, where the administrator configures a “specified list of permitted software applications,” the IDS would compare the application identified for a packet (“*the determined content*”) to the applications on that list (“*the preference list*”). Ex.1005, 5:34; Ex.1003, ¶102; *see* Ex.1017, 229-233.

“determining... whether to modify...”

Burns’s IDS is configurable to take various actions depending on a packet’s identified application or protocol. Exemplary potential actions include (1) dropping (blocking) the packet, (2) closing the related communication session, or (3) throttling (limiting bandwidth consumption of) the related communication session:

In addition, stateful inspection engine 28 may take additional actions, such as **dropping the packets** associated with the communication session, **automatically closing the communication session**, or other actions.

Ex.1005, 7:47-51.

Administrator 42 may also configure IDS 20 to **throttle down (i.e., bandwidth limit) the communication session** associated with the packets to minimize the bandwidth used by the communication session.

Ex.1005, 10:6-9.

Deciding whether to perform any one of these three actions renders obvious the claimed “*determining... whether to modify*” step, as discussed further below.

Ex.1003, ¶¶103-07.

(1) Dropping a packet: The ’249 patent expressly contemplates that “blocking” a packet is a form of “modification.” Ex.1001, 18:14-16. Thus, by deciding whether to block a packet, Burns’s IDS is “*determining... whether to modify the at least one data packet.*” Ex.1003, ¶108; Ex.1005, 7:49-50; *see also* Ex.1016, 763 (one way to close a connection is to modify the packet’s header to add a FIN flag).

(2) Closing a session: A POSITA would have been familiar with techniques for closing a communication session, such as the TCP communication sessions contemplated by Burns. Ex.1003, ¶105; Ex.1005, 7:20-22, 9:16-25, 9:32-33 (“the TCP/IP payload of a packet of the communication session”). As discussed further at [1.7], it would have been obvious for the IDS to close a TCP communication session by modifying a packet header. Thus, by deciding to close a TCP communication session, Burns’s IDS is “*determining... whether to modify the at least one data packet.*” Ex.1003, ¶¶109-10.

(3) Throttling a session: A POSITA also would have been familiar with techniques for throttling a communication session, such as by manipulating the

TCP congestion window size. Material Fact #4. As discussed further at [1.7], it would have been obvious for IDS to throttle a TCP session by modifying a packet header. Thus, by deciding to throttle a TCP communication session, Burns's IDS is "*determining... whether to modify the at least one data packet.*" Ex.1003, ¶107.

[1.7] if so, modifying the at least one data packet; and

As discussed at [1.6], the prior art renders obvious three distinct ways of "*modifying*" a data packet, and one of which is sufficient to render this limitation obvious. Ex.1003, ¶108.

(1) Dropping a packet: Burns describes the IDS dropping a packet. Ex.1005, 7:47-51. Dropping a packet is equivalent to "blocking" the packet from passing through the IDS, which the '249 patent characterizes as a form of "modification." Ex.1001, 18:14-16; Ex.1003, ¶108.

(2) Closing a session: A POSITA would have been familiar with TCP and would have known that closing a TCP session would involve "*modifying*" a packet, specifically, setting a FIN flag value in the TCP packet header. Material Fact #6; Ex.1003, ¶¶109-10.

Based on a POSITA's background knowledge of TCP, it would have been obvious for Burns's IDS to close a session by modifying the TCP header of a packet to set its FIN flag. Ex.1003, ¶¶109-10. The '249 patent expressly states that changing a packet header value is an example of "*modifying*." Ex.1001, 18:14-19.

(3) Throttling a session: A POSITA would have been familiar with techniques for throttling a communication session, such as by manipulating the TCP reception window size. Material Fact #4; *see* Ex.1005, 10:6-9.

The TCP reception window size is a value in a TCP packet header. Ex.1012, 537-538 & Fig. 6-29 (describing the “Window size” field), 547-549; Ex.1034, 15-16 (“Window:... The number of data octets... which the sender of this segment is willing to accept.”). It would have been obvious to a POSITA for throttling a communication session to involve modifying (e.g., reducing) the window size in a TCP header, which is “*modifying*.” Ex.1003, ¶111; Ex.1016, 805-806.

In summary, Burns’s IDS selectively drops or modifies packets, which renders obvious this limitation. Ex.1003, ¶112.

[1.8] sharing, by the remote server, the modified communication.

As discussed at [1.4], Burns’s IDS operates transparently. The IDS includes a “forwarding component 31” that will “forward the packet flows.” Ex.1005, 7:53-55. Thus, it would have been obvious for the IDS to transmit the packet modified as discussed at [1.6]-[1.7]. Ex.1003, ¶113.

Even in the instance where the IDS drops a packet (e.g., blocks a communication), Burns explains that the dropped packet is still “*share[d]*.” For example, the IDS will “log information about the communication session associated with the packet.” Ex.1005, 11:51-52. It would have been obvious for the

log information to include a copy of the packet being blocked to facilitate subsequent investigation. Ex.1003, ¶¶114-15; Ex.1017, 230.

7. Claim 2

[2.0] The method of claim 1,

[2.1] wherein determining a content of the at least one data packet comprises determining at least one data type of the at least one data packet of the communication in accordance with characteristics of the at least one data packet, and

Claim 2 further limits claim 1 such that the determined “*content*” of a packet corresponds to a “*data type*.” The ’249 patent analogizes the concepts of “*content*” and “*data type*.” See, e.g., Ex.1001, 10:46 (“each data type (content)”), 11:13 (“which data types (content)”), Fig.3A (“DATA TYPE (CONTENT)”), Fig.3B (“DATA TYPE (CONTENT)”).

As discussed for claim 1, Burns’s IDS determines a packet’s “*content*,” such as its corresponding application or protocol (as discussed in Yang). It would have been obvious for any of these “*content[s]*” to correspond to a “*data type*” as recited in claim 2. Ex.1003, ¶117. Thus, claim 2 is obvious for the same reasons discussed for claim 1, where a packet’s “*data type*” corresponds to its application or protocol. Ex.1003, ¶118.

For example, Burns’s IDS determines whether a packet contains an application-layer header, and if so, which application the header pertains to. Ex.1005, 3:6-10; see also *id.*, 11:19-22, 11:30-33, 13:5-8, Figs. 3, 4. The

application-specific layer header is a “*data type*.” Ex.1003, ¶¶119-21.

Additionally, Yang describes identifying that a packet’s corresponding application is “unknown,” which is a “*data type*” in the context of the ’249 patent. Ex.1006, 8:14-18; Ex.1003, ¶122; *see* Ex.1001, 9:13-17, 9:25-30 (“unknown data type content”).

Alternatively, the IDS determines whether a packet’s contents correspond to a data type of “encrypted” or “plain text.” It would have been obvious to a POSITA that the encrypted-ness of a packet corresponds to “*at least one data type*.”

Ex.1003, ¶123. It was common in the art to refer to generically refer to a packet’s data type as being “encrypted.” *See* Ex.1037, [0024] (referring to “encrypted” as a “high entropy data type[]”); Ex.1038, [0010] (“data types like... encrypted data”); Ex.1039, [0031] (referring to “encrypted multimedia data” as a “data type[]”).

Burns’s encryption detection module operates consistent with the ’249 patent. The ’249 patent states that “characteristics vectors 304 having at least one identical characteristic may correspond to a particular data type,” and uses “encryption yes/no” as an example of such a characteristic. Ex.1001, 9:6-8.

[2.2] wherein determining whether to modify the at least one data packet is based on the determined data type.

As discussed at [1.6]-[1.7], Burns and Yang render obvious selectively modifying a packet based on its “*determined content*.” As discussed at [2.1], a packet’s “*determined content*” also corresponds to the packet’s “*determined data*

type.” Ex.1003, ¶124.

