

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Gregory G. Raleigh
U.S. Patent No.: 9,615,192 Attorney Docket No.: 39843-0166IP1
Issue Date: April 4, 2017
Appl. Serial No.: 15/211,430
Filing Date: July 15, 2016
Title: MESSAGE LINK SERVER WITH PLURAL MESSAGE
DELIVERY TRIGGERS

DECLARATION OF DR. PATRICK TRAYNOR

TABLE OF CONTENTS

I.	QUALIFICATIONS AND BACKGROUND	6
II.	LEGAL PRINCIPLES	13
	A. ANTICIPATION	13
	B. OBVIOUSNESS.....	14
III.	OVERVIEW OF CONCLUSIONS FORMED	15
IV.	BACKGROUND KNOWLEDGE OF ONE OF ORDINARY SKILL IN THE ART	16
V.	TECHNOLOGY BACKGROUND.....	17
VI.	THE '192 PATENT	23
	A. OVERVIEW OF THE '192 PATENT	23
	B. PROSECUTION HISTORY OF THE '192 PATENT	26
	C. INTERPRETATIONS OF THE '192 PATENT CLAIMS AT ISSUE.....	34
VII.	OVERVIEW OF THE PRIOR ART	35
	A. TS-23.140 (SAMSUNG-1004)	35
	B. SHEN (SAMSUNG-1014).....	38
	C. RAKIC (SAMSUNG-1015)	41

D. HOUGHTON (SAMSUNG-1007)	43
E. MUNSON (SAMSUNG-1017)	47
F. ELLISON (SAMSUNG-1019)	48
VIII. GROUND 1: CLAIMS 1, 5, 6, 7, 9, 11-13, AND 15 ARE	
RENDERED OBVIOUS BY TS-23.140	49
A. CLAIM ANALYSIS	49
1. <i>Claims 1 and 15</i>	49
2. <i>Claim 5</i>	84
3. <i>Claim 6</i>	90
4. <i>Claim 7</i>	93
5. <i>Claim 9</i>	94
6. <i>Claims 11-12</i>	96
7. <i>Claim 13</i>	97
IX. GROUND 2: CLAIMS 2 AND 3 ARE RENDERED OBVIOUS	
BY TS-23.140 AND SHEN	98
A. COMBINATION OF TS-23.140 AND SHEN	98
B. CLAIM ANALYSIS	103
1. <i>Claim 2</i>	103
2. <i>Claim 3</i>	104

X.	GROUND 3: CLAIMS 4 IS RENDERED OBVIOUS BY TS-23.140	
	AND ELLISON	105
	A. ELLISON	105
	B. COMBINATION OF TS-23.140 AND ELLISON	106
	C. CLAIM ANALYSIS	110
	1. <i>Claim 4</i>	110
XI.	GROUND 4: CLAIM 8 IS RENDERED OBVIOUS BY TS-	
	23.140 AND RAKIC.....	115
	A. CLAIM ANALYSIS	115
	1. <i>Claim 8</i>	115
XII.	GROUND 5: CLAIMS 1, 5, 6, 7, 9, 11-13, AND 15 ARE	
	RENDERED OBVIOUS BY HOUGHTON AND MUNSON	122
	A. COMBINATION OF HOUGHTON AND MUNSON	122
	B. CLAIM ANALYSIS	126
	1. <i>Claims 1 and 15</i>	126
	2. <i>Claim 5</i>	167
	3. <i>Claim 6</i>	170
	4. <i>Claim 7</i>	171
	5. <i>Claim 9</i>	172

6.	<i>Claims 11-12</i>	174
7.	<i>Claim 13</i>	176
XIII.	GROUND 6: CLAIMS 2 AND 3 ARE RENDERED OBVIOUS BY HOUGHTON, MUNSON, AND SHEN	177
A.	COMBINATION OF HOUGHTON, MUNSON, AND SHEN	177
B.	CLAIM ANALYSIS	181
1.	<i>Claim 2</i>	181
2.	<i>Claim 3</i>	183
XIV.	GROUND 7: CLAIM 4 IS RENDERED OBVIOUS BY HOUGHTON, MUNSON, AND ELLISON	184
A.	COMBINATION OF HOUGHTON, MUNSON, AND ELLISON.....	184
B.	CLAIM ANALYSIS	189
1.	<i>Claim 4</i>	189
XV.	GROUND 8: CLAIM 8 IS RENDERED OBVIOUS BY HOUGHTON, MUNSON, AND RAKIC	193
A.	CLAIM ANALYSIS	193
1.	<i>Claim 8</i>	193
XVI.	ADDITIONAL REMARKS	199

I, Patrick Gerard Traynor, of Gainesville, Florida, declare that:

I. QUALIFICATIONS AND BACKGROUND

1. My name is Patrick Gerard Traynor and I have been retained as an expert witness by Samsung in the matter of *Headwater Research LLC vs Samsung Electronics Co., Ltd.* My qualifications for forming these conclusions are summarized below and are additionally included in my curriculum vitae provided herewith.

2. I earned a B.S. in Computer Science from the University of Richmond in 2002 and an M.S. and Ph.D. in Computer Science and Engineering from the Pennsylvania State University in 2004 and 2008, respectively. My dissertation, entitled “Characterizing the Impact of Rigidity on the Security of Cellular Telecommunications Networks,” focused on security problems that arise in cellular infrastructure when gateways to the broader Internet were created.

3. I am currently a Professor in the Department of Computer and Information Science and Engineering (CISE) at the University of Florida. I was hired under the “Rise to Preeminence” Hiring Campaign and serve as the Associate Chair for Research in my Department. I also hold the endowed position of the John and Mary Lou Dasburg Preeminent Chair in Engineering.

4. Prior to joining the University of Florida, I was an Associate Professor from March to August 2014 and an Assistant Professor of Computer Science from

2008 to March 2014 at the Georgia Institute of Technology. I have supervised many Ph.D., M.S., and undergraduate students during the course of my career.

5. My area of expertise is security, especially as it applies to mobile systems and networks, including cellular networks and messaging. Also, as noted above, my dissertation concentrated on security issues in messaging for cellular systems. I regularly teach students taking my courses and participating in my research group to program and evaluate software and architectures for mobile and cellular systems. I have taught courses on the topics of network and systems security, cellular networks, and mobile systems at both Georgia Tech and the University of Florida. I also advised and instructed the Information Assurance Officer Training Program for the United States Army Signal Corps in the Spring of 2010.

6. I have received numerous awards for research and teaching, including being named a Kavli Fellow (2017), a Fellow of the Center for Financial Inclusion (2016), and a Research Fellow of the Alfred P. Sloan Foundation (2014). I also won the Lockheed Inspirational Young Faculty Award (2012), was awarded a National Science Foundation (NSF) CAREER Award (2010), and received the Center for Enhancement of Teaching and Learning at Georgia Tech's "Thanks for Being a Great Teacher" Award (2009, 2012, 2013).

7. I have published over 100 articles in top conferences and journals in the areas of information security, mobile systems, and networking. Many of my results

are highly cited, and I have received multiple “Best Paper” Awards. I have also written a book entitled “Security for Telecommunications Networks”, which is used in wireless and cellular security courses at a number of top universities.

