UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.,
Petitioner,

IPR2025-00837
Patent No. 11,106,824

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

TABLE OF CONTENTS

PETITIONER'S EXHIBIT LIST

Ex.1001	U.S. 11,106,824
Ex.1002	Prosecution History of U.S. 11,106,824
Ex.1003	Declaration of Nader Mir, Ph.D. under 37 C.F.R. § 1.68
Ex.1004	*Curriculum Vitae* 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 Publication No. 2005/0078668 to Wittenberg et al. ("Wittenberg")
Ex.1008	Complaint, *QPrivacy USA LLC v. Cisco Systems, Inc*., No. 2:24-cv-00855 (E.D. Tex. Oct. 21, 2024).
Ex.1009	Docket Control Order, *QPrivacy USA LLC v. Cisco Systems, Inc*., 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 Pub. No. 2006/0215695 to Olderdissen
Ex.1014	Perlman, Radia, "Interconnections: Bridges, Routers, Switches, and Internetworking Protocols," 2d ed. (2000).
Ex.1015	*Reserved*
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).
Ex.1047	U.S. Patent Publication No. 2002/0143897 to Patil
Ex.1048	U.S. Patent Publication No. 2007/0248105 to Shinoda et al.
Ex.1049	U.S. Patent Publication No. 2008/0016570 to Capalik
Ex.1050	U.S. Patent No. 7,127,743 to Khanolkar et al.
Ex.1051	U.S. Patent Publication No. 2006/0179472 to Chang et al.
Ex.1052	U.S. Patent Publication No. 2004/0187028 to Perkins et al.
Ex.1053	U.S. Patent Publication No. 2009/0254970 to Agarwal et al.
Ex.1054	U.S. Patent Publication No. 2013/0179753 to Flynn et al.
Ex.1055	U.S. Patent Publication No. 2014/0207997 to Peterson et al.
Ex.1056	U.S. Patent Publication No. 2012/0303952 to Smith et al.

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 1-20 (hereinafter, the "Challenged Claims") of U.S. 11,106,824 (the "'824 patent," Ex.1001).

The '824 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, 1:29-32, Abstract, Claim 1. As this petition demonstrates, such ideas were known before the '824 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 '824 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 '824 patent

The '824 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, 2:4-9. The content of these data packets is encrypted. Ex.1001, 12:38-43. The user's device is "configured to provide a response corresponding to the received request." Ex.1001, 2:8-10. Characteristics of a data packet are used to identify a data packet's "data type." Ex.1001, 9:9-13. Similarly, dynamic features of a data packet are used to identify a data packet's "data pattern." Ex.1001, 9:61-10:5.

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, 2:13-15, 2:25-28. A remote device applies these privacy preferences to data packets. Ex.1001, 11:10-16. If the privacy preferences dictate that a certain "data packet is forbidden for sharing," the content of the data packet can be modified. Ex.1001, 10:48-54. Modifying the packet includes blocking the packet. Ex.1001, 2:44-50.

The '824 patent specification focuses on restricting access to a user's "private data." Ex.1001, 7:48-54. In co-pending litigation, the patent owner asserts the '824 patent's privacy-focused claims against products that "implement Encrypted Traffic Analytics." Ex.1008, ¶3.

VI. Prosecution history

The '824 patent issued from a PCT application filed March 28, 2018, as U.S. Application No. 16/603,252. Ex.1001. The '824 patent claims priority to Israeli Patent No. 251,683, filed on April 9, 2017. Ex.1001, code (30). The Examiner's statement of reasons for allowance identified the entire text of claim 1 as the "limitation" that was not taught or rendered obvious in the prior art of record. Ex.1002, 51-52.

VII. Effective priority date

The '824 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 '824 patent in this or another forum where the issue becomes relevant.

VIII. Level of ordinary skill in the art

The '824 patent "relates to data management," in the context of packet-based "communication between a remote server and a user's device." Ex.1001, 1:15, 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 '824 patent, including packet-based computer networking. *See* Ex.1001, 6:20-23. The '824 patent refers to a variety of computer components, networking terms, and protocol acronyms without explanation, indicating that a POSITA would be familiar with a variety of computer and internet networking topics such as the World Wide Web and TCP/IP. *See* Ex.1001, 9:1-9, 9:67, 5:47-7:39; Ex.1003, ¶20.

A POSITA would have had a bachelor's degree in computer science, computer 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 1-20, which are all claims of the '824 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 '824 patent.

B. Statutory grounds

Grounds	Claims	Basis
#1	17-20	35 U.S.C. § 103 over Burns and Yang
#2	1-16	35 U.S.C. § 103 over Burns, Yang, and Wittenberg

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. Patent Publication 2005/0078668 to Wittenberg et al. (Ex.1007, "**Wittenberg**") published April 14, 2005.

Burns, Yang, and Wittenberg 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).

C. Statement of Material Facts

1. A POSITA would have been familiar with the format of TCP and IP packet headers. Ex.1018, 1:59-60; Ex.1045, [0015]; Ex.1044, [0003].

2. Decrypting an encrypted communication requires access to a corresponding decryption key, which only the endpoints to the communication would typically have. 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.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.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.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.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'").

D. Ground 1

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 randomness, 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 ’824 patent are both related to the same field of dynamically managing network communications. Ex.1001, 1:15-17, 17:28-30; Ex.1005, 2:47-49, 5:54-6:8, 6:46-56; Ex.1003, ¶¶57-62. 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, 1:22-32, 2:35-50; 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.

Yang and the ’824 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, ¶¶63-67. 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, 1:15-35, 2:3-50; 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.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.

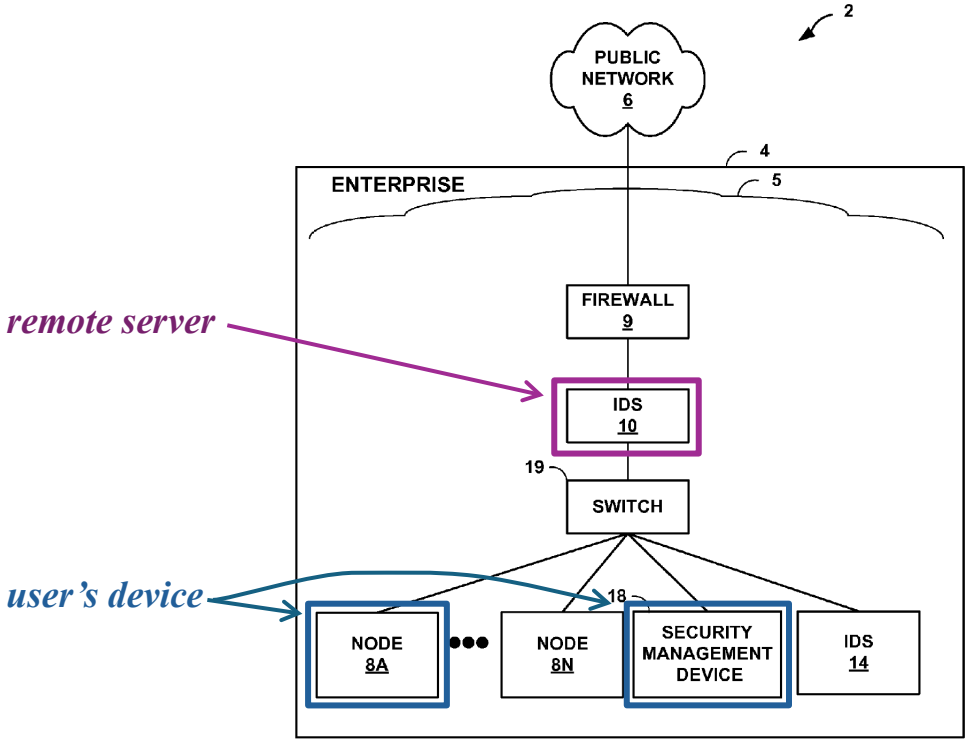
Burns specifically refers to Yang's disclosure of "techniques for identifying specific applications and protocols." Ex.1005, 4:39-40. A POSITA would have found Yang's techniques instructive regarding the operation of Burns's intrusion detection system, which "attempts to identify applications and protocols for each communication session." Ex.1005, 4:36-39; Ex.1003, ¶¶68-69.

4. Claim 17

[17.0] A system for dynamic management of private data during communication between a remote server and at least one user's device, the system comprising:

Burns describes "*system*" embodiments, Ex.1005, 1:6-8. For example, Figure 1 (below) shows an "exemplary system" including an intrusion detection system (IDS). Ex.1005, 4:20-25.

Burns describes dynamically managing private data, such as encrypted data. For example, Burns describes the IDS "blocking unidentified encrypted communication sessions." Ex.1005, Abstract; *see also id.*, 11:30-38. 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, ¶70.



Ex.1003, ¶¶71-73; 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, ¶72. 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, ¶72. 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*." 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, ¶¶73-74. 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 the IDS. Ex.1003, ¶74.

[17.1] 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, ¶75.

[17.2] a communication data type database, comprising at least one communication data type corresponding to sharing of at least one data packet

from the user's device;

Burns's IDS analyzes packets to determine an application or protocol for each communication session:

> 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.