Alternatively, as discussed at [2.1], the encrypted-ness of a packet corresponds to the claimed “*determined data type*.” Burns describes selectively modifying a packet based in part on whether it is encrypted:

[W]hen the data in the packet is determined to be encrypted, execute a response, and, when the data in the packet is determined not to be encrypted, forwarding the packet.

Ex.1005, 3:46-49.

IDS 10 may permit a communication session that is unencrypted to continue, despite being unable to identify a particular application that is using the communication session. In one embodiment, where IDS 10 determines that a communication session, for which an application cannot be identified, is encrypted, IDS 10 may terminate the communication session.

Ex.1005, 5:55-61; *see also* Ex.1005, 9:40-44. Fig. 4.

By permitting an unencrypted communication session to continue (e.g., by deciding not to modify a packet), Burns’s IDS renders obvious “*determining whether to modify the at least one data packet... based on the determined data type*.” Ex.1003, ¶¶125-26. As discussed at [1.7], it would have been obvious for terminating a prohibited encrypted communication session to involve modifying a

packet, such as by setting the FIN flag or congestion window size in a TCP header.

This further renders obvious “*determining whether to modify the at least one data packet... based on the determined data type.*” Ex.1003, ¶¶126-27.

8. Claim 3

[3.0] The method of claim 2, wherein the preference list includes a list of forbidden and/or allowed data packet types for sharing during communication.

As discussed at [1.5], Burns’s IDS is configurable to permit or allow packets based on characteristics such as their application, protocol, and encrypted-ness:

Where IDS 10 is able to identify the application using a particular communication session, IDS 10 may **either permit or prevent** the communication session from continuing. For example, a system administrator may configure IDS 10 to explicitly allow all identifiable applications, allow all applications except for a **specified list of identifiable applications**, or prevent all communications except for communications from a **specified list of permitted software applications**.

Ex.1005, 5:27-34.

[A]dministrator 42 may configure IDS 20 to take particular actions when a fully encrypted packet is identified. For example, the user interface may permit the administrator to **drop all fully encrypted packets**, log details about the communication session..., throttle down the communication session with fully encrypted packets to

reduce the bandwidth usage of that communication session, or other actions.

Ex.1005, 6:47-56; *see also* Ex.1005, 12:7-26 & 13:9-57.

The administrator-provided “specified list of identifiable applications” that are either allowed or prevented renders obvious the claimed “*list of forbidden and/or allowed data packet types for sharing during communication.*” Ex.1003, ¶128. Similarly, the administrator’s choice to “drop all fully encrypted packets” is an “*list of forbidden... data packet types for sharing during communication.*” Ex.1003, ¶129.

9. Claim 4

[4.0] The method of claim 1, further comprising:

[4.1] determining, by the remote server, at least one communication data pattern in accordance with a behavior range of the data packet, and

As discussed at [1.0], Burns’s IDS corresponds to “*the remote server.*” Burns renders obvious this limitation via three independent descriptions of “*communication data pattern[s],*” discussed below. Any one of them is sufficient to render this limitation obvious. Ex.1033, ¶131.

(1) Repeated data: The IDS identifies a “repeating sequence” in one or more packets of a communication session:

Another exemplary method for determining whether a packet is encrypted includes identifying repeating patterns of varying lengths, e.g., one byte, two bytes, four bytes, or

other within a packet.

...

In one embodiment, encryption detection module 58 **compares data in the packet to other data in the packet to find a repeating sequence.** In one embodiment, encryption detection module 58 stores a sequence of data from each packet in a communication session and determines whether there exists **a repeated value in each packet of the communication session**; in this case, the repeated value may indicate a standard header of an unknown protocol.

Ex.1005, 14:32-35 & 18:30-44; *see also* Ex.1005, 8:6-10, 9:37-40.

A repeating sequence is a “*communication data pattern in accordance with a behavior range of the data packet.*” Ex.1003, ¶133; *see* Ex.1001, 9:65-10:6.

(2) Key exchanges: Burns describes analyzing packets to detecting whether they represent an encryption key exchange:

In particular, when "Detect key exchange" check box 342A is selected, administrator user interface 300 sends a message to security management module 44 to detect key exchanges for monitored communication sessions, or to operate in a detect key exchange mode. Accordingly, security management module 44 instructs stateful inspection engine 28 to **analyze incoming packets to determine whether those packets represent a key exchange** for their corresponding communication session.

Stateful inspection engine 28 then begins to determine whether incoming packets represent a key exchange as part of the communication session. In one embodiment, stateful inspection engine 28 may only determine whether a first set of packets of a communication session represents a key exchange.

Ex.1005, 19:25-38.

Detecting that a series of packets represents a key exchange renders obvious determining a “*communication data pattern in accordance with a behavior range.*”
See also Ex.1003 ¶¶136-38; Ex.1046.

(3) Patterns and anomalies: Burns describes detecting a potential attack by comparing a packet to patterns and anomalies associated with known network attacks:

Each of attack definitions 33 specifies a combination of one or more **patterns specified within patterns table 54** and one or more protocol-specific anomalies specified within anomalies table 56.

Ex.1005, 8:6-10.

[A]pplication identification module 51 sends the packet to attack detection module 52 to inspect the packet and to determine whether the packet is associated with a network attack.

Ex.1005, 10:52-55.

If the average size and frequency of data transmission matches known characteristics for a malicious or unwanted application, the IDS may block further communication of that session.

Ex.1005, 2:34-35

Fig. 3 shows that “attack definitions 33” are an input to the “attack detection module 52.” Ex.1005, Fig. 3. Thus, it would have been obvious to a POSITA for the “attack detection module 52” to detect attacks based on the configurable “attack definitions 33.” Ex.1003, ¶¶134-35. Yang provides more details about “protocol-specific anomaly analysis,” including using anomaly information “to detect sophisticated attack **behaviors**.” Ex.1006, 11:49-51. Thus, the information in patterns table 54 and anomalies table 56 corresponds to a “*communication data pattern in accordance with a behavior range*” as claimed. Ex.1003, ¶¶132-34. By using the attack definitions 33 to identify potential network attacks, Burns and Yang render obvious this limitation.

[4.2] wherein the content of the at least one data packet is not decrypted by the remote server, and

As discussed at [1.3], the techniques employed by Burns’s IDS do not involve decrypting data packets passing through the IDS. Ex.1003, ¶139.

[4.3] the determination of the at least one communication data pattern is performed by the remote server in real time during a communication session between the remote server and the user’s device;

As discussed at [4.1], the “patterns” and “anomalies” taught by Burns and

Yang correspond to “*at least one communication data pattern.*”

Burns’s IDS identifies patterns using encryption detection module 58, which is part of the forwarding plane that operates transparently to communications passing through the IDS. Ex.1005, Figs. 2-3, 14:35-38; *see supra*, [1.4]; Ex.1017, 228-229. As discussed at [4.1], it would have been obvious for known anomalies to be identified by attack detection module 52, which is also part of the forwarding plane. Ex.1005, Figs. 2-3, 7:58-64, 6:33-36.

Thus, it would have been obvious for the IDS to identify a pattern “*in real time during a communication session between the remote server and the user’s device.*” Ex.1003, ¶¶140-41.

[4.4] wherein the determining whether to modify the at least one data packet, by the remote server, is further based on a comparison of the determined data pattern and the preference list,

Burns’s IDS uses the presence of a pattern, such as repeated data, as “an indication that the packet is not random” and “should not be blocked.” Ex.1005, 18:33-36. Randomness is used evaluated against a threshold, which would have been obvious for an administrator to configure:

When the randomness value of a particular TCP/IP payload of the communication session exceeds the **randomness threshold**, encryption detection module 58 determines that the communication session is encrypted.

Ex.1005, 9:33-37.