8. I am a Senior Member of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE). I am also a member of the USENIX Advanced Computing Systems Association.

9. I serve as an Associate Editor for IEEE Security and Privacy Magazine, have been the Program Chair for eight conferences and workshops, and have served as a member of the Program Committee for over 50 different conferences and workshops. I am also currently the Security Subcommittee Chair for the ACM US Technology Policy Committee (USACM).

10. I was a co-Founder and Research Fellow for a private start-up, Pindrop Security, from 2012 to 2014. Pindrop provides anti-fraud and authentication solutions for Caller-ID spoofing attacks in enterprise call centers by creating and matching acoustic fingerprints. Pindrop Security currently employs over 200 people, and their technology is based off of my research (U.S. Patent 9,037,113 B2).

11. I was a co-Founder and Chief Executive of a private start-up, CryptoDrop. CryptoDrop developed a ransomware detection and recovery tool to provide state of the art protection to home, small business, and enterprise users. This technology was also based off of my research (U.S. Patent 10,685,114 B2).

12. I was also a co-Founder and Chief Executive of a private start-up, Skim Reaper. Skim Reaper developed tools to detect credit card skimming devices, and worked with a range of banks, international law enforcement, regulators, and retailers. This technology was also based off of my research (U.S. Patent 10,496,914 B2).

13. I am a named inventor on ten US patents. These patents generally describe methods for determining the origin and path taken by phone calls as they traverse various networks, cryptographically authenticating phone calls, providing a secure means of indoor localization using mobile/wireless devices, detecting credit card skimmers, identifying cloned credit cards, and blocking ransomware from encrypting data.

14. My curriculum vitae, included with this declaration, includes a list of publications on which I am a named author. It contains further details regarding my experience, education, publications, and other qualifications to render an expert opinion in connection with this proceeding.

15. In writing this Declaration, I have considered the following: my own knowledge and experience, including my work experience in mobile systems and networks; my experience in teaching those subjects; and my experience in working with others involved in those fields. In addition, I have analyzed the following publications and materials, in addition to other materials I cite in my declaration:

- U.S. Patent No. 9,615,192 (SAMSUNG-1001), and its accompanying prosecution history (SAMSUNG-1002)
- 3GPP TS 23.140 v6.9.0 (2005-03); 3rd Generation Partnership Project; Technical Specification Group Terminals; Multimedia Messaging Service (MMS); Functional Description; Stage 2 (“TS-23.140”) (SAMSUNG-1004)
- U.S. Patent Pub. No. 2006/0190720 to Ozaki et al. (“Ozaki”) (SAMSUNG-1005)
- WO 2008/048075 to Lee et al. (“Lee”) (SAMSUNG-1006)
- WO 2006/077283 to Houghton et al (“Houghton”) (SAMSUNG-1007)
- U.S. Patent No. 7,925,717 to Chou (“Chou”) (SAMSUNG-1009)
- Open Mobile Alliance; Multimedia Messaging Service Architecture Overview (MMSARCH) specification, *available at* https://www.openmobilealliance.org/release/MMS/V1_1-20040715-A/OMA-WAP-MMS-ARCH-V1_1-20040715-A.pdf (SAMSUNG-1010)
- Open Mobile Alliance; OMA-ERELD-MMS-v1_2-20030923-C, Enabler Release Definition for MMS Version 1.2,” available at https://www.openmobilealliance.org/release/MMS/V1_2-20030923-C/OMA-ERELD-MMS-V1_2-20030923-C.pdf (SAMSUNG-1011)

- U.S. Patent No. 7,509,487 to Lu et al. (“Lu”) (SAMSUNG-1012)
- Technical Specification Group Services and System Aspects Meeting #19, TSGS#19(03)0167, European Telecommunications Standards Institute, 3ETSI / AT-F TD18 – Rapporteur Meeting (4-6 Feb. 2003), available at https://www.3gpp.org/ftp/tsg_sa/TSG_SA/TSGS_19/Docs/PDF/SP-030167.pdf (SAMSUNG-1013)
- U.S. Patent Pub. No. 2005/0207379 to Shen et al. (“Shen”) (SAMSUNG-1014)
- U.S. Patent Pub. No. 2009/0282256 to Rakic et al. (“Rakic”) (SAMSUNG-1015)
- U.S. Patent Publication No. 2009/0240807 to Munson et al (“Munson”) (SAMSUNG-1017)
- EP Patent Application EP1853044A1 to Shenfield (“Shenfield”) (SAMSUNG-1018)
- U.S. Patent No. 7,082,615 B1 (“Ellison”) (SAMSUNG-1019)
- IETF RFC 793, Transmission Control Protocol (Sept. 1981), available at <https://www.ietf.org/rfc/rfc793.txt> (SAMSUNG-1027)
- The TLS Protocol Version v 1.0 (Jan. 1999), available at <https://datatracker.ietf.org/doc/html/rfc2246> (SAMSUNG-1028)

- Needham et al., “Using Encryption for Authentication in Large Networks of Computers” (ACM, Vol. 21, No. 12, Dec. 1978) (“Needham”) (SAMSUNG-1030)
- Schroeder et al., “A Hardware Architecture for Implementing Protection Rings” (ACM, Vol. 15, No. 3, Mar. 1972) (“Schroeder”) (SAMSUNG-1031)
- Saltzer et al., “The Protection of Information in Computer Systems” (IEEE Proceedings, Vol. 63, No. 9, Sept. 1975) (“Saltzer”) (SAMSUNG-1032)
- Li et al., “Symbian OS platform security model” (Login Magazine, Aug. 2010) (SAMSUNG-1033)
- Philip Zimmermann, “Pretty Good Privacy: RSA Public Key Cryptography for the Masses” PGP User’s Guide. Version 1.0, June 1991, available at <https://www.tech-insider.org/free-software/research/acrobat/910605.pdf> (“Zimmerman”) (SAMSUNG-1034)
- B. Ramsdell, S/MIME Version 3 Message Specification, IETF RFC 2633, June 1999, available at <https://datatracker.ietf.org/doc/html/rfc2633> (“Ramsdell”) (SAMSUNG-1035)

- Mostafa, “Transporting data between wireless applications using a messaging system—MMS” (Wireless Comms. and Mobile Computing, July 7, 2006) (“Mostafa”) (SAMSUNG-1036)

II. LEGAL PRINCIPLES

A. Anticipation

16. I have been informed that a patent claim is invalid as anticipated under 35 U.S.C. § 102 if each and every element of a claim, as properly construed, is found either explicitly or inherently in a single prior art reference. Under the principles of inherency, if the prior art necessarily functions in accordance with, or includes the claimed limitations, it anticipates.

17. I have been informed that a claim is invalid under 35 U.S.C. § 102(a) if the claimed invention was known or used by others in the U.S., or was patented or published anywhere, before the applicant’s invention. I further have been informed that a claim is invalid under 35 U.S.C. § 102(b) if the invention was patented or published anywhere, or was in public use, on sale, or offered for sale in this country, more than one year prior to the filing date of the patent application (critical date). And a claim is invalid, as I have been informed, under 35 U.S.C. § 102(e), if an invention described by that claim was described in a U.S. patent granted on an

application for a patent by another that was filed in the U.S. before the date of invention for such a claim.