Burns refers to Yang for "[e]xemplary techniques for identifying specific applications and protocols." 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 various "*communication data type[s]*." Ex.1003, ¶¶76-77. For example, 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, ¶78; Ex.1016, 709, Table 43-1; Ex.1033.

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"). It would have been obvious for the data types in Yang's static port mapping to be "*at least one communication data type corresponding to sharing of at least one data packet from the user's device*." Ex.1003, ¶79.

In summary, Yang's static port mapping table renders obvious a "*communication data type database*." Ex.1003, ¶¶77, 80-81.

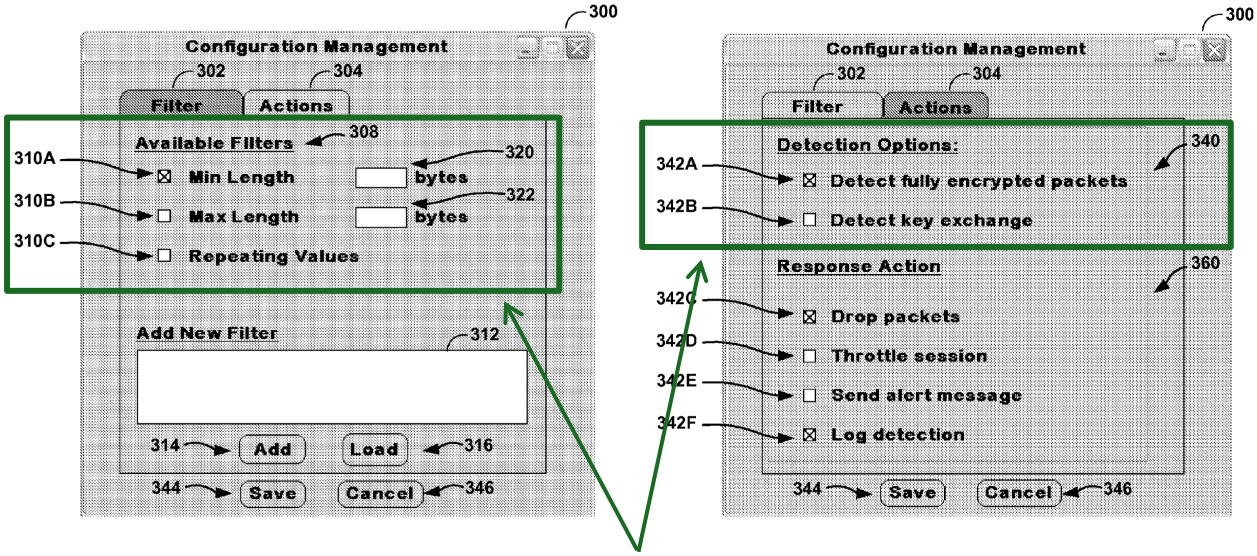
[17.3] a privacy preference database,

Burns's IDS is configurable, allowing its operation to be customized to match the privacy preferences of an administrator. Ex.1003, ¶¶82-83; Ex.1005, 1:21-35. For example, the administrator may specify patterns, attack definitions, and desired responses to be applied in the context of encrypted communications:

> 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.

Burns also illustrates in Figs. 7A-7B (below) various encryption-related 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 *privacy preference* items

Ex.1003, ¶¶83-84; Ex.1005, Figs. 7A-7B

A POSITA would have appreciated that Burns's configuration options pertaining to encryption are "*privacy preference[s]*" because encrypted communications are implicitly private. Ex.1003, ¶84. Thus, Burns's administrator-provided configuration information corresponds to "*preference preference[s]*."

It would have been obvious for Burns's IDS to store its configuration information in a "*database*": Burns explains that the administrator transmits configuration information to the IDS (specifically, to a "security management

module 44") by clicking the "Save" button shown in Figs. 7A-7B. *See, e.g.*, Ex.1005, 19:7-17; *see also id.* 9:64-67, 19:22-33, 6:37-59. 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 in a database by security management module 44 to facilitate instructing other IDS components to operate in accordance with the administrator-provided configuration information. Ex.1003, ¶84.

[17.4] comprising a list of allowed types of data packets for sharing during communication with the at least one user's device;

As discussed at [17.3], 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.

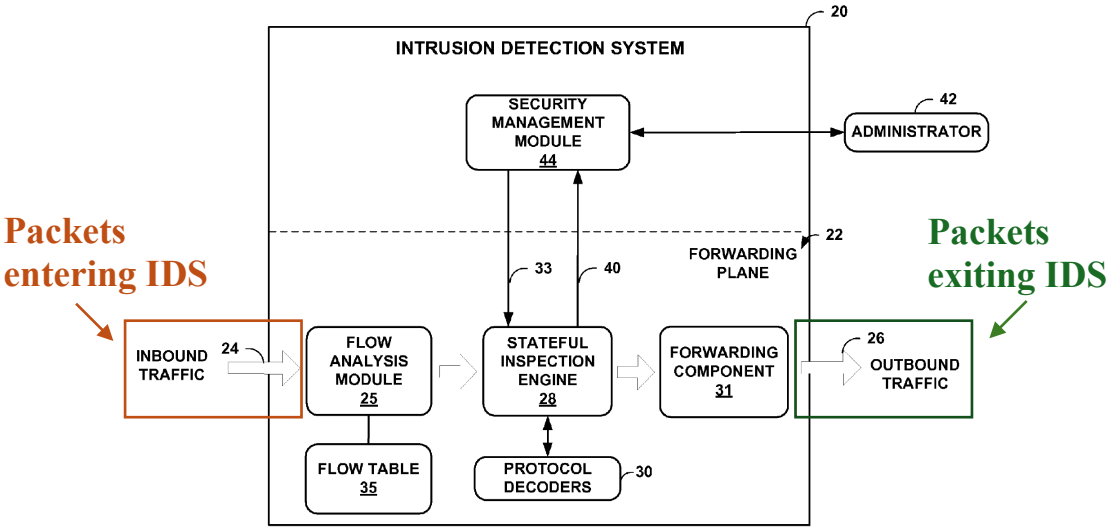
> [T]he user interface may permit the administrator to **drop all fully encrypted packets**, log details about the communication session…, throttle down the communication session with fully encrypted packets to reduce the bandwidth usage of that communication session, or other actions.

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

The administrator-provided "specified list of permitted software applications" renders obvious the claimed "*list of allowed types of data packets for sharing during communication with the at least one user's device*." Ex.1003, ¶¶85-86. Similarly, Burns's disclosure of dropping "all fully encrypted packets" renders obvious permitting non-encrypted packets. Ex.1003, ¶¶85-86.

[17.5] a communication module, 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, ¶87. An IDS network interface corresponds to the claimed "*communication module*."



Ex.1003, ¶¶87-88; Ex.1005, Fig. 1

As discussed at [17.0], 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, ¶88.

[17.6] a processor, coupled to a response database and to the privacy preference 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

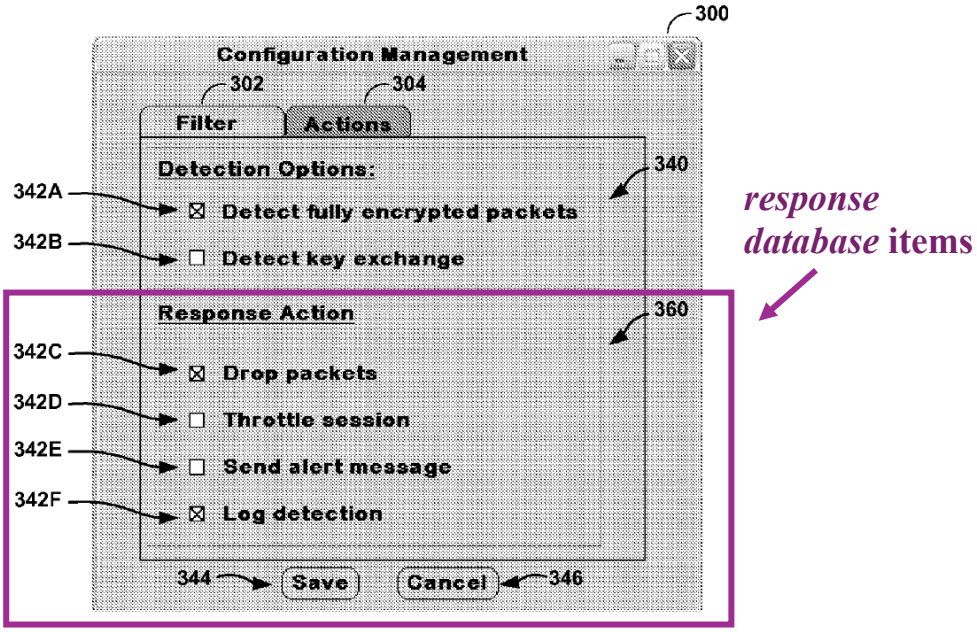
circuit (ASIC) or **a general-purpose processor**.

Ex.1005, 21:14-18.

As discussed at [17.3], 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 privacy preference database*) so that the processor would operate as the administrator had configured the IDS. Ex.1003, ¶89.