Thus, Burns renders obvious using the detection of repeated data (a “*determined data pattern*”) to identify a non-encrypted packet. Ex.1003, ¶143. As discussed at [1.6]-[1.7], Burns’s IDS uses various criteria, including the packet’s encrypted-ness, to determine whether to modify a packet. *See, e.g.*, Ex.1005, 5:44-49. Thus, by using repeated data to determine randomness, randomness to determine encrypted-ness, and administrator preferences about which applications are permitted to employ encryption, Burns renders obvious this limitation. Ex.1003, ¶¶142-43.

Additionally, Burns and Yang describe configurably blocking or changing packets for other “*determined data pattern[s]*,” such as a detected encryption key exchange or attack definition. Ex.1005, 6:1-8, 13:33-57, 18:30-44, 18:61-65; Ex.1006, 5:53-65, 7:27-32; *see supra*, [1.5]. In summary, Burns and Yang also render this limitation obvious by selectively modifying packets based on a combination of observed data patterns and administrator’s configuration settings.

[4.5] whether to modify the at least one data packet, and

This limitation is nonsensical because in context (following [4.4]), the claim recites a tautology: “*wherein the determining whether to modify the at least one data packet... is further based on... whether to modify the at least one data packet.*” To the extent that Patent Owner asserts that this language is nevertheless limiting, it would have been obvious. As discussed at [1.6] and [4.4], Burns renders

obvious determining whether to modify a packet. Ex.1003, ¶144. As discussed at [1.7] and [4.6], Burns renders obvious modifying a packet based on the determination of whether to do so. Thus, it would have been obvious for Burns's IDS, having determined whether to modify a packet, to modify the packet. Ex.1003, ¶144.

[4.6] if so, modifying the at least one data packet; and

As discussed at [1.7], Burns renders obvious modifying a packet. Ex.1003, ¶145. As discussed at [4.1]-[4.4], Burns and Yang render obvious determining whether to modify a packet based on the additional criteria recited in claim 4.

[4.7] wherein the preference list includes a list of forbidden and/or allowed data packet types and communication data patterns for sharing during communication.

As discussed at [1.5]-[1.6] and [4.4], an administrator configures Burns's IDS to selectively permit or deny certain applications or protocols ("*forbidden and/or allowed data packet types*"). This configuration information corresponds to "*the preference list.*" Ex.1003, ¶146.

(1) Repeated data: Burns illustrates in Fig. 5 that the IDS may be configured to make determinations based on multiple factors, including whether the application or protocol is known (step 126) and whether a packet is encrypted (step 128). As discussed at [1.5], Burns's IDS is configurable to act in accordance with an administrator's preferences. Thus, it would have been obvious for the

configuration information to include preferences relating to both applications and protocols (“*data packet types*”) and whether a packet contains repeated values indicative of not being encrypted (“*communication data patterns*”). Ex.1003, ¶147; *see, e.g.*, Ex.1005, 5:29-53, 14:32-38.

(2) Key exchanges: Burns’ Fig. 5 shows that the IDS may be configured to make determinations based on multiple factors, including whether the application or protocol is known (step 126) and whether a key exchange is observed (step 130). As discussed at [1.5], Burns’s IDS is configurable to act in accordance with an administrator’s preferences. Thus, it would have been obvious for the configuration information to include preferences relating to both applications and protocols (“*data packet types*”) and whether a key exchange is observed (“*communication data patterns*”). Ex.1003, ¶148; *see, e.g.*, Ex.1005, 5:29-53.

(3) Patterns and anomalies tables: As discussed initially at [4.4] and [1.5], Burns’s administrator provides configuration information for “patterns or other attack definitions”:

During this process, configuration manager 44 presents a user interface by which **administrator 42 specifies patterns** or other attack definitions 33.

...

In one embodiment, an administrator, such as **administrator 42**, may add, edit, or remove filters of encryption detection module 58. A filter instructs

encryption detection module 58 to determine that a packet is random or not random based on certain filtering characteristics. The **filtering characteristics may include**, for example, the number of bytes in the packet, examining bytes in the packet for **a certain pattern**, determining whether a minimum number of byte values above or below the mean occur, or other characteristics of a random or nonrandom packet.

Ex.1005, 16:43-52.

It would have been obvious for the “filtering characteristics” to also include identified applications or protocols (“*data packet types*”), as Burns and Yang discuss elsewhere. Ex.1003, ¶148; *see supra*, [1.2], [2.1]. Filtering “for a certain pattern” renders obvious “*communication data patterns for sharing during communication*” as claimed.

Burns also describes filtering traffic using “compound attack definitions.” Ex.1003, ¶¶149-52; Ex.1005, 6:44-45. Yang provides more details about such compound attack definitions, indicating that they can provide both “specified patterns” and “protocol anomalies,” which may be specified as occurring “in a required order” (a “*communication data pattern[]*”). Ex.1006, 12:55-67.

In summary, the administrator-configured “attack definitions” described by Burns and Yang are a “*list of forbidden... data packet types and communication data patterns for sharing during communication.*” Ex.1003, ¶152. Since the prior

art contemplates having an administrator provide the attack definitions, it would have been obvious for the compound attack definitions to be part of the configuration information (“*the preference list*”) for Burns’s IDS. Ex.1003, ¶150; see Ex.1005, 6:40-44 (“administrator 42 may specify attack definitions 33”).

Ground 2: To the extent that Patent Owner argues that limitation [4.7] requires a single “*list*” whose items identify both “*data packet types*” and “*communication data patterns*,” the addition of Guruswamy to the Burns-Yang combination renders this limitation obvious. Yang further refers to Guruswamy for further details about the use of compound attack definitions. Ex.1006, 4:52-58.

Guruswamy illustrates in Figs. 6-7 a user interface for creating a compound attack definition. Ex.1007, 7:23-44. One input field “specifies a protocol to which the compound attack definition pertains,” for example “HTTP.” Ex.1007, 7:39-41. Another field provides “constituent members of the compound attack definition,” such as “patterns... defined in the form of regular expressions.” Ex.1007, 7:60-67. The specified protocol in a compound attack definition corresponds to a “*data packet type*,” and the specified patterns correspond to “*communication data patterns*.” Ex.1003, ¶152. Guruswamy teaches that when a “match” to a compound attack definition is detected, actions such as “dropping the packets” or “automatically closing the communication session” are taken. Ex.1007, 7:13-19, Fig. 7; see also Ex.1005, 7:47-51.

It would have been obvious to a POSITA for Burns's compound attack definitions to be implemented using the techniques of Yang and Guruswamy because Burns expressly incorporates the contents of Yang, and Yang expressly incorporates the contents of Guruswamy. Ex.1003, ¶152. Thus, it would have been obvious for each of Burns's compound attack definition to specify both a protocol ("*data packet type*") and patterns ("*communication data patterns*").

In summary, the list of "compound attack definitions" described by Burns, Yang, and Guruswamy are a "*list of forbidden... data packet types and communication data patterns for sharing during communication.*" Ex.1003, ¶152. It would have been obvious for the compound attack definitions to be part of the configuration information ("*the preference list*") for Burns's IDS. Ex.1003, ¶152; see Ex.1005, 6:40-44; Ex.1007, 7:23-25.

10. Claim 5

[5.0] The method of claim 1, wherein the modification of the communication is selected from the group consisting of:

Claim 1 does not provide an explicit antecedent basis for "*the modification of the communication*" recited in this limitation. Petitioner interprets this limitation as referring to the step of "*modifying the at least one data packet*" recited in [1.7]. Ex.1003, ¶153.

[5.1] blocking the at least one data packet that is not compatible with the preference list, data nullification, data randomization, content modification, change of encoding, change of file template, change of header, change of footer,

addition of a predetermined data packet, and encryption.

This limitation recites a Markush group. As discussed at [1.7], the prior art discloses multiple actions that correspond to the listed Markush items. For example, Burns describes selectively “dropping” packets based on the IDS’s configuration, which corresponds to “*blocking the at least one data packet that is not compatible with the preference list.*” Ex.1005, 7:47-51; Ex.1003, ¶155. Burns also describes “automatically closing the communication session” and “throttle[ing],” which would obviously involve a “*change of header*” for the reasons discussed at [1.7]. Ex.1005, 7:47-51, 10:6-9, 11:53-55; Ex.1003, ¶¶156-58; Ex.1016, 765, 805-06.