B. Obviousness

18. I have been informed that a patent claim is invalid as “obvious” under 35 U.S.C. § 103 in light of one or more prior art references if it would have been obvious to a POSITA, taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any so called “secondary considerations” of non-obviousness, which include: (i) “long felt need” for the claimed invention, (ii) commercial success attributable to the claimed invention, (iii) unexpected results of the claimed invention, and (iv) “copying” of the claimed invention by others. For purposes of my analysis above and because I know of no indication from the patent owner or others to the contrary, I have applied a date of January 28, 2009, as the date of invention in my obviousness analyses, although in many cases the same analysis would hold true even at an earlier time than January 28, 2009.

19. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason to modify the single prior art reference, or combine two or more references, in order to achieve the claimed invention. This reason may come from a teaching, suggestion, or

motivation to combine, or may come from the reference or references themselves, the knowledge or “common sense” of one skilled in the art, or from the nature of the problem to be solved, and may be explicit or implicit from the prior art as a whole. I have been informed that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

III. OVERVIEW OF CONCLUSIONS FORMED

20. This expert Declaration explains the conclusions that I have formed based on my analysis. To summarize those conclusions:

- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1, 5-7, 9, 11-13, and 15 of the '192 Patent are obvious over TS-23.140.
- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 2 and 3 of the '192 Patent are obvious over TS-23.140 in view of Shen.
- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 4 of the '192 Patent is obvious over TS-23.140 in view of Ellison.

- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 8 of the '192 Patent is obvious over TS-23.140 in view of Rakic.
- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 1, 5-7, 9, 11-13, and 15 of the '192 Patent are obvious over Houghton in view of Munson.
- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claims 2 and 3 of the '192 Patent are obvious over Houghton in view of Munson and Shen.
- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 4 of the '192 Patent is obvious over Houghton in view of Munson and Ellison.
- Based upon my knowledge and experience and my review of the prior art publications listed above, I believe that claim 8 of the '192 Patent is obvious over Houghton in view of Munson and Rakic.

IV. BACKGROUND KNOWLEDGE OF ONE OF ORDINARY SKILL IN THE ART

21. Based on the foregoing and upon my experience in this area, a person of ordinary skill in the art ("POSITA") relating to the subject matter of the '192 Patent as of January 28, 2009 ("Critical Date") would have had (1) at least

a bachelor's degree in computer science, electrical engineering, or a related field, and (2) three to five years of experience in services and application implementation in communications networks.

Additional graduate education could substitute for professional experience, and *vice versa*.

22. Based on my experiences, I have a good understanding of the capabilities of a POSITA as I was such an individual at the time of the Critical Date. Moreover, I have taught, participated in organizations, and worked closely with many such persons over the course of my career.

V. TECHNOLOGY BACKGROUND

23. The original and still common configuration for networked computing is the client/server architecture. As an early example, users of mainframe computers in the 1960s relied on simple "thin clients" with little to no processing power of their own to outsource their computing tasks to mainframes.

24. The benefits of the client/server architecture have long outlived the mainframe era. Client devices, generally with fewer resources (e.g., storage, bandwidth, processing capabilities, power, uptime), still often rely on servers to centralize services and for content distribution. As an example, an email or instant messaging server would not only be better provisioned than a mobile phone

communicating with it, but also far more likely to be continuously powered and available on the network.

25. Because servers are better provisioned than their corresponding clients, their communication patterns are generally many-to-one. That is, because of their significant resources (and the expense in provisioning them), a small number of servers are expected to interact with a large number of clients.

26. Networking between the devices is facilitated by networking stacks in both the client and the server. The stacks implement what is known as the “Internet Model,” which creates five layers of abstraction for network function: Physical Layer, Data Link Layer, Network Layer, Transport Layer, and Application Layer. The advantage to this approach is that it abstracts away most of the complexity of networking, allowing applications developers to take advantage of communications services without having to know all of the details required to provide said communications.

27. Communication over a network often requires additional security measures. For instance, an adversary may attempt to read or modify the communications between two parties (in both public and private networks). Before the Critical Date, the security community had created a wide array of mechanisms (specifically, protocols using encryption) allowing for secure communications.

28. These protocols are neither new or uncommon. For instance, Needham and Schroeder developed the first network-based encryption key establishment protocol in 1978 (R. Needham and M. Schroeder, “Using encryption for authentication in large networks of computers.” Communications of the ACM. 21 (12): 993–999, 1978)—SAMSUNG-1030. This protocol formed the basis of security protocols (e.g., Kerberos) included in many popular operating systems from the late 1980s and beyond.

29. The need to provide security mechanisms that could be used easily across all communications quickly became widespread. Instead of requiring individual applications or processes to develop their own secure communications protocols, efforts to build secure communications layers into network stacks began in the early 1990s.

30. For instance, the Internet Engineering Task Force (IETF) formed the IP Security (IPsec) Protocol working group in 1992 (IP Security Protocol (ipsec), <https://datatracker.ietf.org/wg/ipsec/history/>). The IPsec protocol, which is implemented in nearly all modern operating systems, can provide cryptographic guarantees for confidentiality and integrity for all communications data above the Network Layer.

31. Another important effort to provide security services directly in the network stack was the Secure Sockets Layer (SSL) standardization process, which

began in 1994 (The SSL Protocol, <https://www-archive.mozilla.org/projects/security/pki/nss/ssl/draft02.html>, 1994). This standard eventually evolved into the Transport Layer Security (TLS) protocol.

32. TLS is implemented above the Transport Layer and provides security for applications. Possibly the most visible of all security protocols, TLS is used widely to secure applications ranging from the Web (e.g., “https” in browsers) and email to teleconferencing and secure messaging. TLS connections now represent over 80% of all requests on the web (Google Transparency Project - HTTPS encryption on the web, <https://transparencyreport.google.com/https/overview?hl=en>, 2023).

33. Security mechanisms within individual systems have also long been known to the state of the art. Separation between process (e.g., different applications, users, operating system-level processes) has long been implemented in modern operating systems. For example, classical time-sharing systems from the 1960s separated user processes into isolated time slots not only to allow for seemingly concurrent execution, but also to limit how those processes interacted with each other.