Burns's IDS configuration options also allow the administrator to select from a range of "response action[s]." Ex.1005, 19:61-66, Fig.7B (below). The suite of available response actions corresponds to the claimed "*response database*." Ex.1003, ¶91. Alternatively, the executable code stored within the IDS for performing the response actions corresponds to the claimed "*response database*." *See* Ex.1005, 3:17, 21:20-31. Ex.1003, ¶91.

Since the selection of response actions is also part of the IDS's configuration, it would have been obvious to a POSITA for Burns's IDS's processor to be *coupled to* the administrator-selected response actions and their corresponding executable instructions. Ex.1003, ¶90.



Ex.1005, ¶¶71-72; Ex.1005, Fig.7B (annotated).

In summary, the configurability of Burns's IDS's processor renders obvious this limitation.

[17.7] wherein the processor is configured to instruct the remote server

As discussed at [17.6], Burns's IDS includes a "*processor*." As discussed at [17.0], 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, ¶¶92-93; Ex.1041, [0025]; Ex.1042, 1:17; Ex.1043, [0031] .

Mapping the claimed "*processor*" to a component within the "*remote server*" is consistent with the '824 specification. *See* Ex.1001, 13:36-37

("processor 402 may be embedded in server 401.").

[17.8] to determine at least one data type for sharing of data packet that is compatible with the list of allowed patterns of data packets for sharing, and

There is no antecedent basis for "*the list of allowed patterns of data packets for sharing*." For the purposes of this IPR, Petitioner shows that this limitation would have been obvious interpreting this language as either (a) introducing "*a list of allowed patterns…*" or (b) referring to the "*list of allowed **types** of data packets for sharing*" introduced in [17.4]. Ex.1003, ¶94.

"*to determine at least one data type for sharing of data packet…*"

As initially discussed at [17.2], Burns and Yang describe the IDS analyzing a received packet to identify a corresponding application or protocol:

> 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.

> Application identification module 51 attempts to identify an application associated with the received packet (76). For example, application identification module 51 may inspect the packet header to determine the destination port of the packet. In some cases, **the destination port is associated with the protocol or application** being used for the communication session.

Ex.1005, 10:32-38.

> [A]pplication identification module 51 may use the well-known static port binding as a default application selection. Table 1, below, shows an example static port binding list.

Ex.1006, 9:64-66; *see also id.* 10:5-24.

Consistent with [17.2]-[17.4], an application or protocol associated with a packet is a "*data type for sharing of data packet*." Ex.1003, ¶95. Thus, the prior art's discussion of identifying an application or protocol renders obvious "*determining at least one data type for sharing of data packet.*" Ex.1003, ¶¶95-99.

(a) "…*compatible with ~~the~~ a list of allowed patterns*…"

The '824 patent does not explain what it means for a *data type* to be "*compatible*" with a *pattern* or a *list of allowed patterns*. As discussed below, however, Burns describes its IDS making decisions about packet handling based on combination of a *data type* and communication patterns. By allowing certain traffic based on both its *data type* and its *pattern*, Burns renders obvious determining that a *data type* is *compatible with a list of allowed patterns*. Ex.1003, ¶99.

The prior art describes three different examples of a communication *pattern*, each of which is sufficient to render this limitation obvious.

(1) Size: The IDS' encryption detection module determines the size of the packet's TCP/IP payload:

> Encryption detection module 58 may then determine the size N of the TCP/IP payload portion or, in another embodiment, just a portion of the TCP/IP payload of the received packet that typically corresponds to an application-layer header (142). In general, the size of the packet's TCP/IP payload refers to the number of bytes in the packet to be analyzed.

Ex.1005, 14:42-47.

Burns describes that size and frequency of a payload could be used to determine whether a packet is encrypted. Since Burns's IDS may permit non-encrypted packets to pass through (Ex.1005, 11:19-25), it would have been obvious to a POSITA that size is an "*allowed pattern[]*. Ex.1003, ¶103.

Furthermore, it would have been obvious to a POSITA that a "configurable input" as in Burns could be configured to forward packets of a certain size.

(2) 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.

A repeating sequence is a "*pattern*." Ex.1003, ¶111; Ex.1001, 9:65-10:6. Burns explains that the presence of a repeating sequence—whether within a single packet or across a series of packets—is indicative of a communication session that is not encrypted. Ex.1005, 14:32-38. Since Burns describes the IDS allowing non-encrypted packets to pass through (Ex.1005, 11:19-25), it would have been obvious to a POSITA that a repeating sequence is an "*allowed pattern[]*. Ex.1003, ¶111.

Yang identifies a variety of applications and protocols that were well-known not to be encrypted. Ex.1006, 10:5-24 (listing FTP, Telnet, Gopher, and HTTP); Ex.1003, ¶112. It would have been obvious to a POSITA that these unencrypted protocols would be "*compatible*" with having a repeating sequence (an "*allowed pattern[]*") in a packet or across a series of packets. Ex.1003, ¶¶111-12.

Thus, by identifying an application or protocol for a packet ("*determine at least one data type for sharing of data packet*") and detecting a repeating sequence and determining that a packet is not encrypted ("*compatible with ~~the~~ a list of*

allowed patterns of data packets for sharing"), Burns renders obvious this limitation.

(3) Key exchanges: Burns describes the IDS analyzing packets to detect 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.

Ex.1005, 19:25-36; *see also id.* 12:47-13:8.

Detecting that a series of packets represents a key exchange renders obvious determining an "*allowed pattern[]*." Ex.1003, ¶106.

As one example of a "*data type*" that is "*compatible*" with a key exchange as an "*allowed pattern[]*," Burns and Yang describe categorizing a packet as pertaining to an "unknown" application or protocol. Ex.1005, 8:66-67, 18:43; Ex.1006, 8:14-18; Ex.1001, 9:13-17, 9:25-30 ("unknown data type content").

Burns describes allowing a communication session to continue for an unknown application that performed a key exchange:

> In one embodiment, where IDS 10 determines that a communication session, for which an **application cannot be identified**, is encrypted, IDS 10 may further determine whether a key exchange associated with the communication session can be identified; **where a key exchange has been identified, IDS 10 may permit the communication session** and where a key exchange has not been identified, IDS 10 may terminate the communication session.

Ex.1005, 5:64-6:4.

Thus, Burns renders obvious for the "unknown" application data type to be "compatible" with *allowed pattern* of conducting a key exchange. Ex.1003, ¶108.

Burns and Yang also refer to other well-known protocols that employ encrypted packets. Ex.1005, 12:47-13:8; Ex.1006, 10:5-24; Ex.1003, ¶107. It would have been obvious for Burns's IDS to identify a packet as using a known encrypted protocol like SSH or HTTPs, and for such protocols to be "*compatible*" with conducting a key exchange (an "*allowed pattern[]*"). Ex.1003, ¶¶106-07. In this additional way, prior art renders this limitation obvious. Ex.1003, ¶¶99-114.

(b) "*…compatible with the list of allowed ~~patterns~~ types…*"

As discussed at [17.3]-[17.4], the list of applications or protocols permitted

to pass through Burns's IDS are a "*a list of allowed types of data packets for sharing*." It would have been obvious for at least some of the packets encountered by Burns's IDS would be for a permitted application or protocol, and thus, "*compatible with the list*." Ex.1003, ¶95. For example, Burns describes handling "HTTP traffic" and "SSH traffic." Ex.1005, 10:38-40. Both HTTP and SSH are *data type[s]* on Yang's static port mapping table. Ex.1006, 10:5-24; *see supra*, [17.4]. Burns also describes allowing all packets whose corresponding application or protocol is identifiable. Ex.1005, 5:29-34. Alternatively, Burns's IDS could determine if a packet was plain text or encrypted, which a POSITA would have understood to correspond to a "*data type*." *See* Ex.1003, ¶97; Ex.1037 [0024]; Ex.1038 [0010]; Ex.1039 [0031].

In summary, Burns and Yang describe identifying and allowing traffic associated with some or all identifiable protocols and applications, which renders obvious determining that a packet has a *data type* "*compatible with the list of allowed ~~patterns~~ types*." Ex.1003, ¶¶95-98.

[17.9] wherein the at least one data type is determined in accordance with characteristics of the communication data packet, and

As discussed at [17.2] and [17.8], Burns and Yang describe analyzing packets to determine the packets' corresponding application or protocol ("*determine at least one data type for sharing*").

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*"). 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 (f10); Ex.1003, ¶115.

Since Burns's IDS identifies an application or protocol ("*at least one data type*") based on packet header values ("*characteristics of the communication data packet*"), Burns and Yang render obvious this limitation. Ex.1003, ¶115.

[17.10] wherein the content of the at least one data packet is not read by the remote server for continued operation by the user's device in real time during communication between the remote server and the user's device.

As discussed at [17.0], Burns explains that at least some of the packets handled by the IDS ("*the remote server*") would be encrypted. It was well-known in the art that network devices such as Burns's IDS 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 IDS not to decrypt packets passing through it. Ex.1003, ¶¶116-17.

Additionally, Yang's technique of identifying an application or protocol is based on TCP and IP header values. *See supra*, [17.2], [17.8]. TCP and IP header values would not have been encrypted. Ex.1003, ¶117. Using TCP and IP header values is consistent with this negative limitation because the '824 patent and claims treat data packer header values as "*characteristics*," not "*content*." *See infra*, [5.0]; Ex.1001, 8:66-9:1 (identifying a data packet "header" as a "characteristic[]").