11. Claim 6

[6.0] The method of claim 2, wherein the data type of the at least one data packet is determined based at least in part on metadata of at least one data packet of the communication.

As discussed at [1.2] and [2.1], Burns describes identifying a packet’s *data type* based in part on a TCP port value. The TCP port value in a packet header is “*metadata.*” Ex.1003, ¶159; Ex.1016, 708-709; Material Fact #5.

12. Claim 7

[7.0] The method of claim 4,

[7.1] wherein the data pattern is determined based on communication behavior,

As discussed at [4.1], Burns renders obvious determining a “*data pattern*” in multiple ways. As discussed below, these mappings also render obvious the

claimed “*communication behavior*.” Ex.1003, ¶161.

First, Burns discloses describes “identifying repeating patterns,” a “repeating sequence,” or a “repeated value” in packets of a communication session. Ex.1005, 14:32-35, 18:30-44. Each of these corresponds to a “*data pattern*” as claimed. For the reasons discussed below at [7.2], they are each also a “*communication behavior*.”

Second, Burns discloses identifying whether an encrypted communication session performed a key exchange when establishing the communication session. Ex.1005, 5:64-6:4, 7:32-36, 8:59-54, Fig. 5. Engaging in the negotiation process for an encryption key is a “*communication behavior*.” Similarly, skipping the key exchange process is a “*communication behavior*.” Thus, Burns renders this limitation obvious. Ex.1003, ¶¶162-64.

Ground 2: Third, Guruswamy describes identifying an attack using an attacked definition when “specified patterns and protocol anomalies are satisfied in a required order.” Ex.1007, 7:9-12. It would have been obvious for such patterns and anomalies to be satisfied over the course of multiple packets, and hence, for the attack definition to require packets arrive in a particular order, which is a “*communication behavior*.” Ex.1003, ¶165.

[7.2] wherein the communication behavior consists of data packet frequency, data packet size, data packet speed, data packet count, data packet ratio compared to other data type flow, data packet repetition, or data packet order.

As discussed at [7.1], Burns describes identifying a repeating pattern, sequence, or value across multiple packets in a communication session. Burns's discussion of repetition corresponds to the claimed "*data packet repetition*," which is enumerated here as a type of "*communication behavior*." Ex.1003, ¶166.

Burns also describes identifying a key exchange; it would have been obvious to a POSITA that a key exchange requires that packets be exchanged in a "*data packet order*." Ex.1003, ¶168; *see, e.g.*, Ex.1040, 34 (explaining the TLS Handshake Protocol); Ex.1016, 162-65, 172-177.

Guruswamy's "required order" for meeting the conditions of a compound attack definition renders obvious detecting a "*communication behavior*" including the specified "*data packet order*." Ex.1007, 7:9-12; Ex.1003, ¶169. Further, a POSITA would have understood that an IDS could look to "the average size and frequency of data transmission" to determine whether it "matches known characteristics for a malicious or unwanted application." Ex.1005, 2:33-36. Matching size or frequency of a transmission corresponds to "*data packet frequency*" and "*data packet size*," which are forms of "*communication behavior*." Ex.1003, ¶¶170-71.

13. Claim 8

[8.0] The method of claim 1, further comprising, the remote server maintaining communication between the remote server and the user's device by sharing of the modified communication.

As discussed at [1.4] and [1.8], Burns’s IDS operates “transparently” and forwards selectively modified network traffic. Ex.1005, 6:29-33; *see also* Ex.1006, 7:17-19. In operating transparently, it would have been obvious to a POSITA for the IDS to “*maintain communication*” by “*sharing of the modified communication*.” Ex.1003, ¶172.

14. Claim 9

[9.0] The method of claim 1, wherein the data type is determined based on data packet characteristics selected from a group consisting of: data packet header, data packet footer, version number, IP (Internet Protocol) address, HTTPS (HyperText Transfer Protocol Secure), file extension, encryption method, encoding method, keywords, driver, and communication protocol.

There is no antecedent basis for “*the data type*” in claim 1. Because the ’249 patent uses the terms “data type” and “content” interchangeably (*see supra*, [2.1]), Petitioner analyzes this limitation (for purposes of this proceeding only) as referring to the “*content*” determined in limitation [1.2].

As discussed at [1.2], Burns and Yang render obvious determining a packet’s *content* by comparing a TCP port value to a list of known applications. *See* Ex.1005, 4:36-42; Ex.1006, 9:59-66, Table 1. The TCP port value is in the TCP header, which renders obvious determining a “*data type*” based on a “*data packet header*.” Ex.1003, ¶173; *see* Ex.1011, 15 (showing “Source Port” and “Destination Port” in the “TCP Header Format”). It would have been obvious to a POSITA that identifying a packet as a TCP packet would be based on the protocol field of an IP

header, which corresponds to “*data packet characteristic*” of “*communication protocol*.” Ex.1003, ¶173; *see* Ex.1034, 14 (“Protocol:… This field indicates the next level protocol used in the data portion of the internet datagram.”); Ex.1035, 6 (showing TCP assigned Internet Protocol Number 6); Ex.1016, 457. And as Dr. Mir explains, other fields of a packet header could similarly be used to determine a “*data type*.” Ex.1003, ¶¶174-81; Ex.1005, 4:1; Ex.1006, 5:35-40; Ex.1016, 1346-48 (discussing version information); Ex.1005, 5:29-34, 11:51-52, 20:60-62, Ex.1016, 330-33 (discussing IP address); Ex.1016, 17-20; 103-105 (discussing the use of CRC in a packet footer).

15. Claim 10

[10.0] The method of claim 1, wherein the remote server is a gatekeeper server.

Burns describes that its IDS “intercepts packet flows” and “filter[s] unwanted or malicious packet flows.” Ex.1005, 4:20-25. In doing this, the IDS monitors bidirectional network traffic flow between a client and server. Ex.1005, 7:8-15, 8:59-61. Because it monitors outflow communications (e.g., from a client to a server), Burns’s IDS acts an “exit gatekeeper.” *See* Ex.1001, 16:31-34. Thus, Burns renders obvious for the IDS to be a “*gatekeeper server*.” Ex.1003, ¶182.

16. Claim 11

[11.0] The method of claim 2, further comprising, the remote server issuing an alert to the user device based on the comparison of the determined data type and the preference list.

As discussed at [1.6]-[1.7] and [2.1]-[2.2], Burns's IDS is configurable to modify a packet based on the packet's data type. Burns also describes taking other actions, such as issuing an alert, based on the packet's *data type* and the administrator-provided configuration ("*preference list*"):

When stateful inspection engine 28 determines that the packet flow is encrypted, stateful inspection engine 28 further determines whether characteristics of the communication session indicates that the packet flow is associated with an unwanted software application.... [W]hen stateful inspection engine 28 determines that a key exchange has not occurred, stateful inspection engine 28 takes a programmed response, e.g., to block the communication session, block future communication sessions from a peer of the communication session, **output an alert**, or other actions.

In the event a security risk is detected, stateful inspection engine 28 outputs an alert 40 to security management module 44 for logging and further analysis.

Ex.1005, 7:28-47.

When encryption detection module determines that the packet is fully encrypted, however, ("YES" branch of 80), encryption detection module 58 may **output an alert** and take a programmed response (82).

Ex.1005, 11:30-33.

When administrator 42 selects send alert message check box 342E, and then selects save button 344, administrator user interface 300 sends a corresponding message to security management module 44. In particular, when **"Send alert message" check box 342E** is selected, administrator user interface 300 sends a message to security management module 44 to send an alert message to, e.g., administrator 42. Accordingly, when IDS 20 detects a fully encrypted packet, security management module 44 **sends an alert to administrator 42**. The alert may be any sort of an alert, e.g., a pop-up window, an e-mail, a text message, a page, an audible alert, or any other sort of an alert to inform administrator 42 that IDS 20 has detected, e.g., a fully encrypted packet.