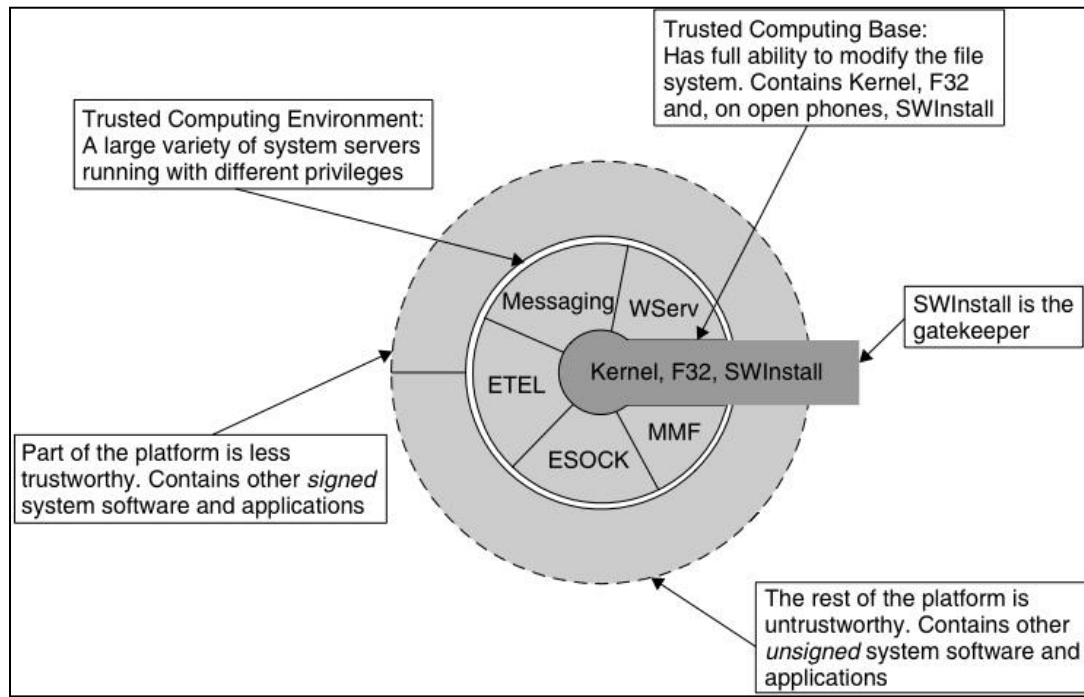
34. Many different techniques were considered. Strong protection from and isolation supported by both hardware and software were codified in the MULTICS system in 1969 (M. Schroeder and J. Saltzer, A Hardware Architecture

for Implementing Protection Rings, *Communications of the ACM*, 15 (3). March 1972)—SAMSUNG-1031. Using “Protection Rings,” operating systems could allow not only for the separation of different processes depending on their security needs, but also allow for communication across such security boundaries through highly-controlled mechanisms. Further innovations in the community lead to increasingly secure enclaves, or operating environments, for individual processes.

35. Due to expense, not all operating systems that followed implemented the entire combined hardware and software controls developed for MULTICS; however, the ideas of process separation and controlled communication between such processes continued to be crucial to systems security (J. Saltzer and M. Schroeder, *The Protection of Information in Computer Systems*, *Proceedings of the ACM Symposium on Operating Systems Principles (SOSP)*, 1974)—SAMSUNG-1032.

36. Advances in hardware made it possible for modern systems, including mobile systems, to implement more secure systems. For example, in the early 2000s, the Symbian mobile operating system (B. Li, E. Reshetova, T. Aura, “Symbian OS platform security model”, *LOGIN: Magazine*, August 2010; Craig Heath, *Symbian OS Platform Security: Software Development Using the Symbian OS Security Architecture* (Wiley, 2006)—SAMSUNG-1033) separated processes into three

“Trust Tiers”: The Trusted Computing Base (TCB), The Trusted Computing Environment (TCE), and Applications, as illustrated below:



37. The first two layers/tiers shown were primarily dedicated to components of the Operating System with relatively higher operating privileged, followed by the Applications tier with applications having relatively lower privileges.

38. The first two tiers included the Trusted Computing Base and the Trusted Computing Environment. As shown, the Trusted Computing Environment includes, among other services, a Messaging service.

39. The Application Tier could be further subdivided into Signed and Unsigned applications. Signed applications could interact with the TCE and have this layer execute higher privileged requests, whereas Unsigned applications could not access that same functionality.

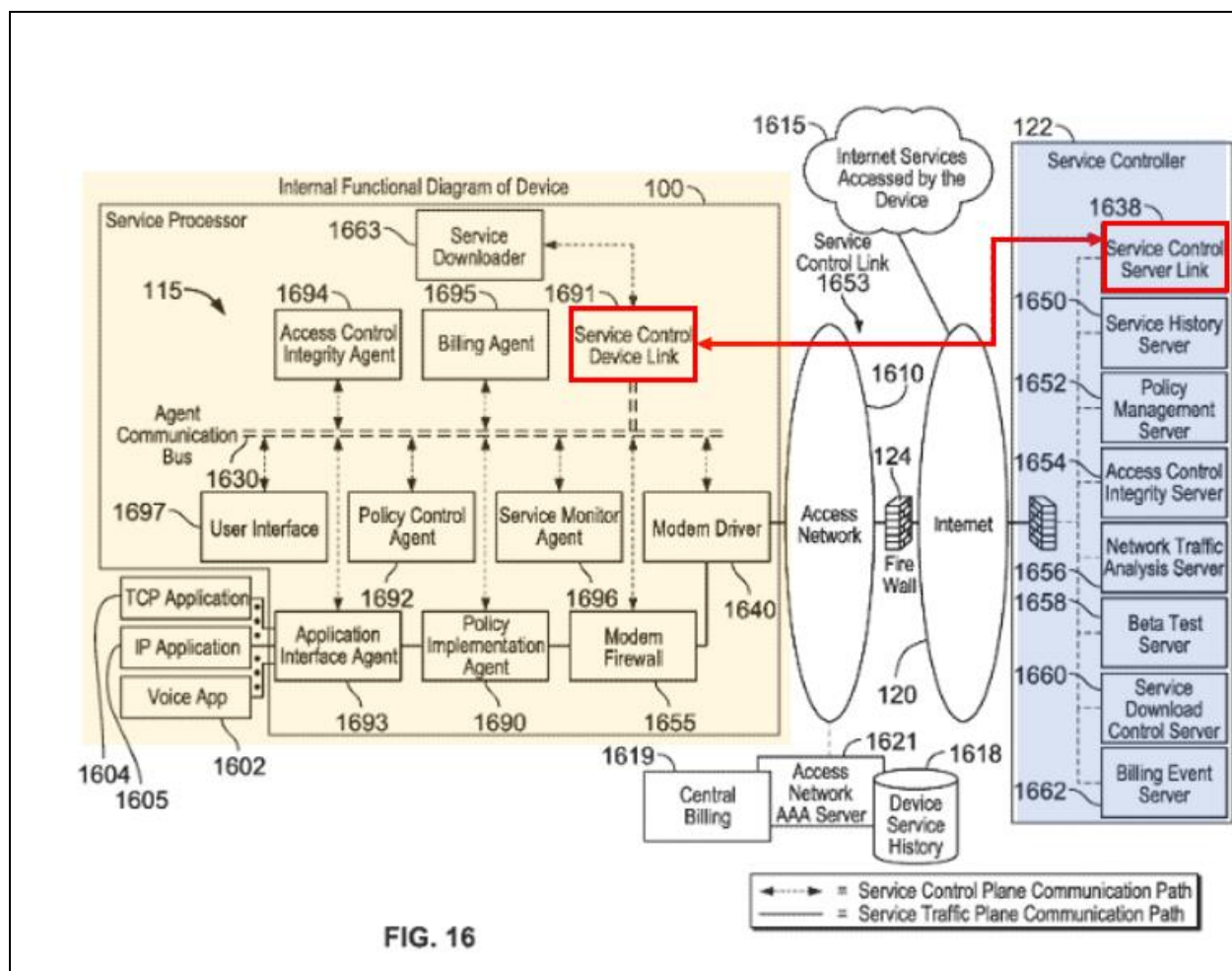
40. Through these mechanisms and others like them on more recent mobile platforms, it has long been possible for processes to operate in secure enclaves on mobile platforms, and with platform-supported encrypted communications across networks.