Thus, Burns and Yang renders obvious identifying a packet's application or protocol where "*the content of the at least one data packet is not read by the remote server*." Ex.1003, ¶¶116-17.

"for continued operation by the user's device in real time…"

It would have been obvious for the IDS to allow *continued operation by the user's device in real time* for multiple reasons.

First, 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 allow *continued operation by the user's device in real time* because delaying traffic would interfere with communication sessions. Ex.1003, ¶¶118-21. In other words, for the IDS to operate "transparently," it would need to make content determinations "*in real time*." Ex.1003, ¶120.

Second, 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. A POSITA would have found it obvious for the IDS's forwarding of packet traffic to facilitate *continued operation by the user's device*. Ex.1003, ¶120. Thus, the combination renders obvious for Burns's IDS to analyze and forward traffic "*for continued operation by the user's device in real time*." Ex.1003, ¶120.

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, ¶121; 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, ¶118; *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).

In summary, Burns and Yang describe an IDS that performs real-time analysis of communications passing through the IDS, which renders obvious making packet determinations *for continued operation by the user's device in real time*. Ex.1003, ¶¶118-21.

"…during communication between the remote server and the user's device…"

As discussed at [17.0], Burns's IDS monitors traffic flowing between enterprise computers such as node 8 or security management device 18 ("*user's computer*") and a public network. Ex.1005, Fig. 1. A POSITA would have found it obvious for the packets analyzed by the IDS to pass through the IDS "*during communication.*" Ex.1003, ¶122. Since the IDS is on the data path of packet traffic, it would have been obvious for the data traffic to also pass "*between*" node 8 (or security management device 18) and the IDS ("*remote server*"). Ex.1003, ¶122.

5. Claim 18

[18.0] The system of claim 17, further comprising

[18.1] a communication data pattern database, coupled to the processor and

As discussed at [17.6], a processor in Burns's IDS corresponds to "*the processor*." Burns renders obvious this limitation via three independent descriptions of "*communication data pattern[s]*," discussed below. As discussed at [17.3], 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 Burns's IDS to store *communication data pattern[s]* in a *database.* Ex.1003, ¶124. It would have been obvious for the IDS's processor to be *coupled* to such a *database* to facilitate storing *communication data pattern[s]* as configured by an administrator, and to facilitate retrieving *communication data pattern[s]* to be applied to packet traffic. *See, e.g.*, Ex.1005, 16:43-45, 18:49-53; 4:20-25; Ex.1003, ¶¶125-30.

Any one of the four *communication data pattern* disclosures discussed below is sufficient to render obvious a *communication data pattern*. Consistent with limitation [18.2], the claimed *communication data pattern database* is satisfied by a single *communication data pattern*.

(1) Packet size: As discussed at [17.8], Burns describes identifying size and frequency of a packet, which renders obvious a "*communication data pattern*."

Ex.1026, ¶126; Ex.1005, 14:42-47.

(2) Repeated data: As discussed at [17.8], Burns describes identifying a "repeating sequence" in one or more packets of a communication session. Ex.1005, 14:32-35 & 18:30-44. A repeating sequence is a "*communication data pattern*." Ex.1003, ¶128; *see* Ex.1001, 9:65-10:6.

(3) Key exchanges: As discussed at [17.8], Burns describes analyzing packets to detect whether they represent an encryption key exchange. Ex.1005, 19:25-38; *see also id.* 12:47-13:8. A series of packets represents a key exchange renders obvious a "*communication data pattern*." Ex.1003, ¶127.

A POSITA would have been familiar with the prior art protocols discussed in Burns and Yang, and a POSITA would have recognized that different protocols conduct their encryption key exchanges differently. Ex.1003, ¶127; *see, e.g.*, Ex.1005, 12:47-13:8 (discussing SSL key exchange involving ClientHello, ClientMasterKey, and ServerVerify messages); Ex.1046 17 & 21 (discussing SSH key exchange involving SSH_MSG_KEXINIT and SSH_MSG_NEWKEYS messages). It would have been obvious to a POSITA for Burns's IDS to store information about multiple detectable key exchanges in a *communication data pattern database*. Ex.1003, ¶127.

(4) Patterns and anomalies tables: Burns further 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.

Fig. 3 shows that the "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, ¶129. 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 "*communication data pattern[s]*" as claimed. Ex.1003, ¶129.

[18.2] comprising at least one data pattern corresponding to sharing of at least one data packet from the user's device,

As discussed at [18.1], the prior art renders obvious a *communication data*

pattern in multiple ways. Each of the four independent *pattern* disclosures discussed at [18.1] relate to packets analyzed by the IDS ("*corresponding to sharing of at least one data packet from the user's device*"). Ex.1003, ¶131.

[18.3] wherein the processor is configured to modify data packets corresponding to requests for retrieval of data packets and communication data types that are not compatible with communication data patterns from a communication data pattern database.

This limitation confusingly re-introduces *communication data patterns* (similar to "*at least one data pattern*" in [18.2]) and *a communication data pattern database* (also recited in [18.1]). For purposes of this IPR, Petitioner interprets these aspects of [18.3] as referring back to the previously introduced concepts in [18.1] and [18.2].

Burns's IDS is configurable to take various actions in handling a packet based in part on the packet's application or protocol ("*data packets corresponding to…communication data types*"). 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.

Any one of these three actions shows that Burns's IDS processor is "*configured to modify data packets*," as discussed further below. Ex.1003, ¶132. Burns also describes analyzing packets flowing in both directions between a client and server as a single "communication session." Ex.1005, 7:8-15. Thus, it would have been obvious to a POSITA for Burns's IDS to analyze, and potentially modify, packets from either the client or server that request a response ("*data packets corresponding to requests for retrieval of data packets*"). Ex.1003, ¶134.

(1) Dropping a packet: The '824 patent expressly contemplates that "blocking" a packet is a form of "modification." Ex.1001, 18:32-37 (claim 8). By blocking a packet, it would have been obvious that Burns's IDS processor is "*configured to modify data packets*." Ex.1003, ¶¶137-38.

(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.1005, 7:20-22, 9:16-25, 9:32-33. It would have been obvious for the IDS to close a TCP communication session by modifying a packet header. Material Fact #6; Ex.1003, ¶139. By closing a TCP communication

session, it would have been obvious that Burns's IDS processor is "*configured to modify data packets*." Ex.1003, ¶139.

(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 the IDS to throttle a TCP session by modifying a packet header. By throttling a TCP communication session, it would have been obvious that Burns's IDS processor is "*configured to modify data packets*." Ex.1003, ¶¶140-41.

"…that are not compatible with communication data patterns from a communication data pattern database"

As discussed above, Burns's IDS provides network security by inhibiting communications (e.g., blocking, closing, or throttling sessions). It would have been obvious to a POSITA for Burns's IDS to take such actions only in response to network threats or other activities that the administrator has indicated are to be discouraged. The administrator's preferences regarding such activity restrictions are expressed, in part, through the filters and rules configured on the IDS. *See supra,* [18.1]. For example, Burns contemplates that encrypted data packets of an unknown application are not "*compatible*" with the pattern of not conducting an encryption key exchange. Ex.1005, 7:23-44. Thus, it would have been obvious for

the IDS processor to be configured to block, close, or throttle "*data packets and communication data types that are not compatible with communication data patterns from a communication data pattern database*." Ex.1003, ¶¶133-41.

6. Claim 19

[19.0] The system of claim 18, wherein data packets from the user's device are selected from the group consisting of user's device files, user's device characteristics, user's device indirect attributes, user's device sensor data, user's device browser data, user's device form data, user's device dynamic memory and user's device static memory.

This claim recites a Markush group that is satisfied by showing any of the listed items in the prior art.

Burns and Yang both discuss analyzing "HTTP traffic," which was well-known to be used for communicating "web-page" data to and from a web browser. Ex.1005, 9:15, 10:39; Ex.1006, 5:9-12, 5:39-42; Ex.1003, ¶144; Ex.1047; Ex.1048. Thus, it would have been obvious for *data packets* analyzed by Burns's IDS to include "*user's device browser data*." Ex.1003, ¶¶143-44.

Yang also describes creating a compound attack definition (a filter) that applies only to "HTTP form data." Ex.1006, 15:35-41. Thus, it would have been obvious for the *data packets* analyzed by Burns's IDS to include "*user's device form data*." Ex.1003, ¶145.

7. Claim 20

[20.0] The system of claim 18, wherein at least the communication module and the processor are embedded on a single hardware component.

As discussed at [17.5], it would have been obvious for the claimed "*communication module*" to be a network interface of Burns's IDS.

As discussed at [17.6], it would have been obvious for the claimed "*processor*" to be the processor of Burns's IDS.

Burns contemplates implementing its IDS technologies "in hardware." Ex.1005, 21:14. A POSITA would have been familiar with IDS devices, including that they are often implemented using hardware. Thus, it would have been obvious for Burns's overall IDS device to be a "*single hardware component*" incorporating both a processor and a network interface. Ex.1003, ¶146.