Ex.1005, 20:41-53.

Burns shows administrator 42 in Fig. 2, where the administrator is outside the IDS. Burns does not show administrator 42 in Fig. 1, which shows node 8 and security management device 18, either of which correspond to a *"user's device."* *See supra*, [1.0]. It would have been obvious for administrator 42 to configure IDS 10 from inside the enterprise network, and thus, for administrator 42 to use node 8 or security management device 18. Thus, to send an alert to administrator 42 as describe by Burns, the IDS would send the alert to the node 8 or device 18 used by the administrator (*"issue an alert to the user device"*). Ex.1003, ¶¶183-84.

17. Claim 12

Claim 12 recites limitations very similar to those in claim 1. The difference between them is that where claim 1 recites in [1.2] “*determining... a **content** of at least one data packet... in accordance with **characteristics**,*” claim 12 recites in [12.2] “*determining... at least one **communication data pattern** of at least one data packet... in accordance with a **behavior range**.*” Other limitations are similarly refocused from “*the content*” to “*the determined data pattern.*” Compare [12.4] and [12.6] to [1.4] and [1.6].

As discussed above for claim 4, the prior art renders obvious determining a *communication data pattern* based on a *behavior range*. See *supra*, [4.1]-[4.3]. This *determined pattern* is then used to determine whether to modify a packet, as recited in [12.6]-[12.7]. See *supra*, [4.4]-[4.6]. In short, claim 12 would have been obvious for the same reasons already discussed regarding substantially the same limitations analyzed above in claims 1 and 4. Ex.1003, ¶¶185-93.

18. Claim 13

Claim 13 recites a subset of the features in limitation [4.7] and would have been obvious for the reasons discussed above. Ex.1003, ¶194.

19. Claim 14

Claim 14 recites the same Markush group as [5.1] and would have been obvious for the reasons discussed above. Ex.1003, ¶195.

20. Claim 15

Claim 15 recites substantially the same features as claim 7 and would have been obvious for the reasons discussed above. Ex.1003, ¶¶196-98.

21. Claim 16

Claim 16 recites substantially the same features as claim 8 and would have been obvious for the reasons discussed above. Ex.1003, ¶199.

22. Claim 17

Claim 17 recites the same “*gatekeeper server*” limitation as claim 10 and would have been obvious for the reasons discussed above. Ex.1003, ¶200.

23. Claim 18

Claim 18 recites a feature like claim 11, but with the *alert* being based on the *determined data pattern* instead of the *determined data type*. As discussed at [4.1], Burns renders obvious detecting various *data pattern[s]*, such as the presence or absence of a key exchange negotiation. Burns shows in Fig. 5 that an alert is generated based on criteria such as whether a packet is encrypted and whether a key exchange was detected. Burns shows in Fig. 7B that the response action of “Send alert message” is part of the IDS’s configuration (“*the preference list*” per [1.5]). Ex.1005, 20:41-53. Thus, for the reasons discussed at [11.0] and [4.1], Burns renders obvious issuing an *alert* based on the IDS’s configuration and a detected *data pattern*, which renders claim 18 obvious. Ex.1003, ¶201.

24. Claim 19

[19.0] A system for dynamic management of encrypted data during communication between a remote server and a user's device, the system comprising:

To the extent this preamble is limiting, it would have been obvious. This preamble is similar to that in [1.0], except that claim 19 is a “system” claim. Burns describes managing encrypted data using “systems.” Ex.1005, 1:6-8. For example, Figure 1 shows an “exemplary system.” Ex.1005, 4:20-25. The remaining recited features of the preamble would have been obvious for the reasons discussed above for [1.0]. Ex.1003, ¶202.

[19.1] a memory;

Burns's system includes memory, such as “random access memory (RAM),” “a hard drive, a network drive, [or] a flash memory stick.” Ex.1005, 8:4-6, 18:51-53; *see also id.*, 21:24-31; Ex.1003, ¶203.

[19.2] a communication data type database comprising at least one communication data type corresponding to sharing of at least one data packet with the user's device;

As discussed at [1.2], Burns's IDS “attempts to identify applications and protocols for each communication session between computing nodes 8 and other computing devices in public network 6.” Ex.1005, 4:36-39. Burns refers by application number to Yang, which provides more details about identifying applications. Ex.1005, 4:39-45, 9:46-10:29.

As discussed at [1.2], it would have been obvious to a POSITA for Yang's exemplary applications and protocols to be associated with particular kinds of "*content*." And as discussed at [2.1], the '249 patent analogizes the concepts of "*content*" and "*data type*." See, e.g., Ex.1001, 10:46, 11:13, Figs. 3A-3B. Thus, it would have been obvious to a POSITA for Yang's applications and protocols to be associated with particular *data types*. See *supra*, [1.2] & [2.1]; see also Ex.1003, ¶204 (explaining Yang's "static port mapping" and corresponding data types to ports); Ex.1006, 9:59-10:24, Table 1; Ex.1016, 709, Table 43-1. Thus, Yang's static port mapping renders obvious the claimed "*communication data type database*." Ex.1003, ¶205.

Burns discusses analyzing traffic associated with some of the applications and protocols in Yang's table. See Ex.1005, 10:32-45 (identifying "HTTP traffic" and "SSH traffic"). Thus, it would have been obvious for the data types in Yang's static port mapping to be "*corresponding to sharing of at least one data packet with the user's device*." Ex.1003, ¶206.

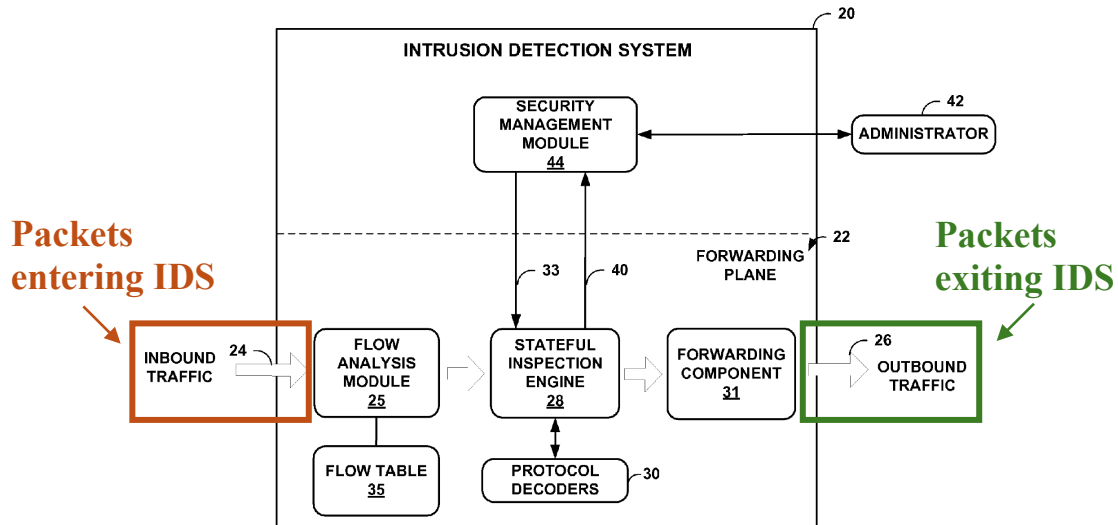
[19.3] a preference list database,

See supra [1.5]. Ex.1003, ¶207.

[19.4] a communication module configured to allow communication between the remote server and the at least one user's device, and

Burns illustrates in Fig. 2 that network traffic flows into and out of the IDS. It would have been obvious for this network traffic to include packets entering and

exiting IDS via network interfaces. Ex.1003, ¶¶208-09. An IDS network interface corresponds to the claimed “*communication module*.”