VI. THE '192 PATENT

A. Overview of the '192 Patent

41. The '192 Patent is directed to “a message link server that maintains secure message links with device link agents on each of a plurality of wireless end-user devices.” SAMSUNG-1001, Abstract.

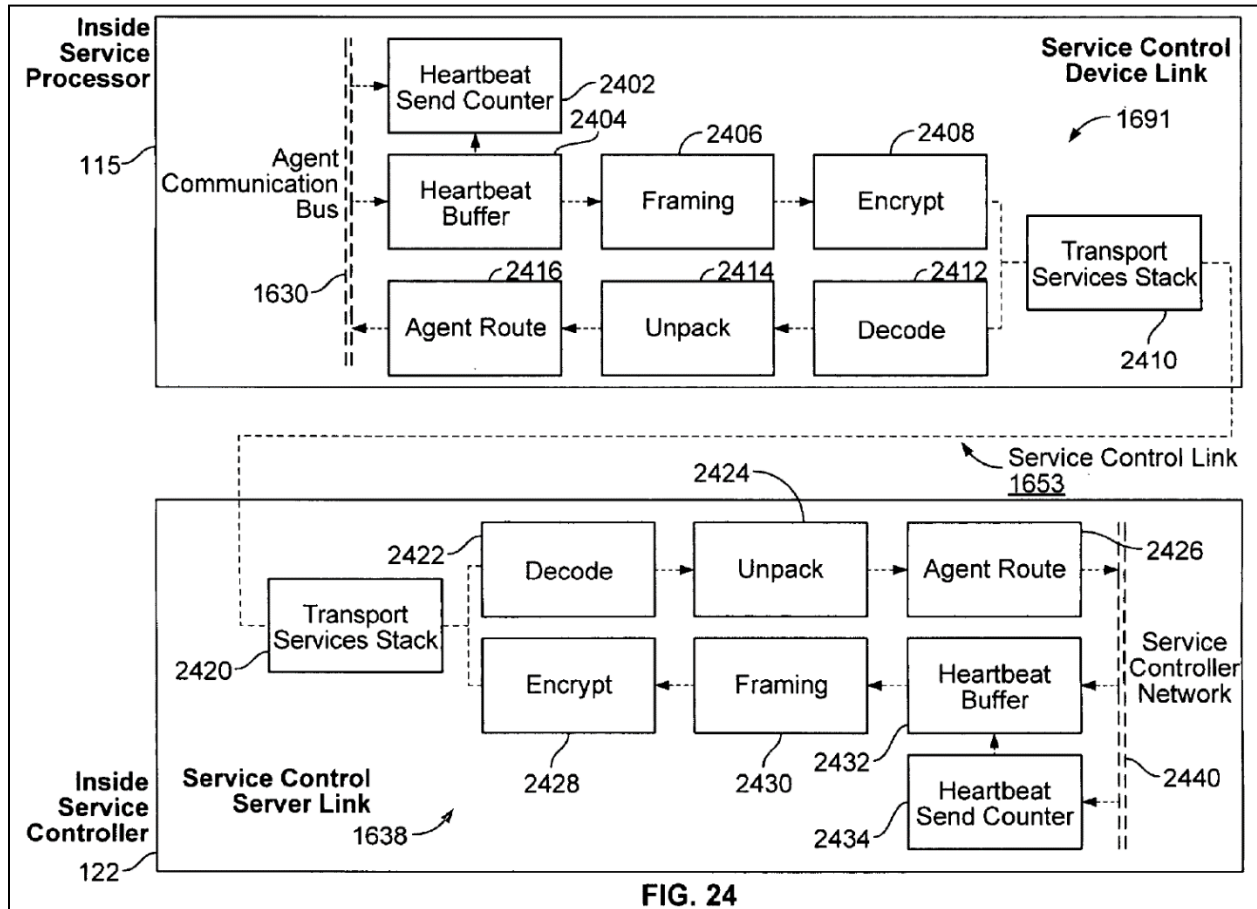
42. FIG. 16 (reproduced below) provides an overview of a network including a server side (referenced as service controller 122), and a device side (referenced as device 100) that are in communication with each other through service control Link 1653:



SAMSUNG-1001, FIG. 16 (annotated); SAMSUNG-1002, 343-355, 81-86.

43. Service Controller 122 includes multiple servers (e.g., 1650, 1652, 1654, 1656, 1662) that are communicatively coupled to the service control server link 1638. SAMSUNG-1001, 68:19-20. On the device side, device 100 includes service control device link 1691, which is communicatively coupled to multiple device agents (e.g., 1697, 1695). SAMSUNG-1001, 37:34-46; 41:45-53; 45:26-30.

44. FIG. 24 (reproduced below) “is a functional diagram illustrating the service control device link 1691 of [device 100] and the service control service link 1638 of the service controller 122.” SAMSUNG-1001, 87:49-52.



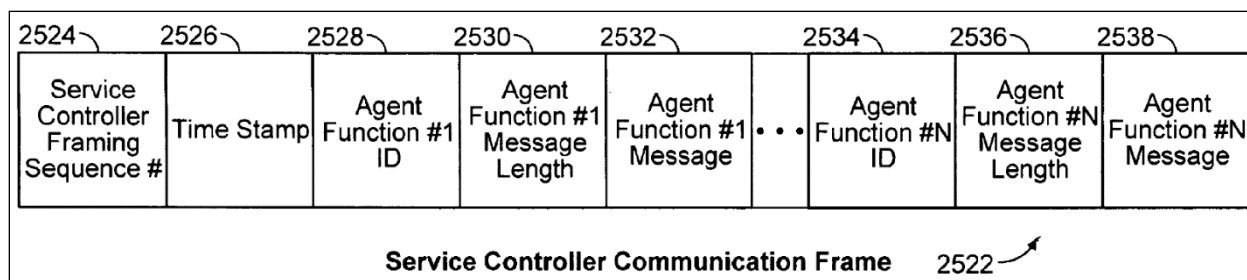
SAMSUNG-1001, FIG. 24

45. As shown, the service control server link 1638 includes a transport services stack 2420 that provides “basic transport services” and can be “implemented with standard secure or open Internet networking protocols, such as TLS or TCP.”

46. “Network elements send messages to” this server, with the messages being “targeted for delivery of message content to specific software components on respective ones of the wireless end-user devices.” SAMSUNG-1001, Abstract.

47. The messages are buffered “until one of several triggers occur, such that messages that do not have time-critical messaging needs may be buffered until the occurrence of a trigger other than the receipt of that message by the server.” *Id.* The service control server link element 1638 can communicate one or more messages to the service control device link 1691. *Id.*, 90:19-53. A message received from server link element 1638 is associated with an identifier (e.g., “an agent first function ID 2528”) and content (e.g., “an agent first function message 2532”) included on the communication. *Id.*

48. The service control device link 1691 routes the content of the message “to the appropriate agent” on device 100 using the identifier. SAMSUNG-1001, 89:21-33.