E. Ground 2

1. Wittenberg (Ground 2)

Wittenberg describes a network security device with a "a built-in HTTP server." Ex.1007, [0016]. The server provides a webpage through which a user logs into the network. Ex.1007, [0046]. By having a user log in, the security device is able to customize its operation for each user. Ex.1007, [0017], [0004], [0016].

Wittenberg and the '824 patent are both related to the same field of dynamically managing network communications. Ex.1003, ¶¶147-51; Ex.1001, 1:15-17, 17:28-30; Ex.1007, [0001], [0016]-[0017], [0019], [0053]. And like the '824 patent, Wittenberg contemplates providing access to "semi-private" resources. Ex.1007, [0018]; Ex.1001, 1:59-63 (discussing "partial" blocking data collection).

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

As discussed regarding Ground 1, Burns and Yang describe a configurable IDS, where the configuration may be specified by "a user, such as a system administrator." Ex.1005, 6:5-8. It would have been obvious to a POSITA for the IDS to require a user to authenticate themselves as an administrator using a well-known login procedure, such as providing a username and password. Ex.1003, ¶152. Requiring authentication to access or modify the IDS's configuration would have ensured that only an authorized administrator, and not merely any user, would be able to make configuration changes. A POSITA would have appreciated the importance of limiting access to the IDS's configuration, since an improper change to the IDS configuration could compromise network security (e.g., allowing malicious traffic) or inhibit legitimate network communications (e.g., blocking non-malicious traffic). Ex.1003, ¶¶152-53.

Wittenberg describes a typical and well-known login process that would have been suitable for authenticating a user as an administrator to a network device like Burns's IDS. Ex.1007, [0046], Fig. 8. Implementing in Burns's IDS a process to authenticate a user as an administrator using Wittenberg's webpage-based login process would have been an obvious approach because configuring network devices via a web-based interface was common. *See, e.g.*, Ex.1049, claim 15 ("web-based visualization interface that facilitates configuration of the system and

forensic analysis of captured attack information by administrators"); [0047] ("secure configuration and administration may be provided…through an HTTPS…Web Browser."); Ex.1050, 6:39-40 ("configuration tables accessible through web client interface"); Ex.1051, [0099] ("System changes are typically accomplished by authenticated administrators that access system 100" through the web); Ex.1052, [0004] (using a browser, "a system administrator can browse to the address of a particular device. The embedded web server returns a web page allowing the administrator to select configuration settings for that device").

The combination also would have been obvious because it is merely the use of a known technique (a login webpage) to improve a similar device (Burns's IDS) in the same way (confirming the identity of a user by collecting the user's username and password). Ex.1003, ¶¶152-55.

The combination also would have been obvious because it represents the application of a known technique (user authentication) to a known device (Burns's IDS) to yield predictable results (limiting access to the IDS's configuration to an authorized administrator). Ex.1003, ¶155.

3. Claim 1

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

As discussed at [17.0], Burns and Yang render obvious the "*dynamic management of private data during communication between a remote server and a*

user's device." Ex.1003, ¶¶157-58. Burns and Yang also disclose implementing such technology in the form of a "*method*." Ex.1005, 21:14-16, 21:36; Ex.1006, 2:56, 2:24-43. Thus, to the extent that the preamble is limiting, it would have been obvious over Burns and Yang. Ex.1003, ¶¶159-63.

[1.1] receiving, by the user's device, a request for retrieval of at least one data packet from the user's device,

As discussed at [19.0], Burns and Yang discuss analyzing "HTTP traffic" and "HTTP form data," which a POSITA would have known are associated with web pages. Ex.1005, 10:39; Ex.1006, 15:35-41; Ex.1003, ¶165. For example, some web pages invite a user to provide an input—like providing a username or password on a login page—which is typically provided back to a web server as HTTP form data. Ex.1003, ¶167.

Wittenberg illustrates a "known subscriber login page" in Figure 8. The login page requests the user provide a username and password. Wittenberg also acknowledges that "[a]dditional information may be collected" using other, similar "pop-up windows." Ex.1007, [0046]. Thus, it would have been obvious for a user's device to receive and display an information-requesting web page (like the login page of Fig. 8), which is "*request for retrieval of at least one data packet from the user's device*." Ex.1003, ¶¶164, 166-67.

[1.2] wherein the user's device is configured to provide a response corresponding to the received request;

Wittenberg explains that the web page shown in Fig. 8 allows a user "to login." Ex.1007, [0033], [0053]. It would have been obvious for the user's device to allow the user to provide their username and password, then click the "LOGIN" button. Ex.1003, ¶168. When user clicks "LOGIN," the web browser would create and send an HTTP RESPONSE message including the provided username and password as HTTP form data. Ex.1003, ¶168. Because this functionality occurs in ordinary course of how known web browsers operated in the prior art, Wittenberg's login page shown in Fig. 8 renders obvious this limitation. Ex.1003, ¶168.

[1.3] determining, by the remote server, at least one communication data type of the at least one data packet corresponding to the received request,

See [17.8] above. Ex.1003, ¶169. There is no substantive difference between a "*data type*" as recited in claim 17 and a "*communication data type*" recited in claim 1.

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

The claim term "*the at least one data packet corresponding to the received request*" is understood to refer to the "*at least one data packet*" recited in [1.1], which is the "*response corresponding to the received request*" recited in [1.2]. Ex.1003, ¶¶170-71.

As discussed at [17.8], Burns and Yang render obvious for Burns's IDS to "*determine at least one data type for sharing of data packet*." The same analysis renders obvious for the IDS to determine "*at least one communication data type*."

Ex.1003, ¶172. For example, it would have been obvious to a POSITA for Yang's exemplary applications and protocols to be associated with particular "*communication data type[s]*." Ex.1003, ¶173.

Yang's list of exemplary applications and protocols includes HTTP, referring to Hyper-Text Transfer Protocol, which was a well-known protocol associated with communicating web page data. As discussed at [1.1], it would have been obvious for a user's device to receive a web page seeking the user's input, such as a login page (the "*received request*"). Since is login webpage, like Wittenberg's Figure 8, is a webpage, it would have been obvious for such a login webpage to be communicated to the user's device using HTTP, and thus, for Burns's IDS to identify the packets containing the login webpage as using the HTTP application or protocol. Ex.1003, ¶174.

Since Burns's IDS ("*remote server*") identifies an application or protocol ("*communication data type*") for the packets passing through it, Burns and Yang render obvious "*determining, by the remote server, at least one communication data type of the at least one data packet corresponding to the received request*." Ex.1003, ¶¶175-76.

[1.4] wherein the at least one communication data type is determined in accordance with characteristics of the communication data packet, and

See [17.9] above. Ex.1003, ¶177.

[1.5] wherein the content of the at least one data packet is not read by the remote

server for continued operation by the user's device in real time during communication between the remote server and the user's device;

See [17.10] above. Ex.1003, ¶¶178-79.

[1.6] receiving, by the user's device, a privacy preference for the user's device,

As discussed at [17.3], Burns describes how the IDS's handling of packets, including potentially encrypted packets, is configurable by an administrative user. Burns illustrates, for example, in Figs. 7A-7B a user interface through which the administrative user provides IDS configuration information. As discussed at [17.0], it would have been obvious for the administrative user to configure the IDS using node 8A or security management device 18, either of which corresponds to "*the user's device*." Since Burns's IDS protects the computers within enterprise network 4, the configuration information for the IDS (including how to handle potentially encrypted packets) renders obvious "*a privacy preference for the user's device.*" Ex.1003, ¶¶180-83.

In summary, Burns's discussion of an administrator or user providing configuration information to the IDS via node 8A or security management device 18 renders obvious "*receiving, by the user's device, a privacy preference for the user's device*." Ex.1003, ¶183.

[1.7] wherein the privacy preference comprises a list of allowed data packet communication types for sharing during communication;

See [17.4] above. Ex.1003, ¶¶184-85.

[1.8] modifying data packets corresponding to requests for sharing of responses that are not compatible with the received privacy preference; and

As discussed at [18.3], Burns's IDS provides network security by inhibiting communications (e.g., blocking, closing, or throttling sessions), and each of these security actions involve *modifying data packets*. In doing this, the IDS monitors network traffic flowing in both directions between a client and server. Ex.1005, 7:8-15. Thus, it would have been obvious for Burns's IDS to analyze and modify packets containing a webpage form (such as a login webpage) sent to a user's device ("*data packets corresponding to requests for sharing of responses*"). As discussed at [17.3]-[17.4] and [1.6], Burns's IDS operates in accordance with configuration information provided by the administrative user ("*the received privacy preference*"). Ex.1003, ¶¶186-87. Thus, it would have been obvious for Burns's IDS to modify such packets to the extent that they are disallowed by the administrative user. Ex.1003, ¶¶188-90.

[1.9] maintaining communication between the remote server and the user's device, with sharing of the modified data packets.

As discussed at [17.10], Burns's IDS operates "transparently" and forwards selectively modified network traffic. Ex.1005, 6:29-33. In operating transparently, it would have been obvious to a POSITA that the IDS is "*maintaining communication*" by "*sharing of the modified data packets*." Ex.1003, ¶¶191-94.

4. Claim 2

[2.0] The method of claim 1, wherein the communication data type of the at least one data packet is determined from metadata of communication with the remote server.