Ex.1003, ¶¶208-09; Ex.1005, Fig. 2 (annotated).

As discussed at [1.0], Burns shows in Fig.1 that the IDS manages communications “between computing nodes 8 and other computing devices in public network 6.” Ex.1005, 4:36-39. Thus, it would have been obvious for the IDS’s network interface(s) to “*allow communication between the remote server [i.e., the IDS itself] and the at least one user’s device.*” Ex.1003, ¶209.

[19.5] a processor, coupled to the preference list database,

Burns describes a “programmable processor.” Ex.1005, 3:37. Burns also explains that the IDS functionality of the IDS may be implemented using a “general-purpose processor”:

Methods described herein may be performed in hardware, software, or any combination thereof within a

network device. For example, methods described herein may be performed by an application specific integrated circuit (ASIC) or **a general-purpose processor**. Methods described herein may also be embodied in a computer readable medium containing instructions. Instructions embedded in a computer readable medium may cause a programmable processor, or other processor, to perform the method, e.g. when the instructions are executed.

Ex.1005, 21:14-23; Ex.1003, ¶210.

As discussed at [19.3], the IDS is configurable to operate in accordance with an administrator's preferences. It would have been obvious for Burns's programmable or general-purpose processor to have access to the administrator-provided configuration information (*coupled to the preference list database*) so that the processor would operate as administrator configured the IDS. Ex.1003, ¶211.

[19.6] wherein the processor is configured to instruct the remote server to:

As discussed at [19.5], Burns's IDS includes a "*processor*." As discussed at [1.0], the IDS corresponds to the "*remote server*."

It would have been obvious for the IDS's processor to "*instruct*" the IDS to perform its various operations because, as was commonly known, a processor is the "brain" of a computer that is responsible for directing its operations. Ex.1003, ¶212; Ex.1041, [0025] ("central processing unit 22 is a brain of computer system" that will "direct the operation of the internal processor components."); Ex.1042,

1:17 (“Processors are the brain of computer systems...”); Ex.1043, [0031]

(“Executed by a device’s processor(s), these instructions direct the operation of the device. These instructions can be stored in a memory of the computer. Instructions can invoke other instructions.”). Burns discloses that “[i]nstructions embedded in a computer readable medium may cause a programmable processor, or other processor to perform the method,” wherein the method is the identification of and response to an encrypted packet. Ex.1005, 21:20-22.

Mapping the claimed “*processor*” to a component within the “*remote server*” is consistent with the ’249 specification. See Ex.1001, 13:41-42 (“In some embodiments, processor 402 may be embedded in server 401.”).

Thus, Burns renders obvious the claimed “*processor*.” Ex.1003, ¶213.

[19.7]-[19.13]

The remaining limitations of claim 19 correspond to the steps of claim 1, specifically [1.2]-[1.4] and [1.6]-[1.8], and would have been obvious for the reasons discussed above. Ex.1003, ¶¶214-20.

25. Claims 20-29

Dependent claims 20-29 correspond to dependent claims previously discussed above. Specifically, these dependent claim limitations are substantially or entirely identical as indicated below:

Claim 19 dependent limitations	Corresponding Claim 1 or Claim 12 dependent limitations
[20.0]-[20.2]	[2.0]-[2.2]
[21.0]	[3.0], [13.0]
[22.0]-[22.7]	[4.0]-[4.7]
[23.0]	[6.0]
[24.0]-[24.2]	[7.0]-[7.2]
[25.0]-[25.1]	[5.0]-[5.1], [14.0]
[26.0]	[8.0], [16.0]
[27.0]	[9.0]
[28.0]	[10.0], [17.0]
[29.0]	[11.0], [18.0]

See Ex.1003, ¶¶221-42.

26. Claim 30

[30.0] The system of claim 22,

[30.1] wherein the preference list includes a list of forbidden and/or allowed data packet types and communication data patterns for sharing during communication, and

See *supra*, [3.0] and [4.7]. Ex.1003, ¶244.

[30.2] wherein, if the at least one data type of the at least one data packet is included in the list of forbidden and/or allowed data packets, the processor is configured to determine whether to modify the at least one data packet

There is no antecedent basis for “*the at least one data type of the at least one data packet.*” Since the ’249 patent uses the terms “*content*” and “*data type*” interchangeably (*see supra*, [2.1]), this language is analyzed as if it referred to the “*content of at least one data packet*” introduced in [19.7].

There is also no antecedent basis for “*the list of forbidden and/or allowed data packets.*” For purposes of this proceeding, this language is analyzed as if it

referred to the “*list of forbidden and/or allowed data packet types*” introduced in [30.1]. As discussed at [1.6], [2.2], and [4.4], the prior art renders obvious determining whether to modify a packet. Since Burns renders obvious making such determinations generally, it would have been obvious for Burns’s IDS to make such a determination in the instance that a “*data packet is included in the list of forbidden and/or allowed data packet[type]s.*” Ex.1003, ¶¶245-46.

[30.3] based on the determined data type and the preference list without performing a comparison of a determined data pattern and the preference list.

There is no antecedent basis for “*the determined data type.*” Since the ’249 patent uses the terms “*content*” and “*data type*” interchangeably (*see supra*, [2.1]), this language is analyzed as if it referred to “*the determined content*” introduced in [19.7] and [19.10].

As discussed at [1.6], [2.2], and [4.4], the prior art renders obvious determining whether to modify a packet based on a determined “*content,*” i.e., “*data type.*” While Burns also describes making a modification determination based on a “*data pattern*” as discussed at [4.4], Burns explains that such pattern-based determinations are optional. For example, Burns describes immediately forwarding a packet (hence, determining not to modify the packet) when the packet’s associated application (“*data type*”) is identifiable:

When application identification module 51 is able to determine the application associated with the

communication session ("YES" branch of 76), application
identification module 51 forwards the packet (78).

Ex.1005, 10:46-49; *see also* Ex.1005, Fig. 4.

In another example, Burns's technique of detecting repeated values may be disabled by an administrator's configuration of the IDS. Ex.1005, Fig. 7A (checkbox 310C), 18:1-7, 18:30-44 ("...IDS 20 **may** utilize a repeating values filter."). Similarly, the administrator can disable the detection of encryption key exchanges. Ex.1005, Fig. 7A (checkbox 342B), 19:22-42.

In summary, Burns describes various example scenarios in which the IDS would determine whether to modify a packet based solely on its determined "*data type*," and without consideration of Burns's various "*pattern*"-based techniques, which renders this limitation obvious. Ex.1003, ¶¶247-50.

XIII. Discretionary denial is inappropriate.

A. No §325(d) denial

Burns, Yang, and Guruswamy were not cited or considered during prosecution of the '249 patent. Two other "Guruswamy" patents were identified on a Notice of References Cited during prosecution of the '249 patent, but neither was ever substantively discussed. *See* Ex.1002, 133. While those other "Guruswamy" patents describe an IDS, they do not have technical disclosures similar to (1) Burns's techniques for identifying encrypted packets based on randomness or repeated values; (2) Yang's techniques for identifying applications and protocols;

(3) Guruswamy’s user interface for creating a compound attack definition.

Thus, discretionary denial under §325(d) is inappropriate at least because the first part of the *Advanced Bionics* framework is not met. *Advanced Bionics, LLC v. MED-EL Elektromedizinische Geräte GmbH*, IPR2019-01469, Paper 6 (Feb. 13, 2020) (precedential).

B. No *Fintiv* denial

The six factors considered for §314 discretionary denial strongly favor institution given Petitioner’s prompt filing and the compelling merits of this case. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (Mar. 20, 2020) (precedential).