SAMSUNG-1001, FIG. 25.

B. Prosecution History of the '192 Patent

49. The application for the '192 Patent was filed on July 15, 2016, and claims priority to multiple non-provisional and provisional applications, with the earliest of such applications being filed on January 28, 2009. SAMSUNG-1002, 730-733.

50. During prosecution, the Examiner initially rejected the then-pending claims on §112 indefiniteness and §103 obviousness grounds. SAMSUNG-1002, 343-355. With respect to the obviousness rejections, the Examiner rejected the then-pending independent claims and certain dependent claims (claims 2, 5, 7-8, 10, 12-14, 16) in view of Heinonen (U.S. Patent No. 7,151,764) and Zhou (U.S. Patent Pub. No. 2006/0182137). SAMSUNG-1002, 348-351; *see generally id.*, 343-355. The Examiner rejected the remaining dependent claims in view of Heinonen and Zhou, and further in view of the following references: Chattopadhyay et al. (US Patent No. 7,191,248), Pineau (US Publication No. 2003/0184793), Takeshima et al. (US Publication No. 2004/0054779), and Dilman et al. (US Publication No. 2002/0138599). *Id.*, 351-355.

51. In response, Applicant made minor amendments to the then-pending independent claims (*see id.*, 74-78), and argued that the proposed combination of Heinonen and Zhou failed to render obvious multiple limitations of the independent claims, which correlate to independent claim features of [1pre], [1a], [1b], [1c1], and [1d1]-[1d3]. *Id.*, 81-86. As to the latter, Applicant advanced the following argument:

First, with respect to Heinonen, it is critical to note that Heinonen is directed to solving a problem with short-range wireless communications (such as Bluetooth) where a mobile device may not be in range of an access point long enough to establish a session, but the access point wants to quickly supply some useful information to passing mobile devices without establishing a session. An access point sends out Bluetooth inquiry packets indiscriminately to any mobile Bluetooth devices in range. (Heinonen, col. 12, 11. 1-3.) Any scanning device can choose to respond with an inquiry response packet that includes "class-of-device" information. (Id, 11. 13-27.) The access point can then respond with a paging packet, and according to Heinonen's technique, the paging packet includes a special "class-of-device" that notifies the responding device that a following paging packet will contain an Access Point Service Indicator (APSI) message. (Id, 11. 53-58.) The access point performs a cache search for stored content appropriate for the mobile device's class-of-device type, and builds the special APSI message from whatever the cache hit returns. (Id, col. 14, 11. 40-53.) In the event that there is a cache miss, the access point enlists the help of a content server in order to locate appropriate content for the particular class-of-device type. (Id, 11. 53-66.) The access point then

builds the APSI message and delivers it to the mobile device. (Id, col. 15, 11. 1-9.) The user device bypasses the protocol stack of the user device (since no session has been established with the access point) to deliver the APSI content directly to the GUI display function of the device. (Id., 11. 9-25.)

SAMSUNG-1002, 81-82.

52. In view of this assertion, Applicant **first** argued that “Heinonen discloses no message link server because Heinonen’s “access point generates its own APSI messages, when it chooses, either from cached content, or from a content server content, upon a cache miss, but has no mechanism to deliver a message from a network element to an identified software component on an identified device.” *Id.*, 81-84.

53. **Second**, Applicant argued that Heinonen discloses “no transport services stack to maintain a respective secure message link through an Internet network between the message link server and a respective device link agent” because “during the time when the simple APSI message is communicated to a device, the access point and the device maintain no link at all (see, e.g., col. 10, 11. 32-36, connection established at some point after APSI message is sent); nothing is secure as the content is in a paging packet; and nothing is through the Internet network, as

these are directly communicated link-layer packets that bypass all ‘middleware’ protocol layers.’” *Id.*

54. **Third**, Applicant argued that “Heinonen discloses no wireless end-user devices comprising multiple software components authorized to receive messages via the device link agent on that device” because, per Applicant, “[t]here is no mechanism to authorize other software components to receive messages, or direct messages to such components.” *Id.*

55. **Fourth**, Applicant argued that “Heinonen discloses no interface for network elements using the access point as a message link server” because Heinonen’s “access point merely communicates with passing mobile devices opportunistically, nothing in Heinonen would lead one to expect that network elements would ever queue specific messages with an access point on the off-chance that a desired device passes within range; the content server merely supplies access-point requested content to an access point, it does not direct messages to the access point for delivery to respective software components on respective wireless end-user devices, as identified in the messages.” *Id.*

56. **Fifth**, Applicant argued that “Heinonen discloses no message buffering system” because “any APSI message delivered to a mobile device is built locally on-the-fly based on the cache content search and delivered immediately. The whole

point of Heinonen it to be able to deliver a small amount of useful display content, quickly, to a passing device that is not connected to the access point.” *Id.*

57. Applicant then argued that Zhou did not overcome the deficiencies in Heinonen. SAMSUNG-1002, 83. Specifically, Applicant argued that

As neither reference has a message link server, the combination would not teach or suggest to one of ordinary skill such a server. One reference [(Heinonen)] maintains no links-secure, for messaging, or through the Internet; the other reference [(Zhou)] is used within a single system and not on the Internet, and resorts to individual kernel-maintained connections between individual process pairs-thus the combination would not teach or suggest to one of ordinary skill the claimed transport services stack. One reference has no interface to receive messages from a plurality of network elements for delivery of respective message content for processing to respective ones of the software components identified in the messages; the other reference sets up many dedicated message channels between different pairs of programs running on the same system, but has no interface shared by multiple elements-thus the combination would not teach or suggest to one of ordinary skill the claimed interface to a network. One reference [(Heinonen)] teaches no buffering and a need for immediate delivery

of a locally-generated message; the other reference [(Zhou)] teaches individual buffers for individual pairs of communicating programs, but no buffer shared for multiple sources that may be directing messages to a same remote device, where a message delivery trigger will cause the delivery of messages to multiple "consumers" on the same remote device.

SAMSUNG-1002, 83-84.

58. Applicant further argued that neither reference teaches the claim element reciting “as a trigger, the occurrence of an asynchronous event with time-critical messaging needs.” SAMSUNG-1002, 84.

59. With respect to dependent claims, Applicant relied upon the arguments with respect to the independent claims and offered additional arguments for certain dependent claims. *See* SAMSUNG-1002, 84-86.