As discussed at [1.3], Burns describes identifying a packet's *communication data type* based in part on a TCP port value. The TCP port value in a packet header is "*metadata*." Ex.1003, ¶195; Material Fact #5.

5. Claim 3

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

[3.1] determining, by the user's device, at least one communication data pattern of at least one data packet corresponding to the received request; and

As discussed at [17.8] and [18.1]-[18.3], Burns and Yang render obvious determining a "*communication data pattern*" for a packet in multiple ways. Ex.1003, ¶197.

Burns and Yang contemplate identifying *patterns* in packets flowing through the IDS, which Burns illustrates as a stand-alone device within an enterprise network. Those of skill in the art would have recognized this as a network-based IDS. Ex.1003, ¶¶198-200. Those of skill in the art would have also been familiar with the other commonly used IDS structure, the host-based IDS. Ex.1036. Combining features of host-based and network-based IDS designs was also known in the prior art:

> IDS' normally fall into a number classifications. These

> classifications include network-based, host-based, protocol-based, and application-based intrusion detection systems. Combinations of these classifications are common. These combinations, also known as hybrid intrusion detection systems, including, for example, a **combination of network-based and host-based intrusion detection systems**.

Ex.1053, [0005].

A POSITA would have found it obvious for Burns's user devices (e.g., node 8 and security management device 18) to perform their own analysis of data patterns in packets they send and receive. Having the user devices perform *pattern* analysis while Burns's IDS performs *data type* analysis merely represents a known integration of the host-based and network-based IDS types. Ex.1053, [0005]; Ex.1003, ¶201.

[3.2] modifying data packets corresponding to requests for sharing of responses corresponding to at least one data pattern, that are not compatible with the received privacy preference,

As discussed at [18.3], Burns's IDS is configurable to take various actions in handling a packet, including "*modifying*" a packet by (1) dropping (blocking) the packet, (2) closing the related communication session, or (3) throttling (limiting bandwidth consumption of) the related communication session.

As discussed at [18.3], Burns and Yang render obvious for the IDS to modify packets that are "*not compatible with not with communication data patterns from a*

communication data pattern database." As discussed at [18.1], the patterns applied by the IDS are configurable by the administrator, and as discussed at [1.6], the IDS configuration information corresponds to the "*privacy preference*." Thus, it would have been obvious for the patterns applied by the IDS to be part of *the received privacy preference*. Ex.1003, ¶¶202-04.

[3.3] wherein the privacy preference comprises a list of allowed data patterns for sharing during communication.

As discussed at [17.3] and [1.6], an administrator provides configuration information to Burns's IDS, which corresponds to "*the privacy preference*." Ex.1003, ¶205.

As discussed at [17.8] and [18.1], Burns and Yang render obvious three independent descriptions of "*data patterns*." As discussed below, the disclosure of size, repeated data and key exchange correspond to *allowed data patterns*.

(1) Size: Burns describes that size of a payload could be used to determine whether a packet is encrypted. Since Burns describes the IDS allowing non-encrypted packets to pass through (Ex.1005, 11:19-25), it would have been obvious to a POSITA that size is an "*allowed pattern[]*. Ex.1003, ¶206. Further, a POSITA would have recognized that the administrator's configuration ("*the privacy preference*") in Burns could be configured to forward packets of a certain size.

(2) Repeated data: Burns illustrates in Fig. 5 that the IDS may be configured to make determinations based on multiple factors, including whether a

packet is encrypted based on a "*data pattern*" of containing repeated values. *See, e.g.,* Ex.1005, 5:29-53, 14:32-38. Thus, it would have been obvious for the configuration information to include preferences relating to whether a key exchange is observed. Ex.1003, ¶209.

(3) Key exchanges: Burns illustrates in Fig. 5 that the IDS may be configured to make determinations based on multiple factors, including whether a key exchange is observed (an "*allowed data pattern[]*").Thus, it would have been obvious for the configuration information to include preferences relating to whether a key exchange is observed. Ex.1003, ¶¶207-08; *see* Ex.1005, 5:29-53.

In summary, Burns describes configuring the IDS to block, terminate, or throttle communications ("*modifying data packets*" per [3.2]) where the packets are not "*compatible*" with "*allowed data patterns*" per the administrator's configuration ("*privacy preference*"), which renders this limitation obvious. Ex.1003, ¶¶210-12.

6. Claim 4

[4.0] The method of claim 3, wherein the communication data pattern is determined based on a range 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.

As discussed at [18.1]-[18.2], Burns describes identifying a repeating pattern, sequence, value, size, or frequency, 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 "*range*" for a "*communication data pattern*." Ex.1003, ¶¶213-15, 217; *see supra*, [3.1]-[3.3].

Burns also describes identifying a key exchange (*see supra*, [18.1] and [3.3]), which it would have been obvious to a POSITA requires that packets be exchanged in a "*data packet order*." Ex.1003, ¶¶216, 218; *see, e.g.,* Ex.1040, 34 (explaining the TLS Handshake Protocol).

7. Claim 5

[5.0] The method of claim 1, wherein the communication 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.

As discussed at [1.3], Burns and Yang render obvious determining a packet's *communication data type* by comparing a TCP port value to a list of known applications. The TCP port value is in the TCP header, which renders obvious determining a "*data type*" based on a "*data packet header*." Ex.1003, ¶220; *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, ¶¶221-25; *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). Thus, by identifying a packet as a TCP packet and identifying its application from a TCP port value, Burns and Yang render obvious determining a *communication data type* based on (plural) *data packet characteristics*. Ex.1003, ¶¶220-25.

8. Claim 6

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

[6.1] determining, by the remote server, a communication data pattern of the at least one data packet of the received request; and

As discussed at [17.8] and [18.1]-[18.3], Burns and Yang render obvious IDS determining a "*communication data pattern*" for a packet in multiple ways. Ex.1003, ¶¶227-30.

[6.2] linking at least one response from the user's device to a type of data packet from the user's device,

See [1.3] above. There is no meaningful difference between a "*at least one data packet corresponding to the received request*" in [1.3] and a "*response from the user's device*" here. There is no meaningful difference between a "*communication data type*" in [1.3] and a "*type of data packet*" here. *See* Ex.1001, 9:38-39, 8:26. By identifying a "*type of data packet*" for a packet, Burns's IDS implicitly "*link*[s]" the packet to that "*type of data packet*." Ex.1003, ¶¶231-32.

Petitioner notes that [6.2] and [6.3], taken together, recite linking a response (i.e., a packet) "*to a type of data packet…based on*" the packet's "*communication data type*." The '824 patent, however, appears to use the terms "*type of data packet*" and "*communication data type*" interchangeably. While the specification never uses the term "*communication data type*," it does repeatedly refer to the "data type" of a "data packet." *See, e.g.*, Ex.1001, 9:38-39 ("the data type 207 of a data packet"), 8:26. To the extent that the terms "*type of data packet*" and "*communication data type*" are understood to mean the same thing, limitations [6.2] and [6.3] appear to suggest an iterative, recursive process in which a packet's "*type*" is both an input and output of an identification process. Yang describes such an iterative *type*-identification process:

> For example, HTTP and FTP packet flows may share similar characteristics…. [U]pon receiving a packet flow that, upon initial classification, may be either HTTP or FTP, application identification module 51 may **first select HTTP as the type of application** based on the higher priority given to the type of protocol…..**Upon receiving a reply packet flow from the server**, application identification module 51 may examine the reply packet flow and determine that the communication session shares more properties with an FTP communication session. Application identification module 51 may then **re-classify the communication as an FTP communication session**.

Ex.1006, 10:60-11:10.

Yang's example shows that in processing the reply packet, the IDS's analysis would receive the currently identified protocol type (e.g., HTTP) and make a new protocol type identification (e.g., FTP). Ex.1003, ¶233.

Alternatively, to the extent Patent Owner argues that the claimed "type of data packet" is different form of classification that the "communication data type," Burns and Yang still render this limitation obvious by applying an "attack definition" to identify a packet that is "malicious." Ex.1005, 4:20-25; Ex.1006, 11:20-23, 14:12-21; *see also infra*, [6.3]. Malicious is a "*type of data packet*." Ex.1003, ¶234. By identifying a packet from the user's device as malicious, Burns and Yang render obvious "*linking*" that packet "*to a type of data packet*" as claimed. Ex.1003, ¶234.

[6.3] wherein the linking is based on the communication data type and communication data pattern of the at least one response.

As discussed at [6.2], Burns and Yang render obvious the claimed "*linking*" by identifying a packet's protocol or application based in part on a previously identified protocol or application ("*the communication data type*") for the packet's communication session. Ex.1003, ¶235.

Yang also describes how the initial identification of an application or protocol is, itself, based partly on pattern information:

> After identifying the beginning of the packet flow,

> application identification module 51 makes a preliminary determination of the type of application and protocol of the packet flow (81). This **preliminary determination may be based on the pattern of the received packet flow**, initial inspection of the payloads of the packets of the packet flow, the amount of data received in the packet flow or other characteristics.

Ex.1006, 12:17-23; *see id.*, 13:25-26 ("pattern matching to determine the type of application and underlying protocol").