1. No evidence regarding a stay

No motion to stay has been filed, so the Board should not infer the outcome of such a motion. *Sand Revolution II LLC v. Continental Intermodal Group – Trucking LLC*, IPR2019-01393, Paper 24 at 7 (PTAB June 16, 2020) (informative); *see also Dish Network L.L.C. v. Broadband iTV, Inc.*, IPR2020-01359, Paper 15 (Feb. 12, 2021) (“It would be improper to speculate, at this stage, what the Texas court might do regarding a motion to stay...”). Thus, this factor is neutral.

2. Parallel proceeding trial date

The co-pending district court case was filed in the Eastern District of Texas on October 21, 2024. Ex.1008. Trial is currently scheduled for June 22, 2026, and the most recent statistics currently available for the Eastern District of Texas show

a median time-to-trial of 23.0 months, suggesting an expected trial date in September 2026. Ex.1009, 1; Ex.1010, 35. The Board’s final written decision is expected in approximately October 2026. Thus, this factor is neutral.

3. Investment in the parallel proceeding

The co-pending litigation is in its early stages, with major activities including a Markman hearing and the close of discovery scheduled for after the Board’s expected decision on institution. *See* Ex.1009, 4 (Claim Construction hearing on December 15, 2025; close of fact discovery on January 21, 2026; close of expert discovery on March 10, 2026). Moreover, Petitioner has acted with diligence to prepare this Petition approximately five months after Petitioner was served with a complaint. Under *Fintiv*, Petitioner’s prompt filing “weigh[s] against exercising the authority to deny institution.” *Fintiv*, Paper 11 at 11 (“If the evidence shows that the petitioner filed the petition expeditiously, such as promptly after becoming aware of the claims being asserted, this fact has weighed against exercising the authority to deny institution under NHK”).

4. Overlapping issues with the parallel proceeding

If the Board institutes an IPR, Petitioner hereby stipulates that Petitioner will not pursue in the co-pending district court litigation the specific grounds asserted here, or any other ground that could have been reasonably raised against the Challenged Claims in an IPR (i.e., grounds that could have been raised under

§§102 or 103 on the basis of prior art patent or printed publications). See *Sotera Wireless, Inc. v. Masimo Corp.*, IPR2020-00109, Paper 12 at 18-19 (Dec. 1, 2020).

Petitioner does not relinquish any rights or opportunities to challenge the Asserted Patents' claims on any other ground (i.e., any ground that could not have been raised under §§102 or 103 on the basis of prior art patent or printed publications) or based on Cisco's own prior art related to the functionality accused of infringement in the co-pending district court litigation, including any grounds that Cisco may use as a defense to infringement under 35 U.S.C. §273.

5. Identity of parties

Petitioner is a defendant in the litigation. That is true of most Petitioners in IPR proceedings. Accordingly, this factor should not be a basis for denying institution.

6. Other circumstances

The prior art presented in this Petition presents a particularly strong case on the merits. As explained above, in the '249 patent claims recite steps for handling potentially encrypted network traffic that were already known many years before its earliest priority date. The relied-upon prior art references include explicit teachings that would have motivated their combination. The evidence of unpatentability weighs heavily against discretionary denial, and the Board should institute trial irrespective of how it evaluates the other factors.

C. No General Plastic denial

The '249 patent has not been challenged in any prior IPR petition, so none of the *General Plastic* discretionary institution factors apply to this Petition. *General Plastic Indus. Co., Ltd. v. Canon Kabushiki Kaisha*, IPR2016-01357, Paper 19, 16 (Sept. 6, 2016) (Section II.B.4.i. precedential).

XIV. Conclusion

Petitioner has established a reasonable likelihood that the Challenged Claims are unpatentable.

Respectfully submitted,

Dated: April 10, 2025
HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, Texas 75201
Customer No. 27683

/Theodore M. Foster/
Theodore M. Foster
Lead Counsel for Petitioner
Registration No. 57,456

CLAIMS APPENDIX

[1.0] A method of dynamic management of encrypted data during communication between a remote server and a user's device, the method comprising:

[1.1] receiving, by the remote server, a communication comprising encrypted data packets;

[1.2] determining, by the remote server, a content of at least one data packet of the communication in accordance with characteristics of the at least one data packet, and

[1.3] wherein the content of the at least one data packet is not decrypted by the remote server, and

[1.4] the determination of the content is performed by the remote server in real time during a communication session between the remote server and the user's device;

[1.5] storing, by the remote server, a preference list;

[1.6] determining, by the remote server, based on a comparison of the determined content, whether to modify the at least one data packet, and

[1.7] if so, modifying the at least one data packet; and

[1.8] sharing, by the remote server, the modified communication.

[2.0] The method of claim 1,

[2.1] wherein determining a content of the at least one data packet comprises determining at least one data type of the at least one data packet of the communication in accordance with characteristics of the at least one data packet, and

[2.2] wherein determining whether to modify the at least one data packet is based on the determined data type.

[3.0] The method of claim 2, wherein the preference list includes a list of forbidden and/or allowed data packet types for sharing during communication.

[4.0] The method of claim 1, further comprising:

[4.1] determining, by the remote server, at least one communication data pattern in accordance with a behavior range of the data packet, and

[4.2] wherein the content of the at least one data packet is not decrypted by the remote server, and

[4.3] the determination of the at least one communication data pattern is performed by the remote server in real time during a communication session between the remote server and the user's device;

[4.4] wherein the determining whether to modify the at least one data packet, by the remote server, is further based on a comparison of the determined data pattern and the preference list,

[4.5] whether to modify the at least one data packet, and

[4.6] if so, modifying the at least one data packet; and

[4.7] wherein the preference list includes a list of forbidden and/or allowed data packet types and communication data patterns for sharing during communication.

[5.0] The method of claim 1, wherein the modification of the communication is selected from the group consisting of:

[5.1] blocking the at least one data packet that is not compatible with the preference list, data nullification, data randomization, content modification, change of encoding, change of file template, change of header, change of footer, addition of a predetermined data packet, and encryption.

[6.0] The method of claim 2, wherein the data type of the at least one data packet is determined based at least in part on metadata of at least one data packet of the communication.

[7.0] The method of claim 4,

[7.1] wherein the data pattern is determined based on communication behavior,

[7.2] wherein the communication behavior consists of data packet frequency, data packet size, data packet speed, data packet count, data packet ratio compared to

other data type flow, data packet repetition, or data packet order.

[8.0] The method of claim 1, further comprising, the remote server maintaining communication between the remote server and the user's device by sharing of the modified communication.

[9.0] The method of claim 1, wherein the data type is determined based on data packet characteristics selected from a group consisting of: data packet header, data packet footer, version number, IP (Internet Protocol) address, HTTPS (HyperText Transfer Protocol Secure), file extension, encryption method, encoding method, keywords, driver, and communication protocol.

[10.0] The method of claim 1, wherein the remote server is a gatekeeper server.

[11.0] The method of claim 2, further comprising, the remote server issuing an alert to the user device based on the comparison of the determined data type and the preference list.

[12.0] A method of dynamic management of encrypted data during communication between a remote server and a user's device, the method comprising:

[12.1] receiving, by the remote server, a communication comprising encrypted data packets;

[12.2] determining, by the remote server, at least one communication data pattern of at least one data packet of the communication in accordance with a behavior range of the data packet, and

[12.3] wherein content of the at least one data packet is not decrypted by the remote server, and

[12.4] the determination of the at least one data pattern is performed by the remote server in real time during a communication session between the remote server and the user's device;

[12.5] storing, by the remote server, a preference list;

[12.6] determining, by the remote server, based on a comparison of the determined data pattern and the preference list, whether to modify the at least one data packet, and

[12.7] if so, modifying the at least one data packet; and

[12.8] sharing, by the remote server, the modified communication.

[13.0] The method of claim 12, wherein the preference list includes a list of forbidden and/or allowed communication data patterns for sharing during communication.