60. In the next Action, the Examiner allowed the application upon conducting an interview with the Applicant. The allowance included an examiner’s amendment that amended the independent claim in the manner shown below (and made similar amendment to the other independent claim as well as conforming edits to certain dependent claims):

2. (Currently Amended) A message link server comprising:

a transport services stack to maintain a respective secure message link through an Internet network between the message link server and a respective device link agent on each of a plurality of wireless end-user devices, each of the wireless end-user devices comprising multiple software components authorized to receive and process data from secure message link messages received via ~~[[the]]~~ a device link agent on that device;

an interface to a network to receive network element messages from a plurality of network elements, the received network element messages comprising respective

message content and requests for delivery of the respective message content to respective wireless end-user devices, the respective message content including data for, and an identification of, a respective one of the authorized software components for processing to respective ones of the software components identified in the messages; and

a message buffer system including a memory and logic, ~~[[to]]~~

the memory to buffer content from the received network element messages for which delivery is requested ~~messages identified for delivery~~ to a given one of the wireless end-user devices ~~for which the respective secure message link is maintained,~~

the logic to determine when one of a plurality of message delivery triggers for the given one of the wireless end-user devices has occurred, wherein for at least some of the received network element messages, the receipt of such a message by the message buffer system is not a message delivery trigger, and for at least one of the message delivery triggers, the trigger is ~~[[the]]~~ an occurrence of an asynchronous event with time-critical messaging needs, and

upon determining that one of the message delivery triggers has occurred, the logic further to supply one or more messages comprising the buffered content messages to the transport services stack for delivery on the secure message link maintained between the transport services stack and ~~[[the]]~~ a device link agent on the given one of the wireless end-user devices.

SAMSUNG-1002, 13-19.

61. In the statement of reasons for allowance, the Examiner noted as follows: “The claims are allowable over the prior art of record in view of the claim amendments above, for the reasons put forth in the Remarks (pages 9-11) of 02-01-2017” (which correspond to SAMSUNG-1002, 81-84) and “after further search and consideration, the prior arts of record either taken alone or in combination neither anticipates nor renders obvious the claimed subject matter of the instant application that is taken as a whole including the particular features incorporated in each independent claims.” *Id.*, 18.

62. The application subsequently issued as the '192 Patent. *Id.*, 1.

C. Interpretations of the '192 Patent Claims at Issue

63. I have been informed by Counsel and understand that the best indicator of claim meaning is its usage in the context of the patent specification as understood by one of ordinary skill. I further understand that the words of the claims should be given their plain meaning unless that meaning is inconsistent with the patent specification or the patent's history of examination before the Patent Office.

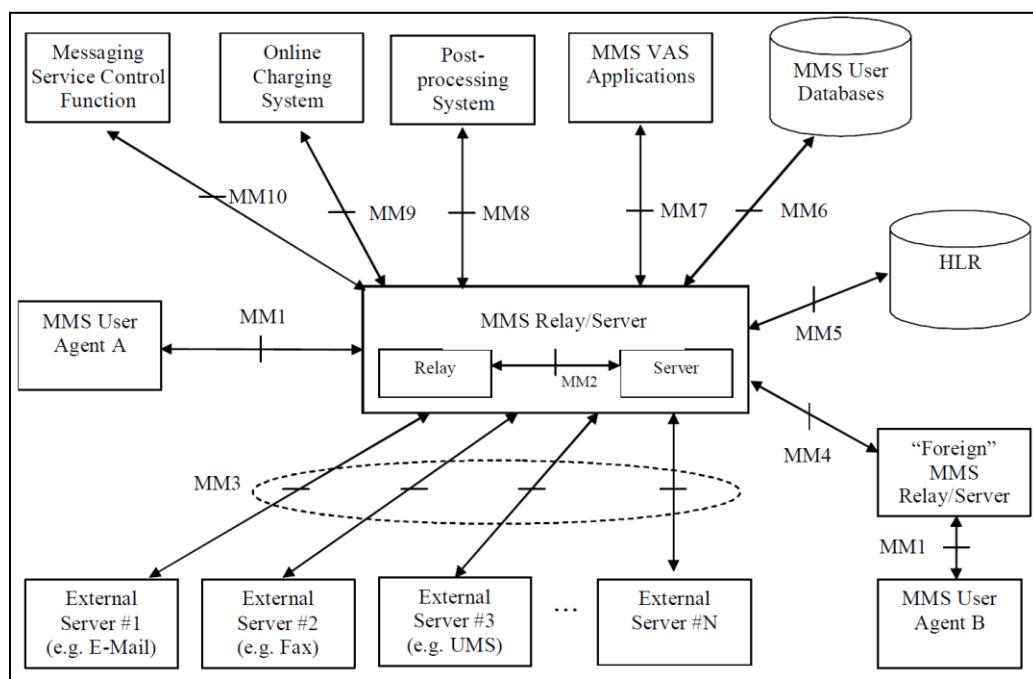
64. Counsel has also informed me, and I understand that, the words of the claims should be interpreted as they would have been interpreted by one of ordinary skill as of the Critical Date (January 28, 2009). I have been informed by Counsel

that I should use the Critical Date as the point in time for claim interpretation purposes.

VII. OVERVIEW OF THE PRIOR ART

A. TS-23.140 (SAMSUNG-1004)

65. TS-23.140 describes “non-realtime Multimedia Messaging Service, MMS,” with an example environment where MMS is implemented being shown below:



SAMSUNG-1004, §1, pp. 10, 23, FIG. 3.

66. In this environment, “MMS Relay/Server is responsible for storage and notification, reports, and general handling of messages” between different entities, including among others, a MMS User Agent, a Foreign MMS Relay/Server, and MMS VAS (Value Added Services) Applications. *Id.*, §5.2, p. 21. For example,

MMS Relay/Server communicates with MMS User Agent through communication link MM1, and with MMS VAS Applications through communication link MM7. *Id.*, pp. 23-24.

67. “MMS User Agent [is an] application residing on a [user equipment (UE)] that performs MMS-specific operations on a user’s behalf.” *Id.*, p. 14. Communication link “MM1 is used to submit Multimedia Messages [(MM)s] from MMS User Agent to MMS Relay/Server, to let the MMS User Agent pull MMs from the MMS Relay/Server, let the MMS Relay/Server push information about MMs to the MMS User Agent as part of an MM notification, and to exchange delivery reports between MMS Relay/Server and MMS User Agents.” *Id.*, §6.3, p. 24.

68. MMS VAS Applications are “[a]pplications providing Value Added Services (e.g. news service or weather forecasts) to MMS users.” *Id.*, p. 14. Communication link “MM7 is used to transfer MMs from MMS Relay/Server to MMS VAS applications and to transfer MMs from MMS VAS applications to MMS Relay/Server.” *Id.*, §6.9, p. 25. The received MMs are then “targeted to the recipient” MMS User Agent by the MMS Relay/Server and through MM1. *Id.*, §7.1.13, p. 41.

69. “Apart from using MMS as a service for users to exchange messages, MMS may also be used to transport data specific to applications.” *Id.*, §7.1.18, p. 54-55. In particular, MMS can be “used to transport data specific to applications

between two MMS User Agents or an MMS User Agent and an MMS VAS Application (or vice versa).” *Id.*

70. An application-specific message includes an “application identifier of the destination application” residing on the recipient user device. *Id.*, p. 55. “Upon reception of an abstract message containing a destination application identifier[,] the receiving MMS User Agent...shall first check if the destination application resides on [the UE device].” SAMSUNG-1004, §7.1.18.2.2, p. 56.