> Application identification module 51 may analyze the pattern of the TCP data reassembled from packet flow to make the initial determination. For example, application identification module 51 may inspect the packet flow to find characteristics of known applications and the corresponding protocols, such as HTTP, FTP sendmail, SMTP, etc.

Ex.1006, 13:35-41; Ex.1003, ¶236.

Thus, by identifying an application or protocol based on an initial application or protocol determination, which, in turn, is based on a pattern, Burns and Yang render obvious identifying an application or protocol based on both an initially identified application or protocol ("*communication data type*") and pattern ("*communication data pattern*"). Ex.1003, ¶237.

Alternatively, as introduced at [6.2], Burns and Yang describe using an

"attack definition" to identify packets as malicious or not malicious ("*the linking*"). *See, e.g.*, Ex.1005, 9:67-10:31; Ex.1006, 15:63-64. Burns and Yang further describe the use of a "compound attack definition," explaining that a compound attack definition specifies both a "protocol" and "patterns" to be matched. Ex.1005, 6:44-45; Ex.1006, 15:29-34, 15:50-62, 12:55-67. Yang describes an example scenario of a compound attack definition that specifies protocol-specific pattern matching:

> For example, a system administrator may specify a compound network attack that includes the protocol anomaly of repeated FTP login failure and a pattern that matches a login username of "root."

Ex.1006, 5:55-59; *see also id.* 8:39-48 (identifying FTP as a protocol).

Thus, Burns and Yang render obvious identifying a packet as malicious ("*the linking*") based on both the packet's protocol ("*communication data type*") and a matched pattern ("*communication data pattern*"). Ex.1003, ¶¶238-40.

9. Claim 7

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

[7.1] identifying a response corresponding to a request for sharing of data packets,

As discussed at [1.8] and [18.3], Burns's IDS monitors network traffic flowing in both directions between a client and server, including "identify[ing] pairs of [unidirectional] packet flows that collectively form a single

communication session." Ex.1005, 7:8-15. By describing identifying packets flowing in opposite directions that relate to one another, Burns renders this limitation obvious. Ex.1003, ¶¶242-43.

[7.2] wherein the identified response is not compatible with the received privacy preference; and

As discussed at [1.8] and [18.3], Burns's IDS provides network security by inhibiting disallowed communications (e.g., blocking, closing, or throttling sessions in accordance with an administrator's configuration information). Burns and Yang an IDS performing its security functions on all packets, including response packets. Ex.1005, 10:25-26; Ex.1006, 8:3-7, 8:18-22, 13:10-12; *see also supra,* [6.1]-[6.2].

Thus, it would have been obvious to a POSITA for Burns's IDS to identify a "*response*" packet as being prohibited by the administrator's configuration information ("*received privacy preference*"). Ex.1003, ¶¶244-46.

[7.3] providing a response with at least one modified data packet corresponding to the request for sharing types of data packets.

As discussed at [18.3] and [1.8]-[1.9], Burns and Yang render obvious having the IDS modify a packet and then share the modified packet, which renders this limitation obvious. Ex.1003, ¶247.

10. Claim 8

[8.0] The method of claim 7, wherein the modification of the at least one modified data packet is selected from the group consisting of data nullification,

blocking, 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 [18.3], 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*." Ex.1005, 7:47-51; Ex.1003, ¶249; *see* Ex.1001, 2:44-50. Burns also describes "automatically closing the communication session" and "throttle[ing]," which would obviously involve a "change of header" for the reasons discussed at [18.3]. Ex.1005, 7:47-51, 10:6-9, 11:53-55; Ex.1003, ¶¶249-52; Ex.1016, 765, 805-06.

11. Claim 9

Claim 9 recites limitations very similar to those in claim 1. One difference between them is that where claim 1 recites in [1.3]-[1.4] "*determining…at least one communication data* ***type****…in accordance with* ***characteristics*** *of the communication data packet*," claim 9 recites in [9.3]-[9.4] "*determining…at least one communication data* ***pattern****…accordance with* ***a behavior range*** *of the communication data packet*." A later limitation is similarly refocused from "*data packet communication types*" to "*data patterns*." *Compare* [9.7] *to* [1.7].

As discussed above for claim 3, the prior art renders obvious determining a *communication data pattern*. *See supra*, [3.1]. As discussed at [17.8], the prior art

renders obvious identifying three different *allowed communication data patterns*, each of which is alone sufficient: size, repeated data and key exchanges. As discussed at [4.0], each of these is a "*range*." The '824 patent explains that these *range[s]* correspond to "dynamic" features, and thus, "behavior." Ex.1001, 9:61-10:2, 9:40-42.

The prior art also describes using the determined *communication data pattern* to determine whether to modify a packet, recited in [9.8]. *See supra*, [3.2].

Claim 9 differs slightly from claim 1 in its "*modifying*" step, with the new language indicated here in bold: "*modifying data packets corresponding to requests for sharing of responses* ***and corresponding data patterns*** *that are not compatible with the received privacy preference*." This language largely mirrors the language of step 614 in the '824 patent Figure 6B. In context, this step refers to modifying a packet that was selected based on a data pattern configuration. Ex.1001, 15:38-49 ("*modifying according to…forbidden data patterns*"). As discussed at [3.1]-[3.3], the prior art renders obvious *modifying data packets* based in part on the packets not matching *data patterns* configured to be *allowable* by an administrator. Ex.1003, ¶¶253-68.

In summary, claim 9 is rendered obvious by the prior art for the reasons discussed above regarding similar language in other claims. Ex.1003, ¶¶253-68.

12. Claim 10

[10.0] The method of claim 9, wherein the communication data pattern of the at least one response is determined from metadata of communication with the remote server.

Claim 10 is similar to claim 2, except that claim 10 refers to a "*communication data pattern*" instead of a "*communication data type*."

Burns describes a prior art technique of identifying potentially malicious traffic based on packet size and frequency ("*metadata*"):

> For example, a conventional IDS may, for example, attempt to apply behavior analysis to the overall communication session, such as by **determining an average size and frequency of data transmission** for a certain port or session. 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, 2:31-37.

From this disclosure, it would have been obvious for Burns's IDS to calculate an "average size and frequency" for packets in a communication session. These averages correspond to "*the communication data pattern*," and would be determined from the size and frequency of individual packets ("*metadata of communication*"). Ex.1003, ¶269.

It would have been obvious to a POSITA for Burns's IDS to perform this

metadata-based technique because the technique was known in the art to be useful for an IDS to identify a "malicious or unwanted application" that should be blocked. Ex.1005, 2:34-37; Ex.1003, ¶¶270-72 (metadata includes packet size, frequency, and key exchange); Ex.1054; Ex.1055; Ex.1056. Notably, while Burns notes that this behavior analysis technique "may" have some shortcomings in certain situations, Burns says nothing to discourage its use. Ex.1005, 2:37-43.

13. Claim 11

[11.0] The method of claim 9, further comprising:

[11.1] determining, by the remote server, the communication data type of the at least one data packet of the received request; and

There is no antecedent basis for "*the communication data type*" recited in [11.1]. For purposes of this IPR, Petitioner analyzes this claim as if it recited "*determining…**a** communication data type.*" Thus, this limitation is substantially identical to [1.3]. Ex.1003, ¶274.

[11.2] modifying data packets corresponding to requests for sharing of responses and corresponding data patterns, corresponding to at least one communication data packet type, that are not compatible with the received privacy preference,

See [1.8] and [3.2] above. Ex.1003, ¶275. As discussed at [17.8], the prior art renders obvious determining whether a packet is "*compatible*" with configuration information based on both the packet's determined *communication data pattern* and *data type*. Ex.1003, ¶¶276-77.

[11.3] wherein the privacy preference comprises a list of allowed data packet

types for sharing during communication.

See [1.7] above. Ex.1003, ¶278.

14. Claim 12

[12.0] The method of claim 11, wherein the communication 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.

See [5.0] above. Ex.1003, ¶279.

15. Claim 13

[13.0] The method of claim 9,

[13.1] wherein the user's device is in communication with an external database with at least one communication data pattern corresponding to a request for sharing of data packets from the user's device, and

As discussed at [17.3] and [18.1], Burns renders obvious storing configuration information, including *pattern[s]*, in a *database* located at the IDS. Since configuration information comes from an administrator using a computer ("*user's device*"), it would have been obvious for the administrator's computer to be "*in communication*" with the configuration database, which is "*external*" to the administrator's computer. Ex.1003, ¶¶281-82.

To the extent that Patent Owner argues that the claimed "*database*" is *external* to not only the "*user's device*," but also the "*remote server*," such an arrangement would have been obvious from the prior art. For example, Burns

discloses that a node in its Figure 1 enterprise network may represent a "database server[]." Ex.1005, 4:32-35. Since Burns contemplates the network including multiple centrally managed "IDSs 10 and 14" (Ex.1005, 6:10-17), it would have been obvious to a POSITA to centralize the storage of the IDSs' configuration information in a single database server. Ex.1003, ¶282.

[13.2] wherein at least one data pattern compatible with the received request is determined from the database.