[14.0] The method of claim 12, wherein the modification of the communication is selected from the group consisting of: blocking the at least one data packet that is not compatible with the preference list, data nullification, data randomization, content modification, change of encoding, change of file template, change of header, change of footer, addition of a predetermined data packet, and encryption.

[15.0] The method of claim 12,

[15.1] wherein the communication data pattern of the at least one data packet is determined based on communication behavior,

[15.2] wherein the communication behavior consists of data packet frequency, data packet size, data packet speed, data packet count, data packet ratio compared to other data type flow, data packet repetition, or data packet order.

[16.0] The method of claim 12, further comprising, the remote server maintaining communication between the remote server and the user's device by sharing of the modified communication.

[17.0] The method of claim 12, wherein the remote server is a gatekeeper server.

[18.0] The method of claim 12, further comprising, the remote server issuing an alert to the user device based on the comparison of the determined data pattern and the preference list.

[19.0] A system for dynamic management of encrypted data during communication between a remote server and a user's device, the system comprising:

[19.1] a memory;

[19.2] a communication data type database comprising at least one communication data type corresponding to sharing of at least one data packet with the user's device;

[19.3] a preference list database,

[19.4] a communication module configured to allow communication between the remote server and the at least one user's device, and

[19.5] a processor, coupled to the preference list database,

[19.6] wherein the processor is configured to instruct the remote server to:

[19.7] determine a content of at least one data packet of the communication in accordance with characteristics of the at least one data packet, and

[19.8] wherein the content of the at least one data packet is not decrypted by the remote server, and

[19.9] the determination of the content is performed by the remote server in real time during a communication session between the remote server and the user's device,

[19.10] based on a comparison of the determined content and the preference list,

[19.11] determine whether to modify the at least one data packet, and

[19.12] if so, modify the at least one data packet; and

[19.13] share the modified communication.

[20.0] The system of claim 19,

[20.1] wherein determining a content of the at least one data packet comprises determining at least one data type of the at least one data packet of the communication in accordance with characteristics of the at least one data packet, and

[20.2] wherein determining whether to modify the at least one data packet is based on the determined data type.

[21.0] The system of claim 20, wherein the preference list database comprises a list of forbidden and/or allowed data packet types for sharing during communication.

[22.0] The system of claim 19,

[22.1] wherein, the processor is further configured to determine at least one communication data pattern in accordance with a behavior range of the data packet, and

[22.2] wherein the content of the at least one data packet is not decrypted by the remote server, and

[22.3] the determination of the at least one communication data pattern is performed by the remote server in real time during a communication session between the remote server and the user's device; and

[22.4] wherein the determining whether to modify the at least one data packet, by the remote server, is further based on a comparison of the determined data pattern and the preference list,

[22.5] whether to modify the at least one data packet, and

[22.6] if so, modifying the at least one data packet; and

[22.7] wherein the preference list includes a list of forbidden and/or allowed data packet types and communication data patterns for sharing during communication.

[23.0] The system of claim 20, wherein the data type of the at least one data packet is determined based at least in part on metadata of at least one data packet of the communication.

[24.0] The system of claim 22,

[24.1] wherein the at least one communication data pattern is determined in accordance with a behavior range of the data packet,

[24.2] wherein the behavior range is selected from a group consisting of: data packet frequency, data packet size, data packet speed, data packet count, data packet ratio compared to other data type flow, data packet repetition, and data packet order.

[25.0] The system of claim 19, wherein the modification of the communication is selected from the group consisting of

[25.1] blocking the at least one data packet that is not compatible with the preference list, data nullification, data randomization, content modification, change

of encoding, change of file template, change of header, change of footer, addition of a predetermined data packet and encryption.

[26.0] The system of claim 19, wherein the remote server is configured to maintain communication between the remote server and the user's device by sharing of the modified communication.

[27.0] The system of claim 20, wherein the data type is determined based on data packet characteristics selected from a group consisting of: data packet header, data packet footer, version number, IP (Internet Protocol) address, HTTPS (HyperText Transfer Protocol Secure), file extension, encryption method, encoding method, keywords, driver, and communication protocol.

[28.0] The system of claim 19, wherein the remote server is a gatekeeper server.

[29.0] The system of claim 20, wherein the processor is configured to issue an alert to the user device based on the comparison of the determined data type and the preference list.

[30.0] The system of claim 22,

[30.1] wherein the preference list includes a list of forbidden and/or allowed data packet types and communication data patterns for sharing during communication, and

[30.2] wherein, if the at least one data type of the at least one data packet is included in the list of forbidden and/or allowed data packets, the processor is configured to determine whether to modify the at least one data packet

[30.3] based on the determined data type and the preference list without performing a comparison of a determined data pattern and the preference list.

XV. Mandatory notices

A. Real party-in-interest

Pursuant to 37 C.F.R. §42.8(b)(1), Petitioner certifies that the real party-in-interest is Cisco Systems, Inc.

B. Related matters

Pursuant to 37 C.F.R. §42.8(b)(2), to the best knowledge of the Petitioner, the '249 Patent is or was involved in the following cases:

Case Heading	Number	Court	Filed
<i>QPrivacy USA LLC v. Cisco Systems, Inc.</i>	No. 2:24-cv-00855	E.D. Tex.	Oct. 21, 2024
<i>Ex parte reexamination</i>	90/019,896	USPTO	April 1, 2025

Petitioner is also concurrently filing a petition for *inter partes* review (“IPR”) of claims 1-19 of U.S. Patent No. 11,106,824.

C. Lead and back-up counsel and service information

Lead Counsel

Theodore M. Foster
HAYNES AND BOONE, LLP
2801 Harwood St. Suite 2300
Dallas, TX 75201

Phone: (303) 382-6205
Fax: (214) 200-0853
ipr.theo.foster@haynesboone.com
USPTO Reg. No. 57,456

Back-up Counsel

David L. McCombs
HAYNES AND BOONE, LLP
2801 Harwood St. Suite 2300
Dallas, TX 75201

Phone: (214) 651-5533
Fax: (214) 200-0853
david.mccombs.ipr@haynesboone.com
USPTO Reg. No. 32,271

Eugene Goryunov
HAYNES AND BOONE, LLP
2801 Harwood St. Suite 2300
Dallas, TX 75201

Phone: (312) 216-1630
Fax: (214) 200-0853
eugene.goryunov.ipr@haynesboone.com
USPTO Reg. No. 61,579

Allyson Malecha
HAYNES AND BOONE, LLP
2801 Harwood St. Suite 2300
Dallas, TX 75201

Phone: (415) 293-8937
Fax: (214) 200-0853
al.malecha.ipr@haynesboone.com
USPTO Reg. No. 82,832

Please address all correspondence in this proceeding to lead and back-up counsel. Petitioner consents to service in this proceeding by email at the addresses above.

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. §42.24(d), Petitioner hereby certifies, in accordance with and reliance on the word count provided by the word-processing system used to prepare this Petition, that the number of words in this paper is 12,800. Pursuant to 37 C.F.R. §42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, appendix of exhibits, and any claim listing.

Dated: April 10, 2025

/Theodore M. Foster/
Theodore M. Foster
Lead Counsel for Petitioner
Registration No. 57,456

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§ 42.6(e) and 42.105, the undersigned certifies that a true and correct copy of the foregoing was served on the Patent Owner as detailed below:

<i>Date of service</i>	April 10, 2025
<i>Manner of service</i>	Fed Ex
<i>Documents served</i>	Petitioner for <i>Inter Partes</i> Review Under 35 U.S.C. § 312 and 37 C.F.R. § 42.104; Petitioner’s Power of Attorney; Exhibit List; and Exhibits Ex.1001-Ex.1014 and Ex.1016-Ex.1046.
<i>Persons Served</i>	Pearl Cohen Zedek Latzer Baratz LLP 7 Times Square, 19th Floor New York, NY 10036

/Theodore M. Foster/
Theodore M. Foster
Lead Counsel for Petitioner
Registration No. 57,456