As discussed at [17.8] and [18.3], the prior art renders obvious identifying a *data pattern* corresponding to a packet. Ex.1003, ¶283. It would have been obvious for the *data pattern* to be "*compatible*" with a packet received by the user's device ("*the received request*"), for example, because the pattern is allowed by the administrator's configuration ("*determined from the database*"). Ex.1003, ¶¶283-84; *see supra*, [17.8], [18.3], and [3.3]. For example, Burns describes allowing traffic where a key exchange *data pattern* has been identified. Ex.1005, 6:1-3.

16. Claim 14

[14.0] The method of claim 9, further comprising:

[14.1] determining, by the remote server, a type of the at least one data packet of the received request; and

See [1.3] above. Ex.1003, ¶286.

[14.2] linking at least one data pattern to a type of data packet from the user's device,

As discussed at [17.8], the prior art renders obvious analyzing a packet

based on both the packet's determined *communication data pattern* and *data type*. By considering both a packet's *data pattern* and its *type*, Burns's IDS at least implicitly performs a step of "*linking*" these concepts to one another, which renders this limitation obvious. Ex.1003, ¶¶287-89.

[14.3] wherein the linking is based on the communication data type and the communication data pattern of the at least one response.

As discussed at [17.8] and [14.2], the prior art's consideration of both a packet's *data pattern* and its *type* at least implicitly renders obvious this claimed concept of "*linking*." Ex.1003, ¶290.

17. Claim 15

[15.0] The method of claim 9, further comprising:

[15.1] identifying a communication data pattern corresponding to a request for sharing of data packet types,

See [3.1] above. Ex.1003, ¶292.

[15.2] wherein the identified data pattern is not compatible with the received privacy preference; and

See [3.2] above. Ex.1003, ¶293.

[15.3] providing a response with at least one modified data packet corresponding to the request for sharing types of data packets.

As discussed at [18.3] and [1.8]-[1.9], Burns and Yang render obvious having the IDS modify a packet and then share the modified packet, which renders this limitation obvious. Ex.1003, ¶294.

18. Claim 16

[16.0] The method of claim 15, wherein the modification of the at least one modified data packet is selected from the group consisting of data nullification, blocking, 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.

See [8.0] above. Ex.1003, ¶295.

XII. Discretionary denial is inappropriate.

A. No §325(d) denial

Burns, Yang, and Wittenberg were not cited or considered during prosecution of the '824 patent. 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 ’824 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.

C. No General Plastic denial

The ’824 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).

XIII. Conclusion

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

Respectfully submitted,

Dated: April 10, 2025
/Theodore M. Foster/
Theodore M. Foster
Lead Counsel for Petitioner
Registration No. 57,456

HAYNES AND BOONE, LLP
2801 N. Harwood St., Suite 2300
Dallas, Texas 75201
Customer No. 27683

CLAIMS APPENDIX

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

[1.1] receiving, by the user's device, a request for retrieval of at least one data packet from the user's device,

[1.2] wherein the user's device is configured to provide a response corresponding to the received request;

[1.3] determining, by the remote server, at least one communication data type of the at least one data packet corresponding to the received request,

[1.4] wherein the at least one communication data type is determined in accordance with characteristics of the communication data packet, and

[1.5] wherein the content of the at least one data packet is not read by the remote server for continued operation by the user's device in real time during communication between the remote server and the user's device;

[1.6] receiving, by the user's device, a privacy preference for the user's device,

[1.7] wherein the privacy preference comprises a list of allowed data packet communication types for sharing during communication;

[1.8] modifying data packets corresponding to requests for sharing of responses that are not compatible with the received privacy preference; and

[1.9] maintaining communication between the remote server and the user's device, with sharing of the modified data packets.

[2.0] The method of claim 1, wherein the communication data type of the at least one data packet is determined from metadata of communication with the remote server.

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

[3.1] determining, by the user's device, at least one communication data pattern of at least one data packet corresponding to the received request; and

[3.2] modifying data packets corresponding to requests for sharing of responses

corresponding to at least one data pattern, that are not compatible with the received privacy preference,

[3.3] wherein the privacy preference comprises a list of allowed data patterns for sharing during communication.

[4.0] The method of claim 3, wherein the communication data pattern is determined based on a range 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.

[5.0] The method of claim 1, wherein the communication 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.

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

[6.1] determining, by the remote server, a communication data pattern of the at least one data packet of the received request; and

[6.2] linking at least one response from the user's device to a type of data packet from the user's device,

[6.3] wherein the linking is based on the communication data type and communication data pattern of the at least one response.

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

[7.1] identifying a response corresponding to a request for sharing of data packets,

[7.2] wherein the identified response is not compatible with the received privacy preference; and

[7.3] providing a response with at least one modified data packet corresponding to the request for sharing types of data packets.

[8.0] The method of claim 7, wherein the modification of the at least one modified data packet is selected from the group consisting of data nullification, blocking, 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.

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

[9.1] receiving, by the user's device, a request for retrieval of at least one data packet from the user's device,

[9.2] wherein the user's device is configured to provide a response corresponding to the received request;

[9.3] determining, by the remote server, at least one communication data pattern corresponding to the received request,

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

[9.5] wherein the content of the at least one data packet is not read by the remote server for continued operation by the user's device in real time during communication between the remote server and the user's device;

[9.6] receiving, by the user's device, a privacy preference for the user's device,

[9.7] wherein the privacy preference comprises a list of allowed data patterns for sharing during communication;

[9.8] modifying data packets corresponding to requests for sharing of responses and corresponding data patterns that are not compatible with the received privacy preference; and

[9.9] maintaining communication between the remote server and the user's device, with sharing of the modified data packets.

[10.0] The method of claim 9, wherein the communication data pattern of the at least one response is determined from metadata of communication with the remote server.

[11.0] The method of claim 9, further comprising:

[11.1] determining, by the remote server, the communication data type of the at

least one data packet of the received request; and

[11.2] modifying data packets corresponding to requests for sharing of responses and corresponding data patterns, corresponding to at least one communication data packet type, that are not compatible with the received privacy preference,

[11.3] wherein the privacy preference comprises a list of allowed data packet types for sharing during communication.

[12.0] The method of claim 11, wherein the communication 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.

[13.0] The method of claim 9,

[13.1] wherein the user's device is in communication with an external database with at least one communication data pattern corresponding to a request for sharing of data packets from the user's device, and

[13.2] wherein at least one data pattern compatible with the received request is determined from the database.

[14.0] The method of claim 9, further comprising:

[14.1] determining, by the remote server, a type of the at least one data packet of the received request; and

[14.2] linking at least one data pattern to a type of data packet from the user's device,

[14.3] wherein the linking is based on the communication data type and the communication data pattern of the at least one response.

[15.0] The method of claim 9, further comprising:

[15.1] identifying a communication data pattern corresponding to a request for sharing of data packet types,

[15.2] wherein the identified data pattern is not compatible with the received

privacy preference; and

[15.3] providing a response with at least one modified data packet corresponding to the request for sharing types of data packets.

[16.0] The method of claim 15, wherein the modification of the at least one modified data packet is selected from the group consisting of data nullification, blocking, 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.

[17.0] A system for dynamic management of private data during communication between a remote server and at least one user's device, the system comprising:

[17.1] a memory;

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

[17.3] a privacy preference database,

[17.4] comprising a list of allowed types of data packets for sharing during communication with the at least one user's device;

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

[17.6] a processor, coupled to a response database and to the privacy preference database,

[17.7] wherein the processor is configured to instruct the remote server

[17.8] to determine at least one data type for sharing of data packet that is compatible with the list of allowed patterns of data packets for sharing, and

[17.9] wherein the at least one data type is determined in accordance with characteristics of the communication data packet, and

[17.10] wherein the content of the at least one data packet is not read by the remote server for continued operation by the user's device in real time during

communication between the remote server and the user's device.

[18.0] The system of claim 17, further comprising

[18.1] a communication data pattern database, coupled to the processor and

[18.2] comprising at least one data pattern corresponding to sharing of at least one data packet from the user's device,

[18.3] wherein the processor is configured to modify data packets corresponding to requests for retrieval of data packets and communication data types that are not compatible with communication data patterns from a communication data pattern database.

[19.0] The system of claim 18, wherein data packets from the user's device are selected from the group consisting of user's device files, user's device characteristics, user's device indirect attributes, user's device sensor data, user's device browser data, user's device form data, user's device dynamic memory and user's device static memory.

[20.0] The system of claim 18, wherein at least the communication module and the processor are embedded on a single hardware component.

XIV. 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 '824 Patent is or was involved in the following cases:

Case Heading	Number	Court	Filed
QPrivacy USA LLC v. Cisco Systems, Inc.	No. 2:24-cv-00855	E.D. Tex.	Oct. 21, 2024
U.S. Patent No. 11,816,249, which is related to the '824 Patent: *Ex parte reexamination*	90/019,896	USPTO	April 1, 2025

Petitioner is also concurrently filing a petition for *inter partes* review ("IPR") of claims 1-30 of U.S. Patent No. 11,816,249.

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

Phone: (214) 651-5533
Fax: (214) 200-0853
david.mccombs.ipr@haynesboone.com

Dallas, TX 75201 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 13,978. 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:

Date of service	April 10, 2025
Manner of service	Fed Ex
Documents served	**Petitioner for *Inter Partes* 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.1056.**
Persons Served	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