71. “If the destination application resides on a receiving MMS User Agent, the MMS User Agent shall immediately route the received MMS information on to the destination application that is referred to from the destination application identifier.” *Id.*

72. TS-23.140 contemplates using security protocols such as Transport Layer Security (TLS) for its communication links. *See, e.g.*, §7.1.13.1 (“For additional security, HTTP may be carried over a TLS [66] session to the MM7 interface.”). For example, TS-23.140 notes that [d]etails for implementation of the MM1 transfer protocol are described in Annex B.” SAMSUNG-1004, §6.3, pp. 24-25. Annex B refers to SAMSUNG-1011, which in turn refers to SAMSUNG-1010 for implementation details of MM1. SAMSUNG-1004, p. 162. SAMSUNG-1010 explains that “[t]he TLS [WP-TLS] transport layer security protocol provides for secure data transmission between the MMS Client and the MMS Proxy-Relay in

architectural configurations that employ HTTP based protocol stacks for MMSM implementation.” SAMSUNG-1010, p. 22.

B. Shen (SAMSUNG-1014)

73. Like TS-23.140, Shen’s “invention relates to multimedia message service (MMS), particularly a multimedia messaging service system, a gateway and a user device” and “a method for receiving/sending multimedia message.” SAMSUNG-1014, [0001].

74. Shen explains that, “[b]ecause MMS is a message store/forward system, MMS messages have to be stored and forwarded by the MMS Center owned by a telecom operator when enterprise users communicate with the backend system (MMS applications 112) with a multimedia messaging user device (hereinafter referred to as “MMS user device”) via the MMS channel.” SAMSUNG-1014, [0030], FIG. 1 (showing MMS Gateway including the MMS Server/Relay facilitating communication between MMS applications and a MMS User Device).

75. “Therefore there is a potential security problem in ... [such] systems since it is not an end-to-end solution for enterprise mobile applications.” SAMSUNG-1014, [0004]. Shen’s invention contemplates multiple different embodiments to solve this problem, one of which is described with reference to Shen’s FIG. 5 below.

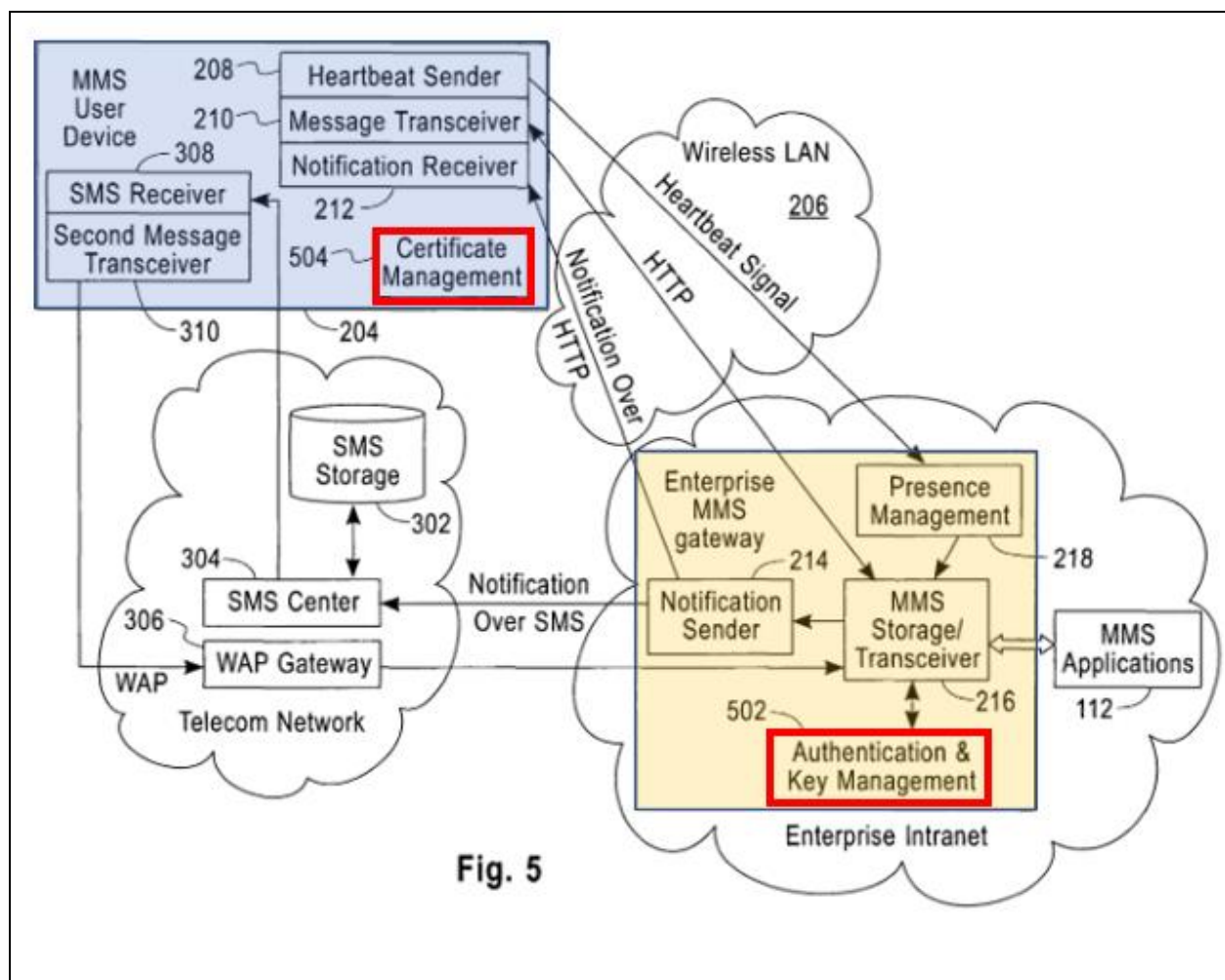


Fig. 5

SAMSUNG-1014, FIG. 5.

76. Specifically, in this embodiment, “an authentication and key management module 502 is added in the MMS gateway to distribute symmetric keys to users and a certificate management module 504 is added in the user device.” SAMSUNG-1014, [0054]. “Both the MMS gateway and the user device deploy the same symmetric cipher to encode and decode MMS messages to protect the privacy of these messages.” *Id.* The encryption may be realized in the following manner:

- First, if “the MMS user device has no certificate of the MMS gateway, it ... sends a CA (certificate) requirement to the [MMS] gateway. After the gateway confirms that the user has the access to send and receive MMS according to the user device number, the [MMS] gateway's CA is sent to the user device....” SAMSUNG-1014, [0056].
- Next, “the MMS user device sends a Master Key requirement to the gateway and the gateway creates a session ID and a master key, generally a random object, and sends them to the device via SMS or WAP response. The random object is encrypted by the gateway's private key.” SAMSUNG-1014, [0057].
- The “random object” is then “decoded by the user device and mixed with its phone number and session ID to generate a shared secret key.... At the same time, the gateway also generates a shared secret key with the same algorithm.” SAMSUNG-1014, [0058].
- And then, “[b]oth the MMS user device and the MMS gateway can secretly communicate with each other by encrypting MMS messages with the shared secret key.” SAMSUNG-1014, [0059].

77. Shen further contemplates that, in the above encryption process, both the MMS user device and the MMS gateway can update “the shared secret key by

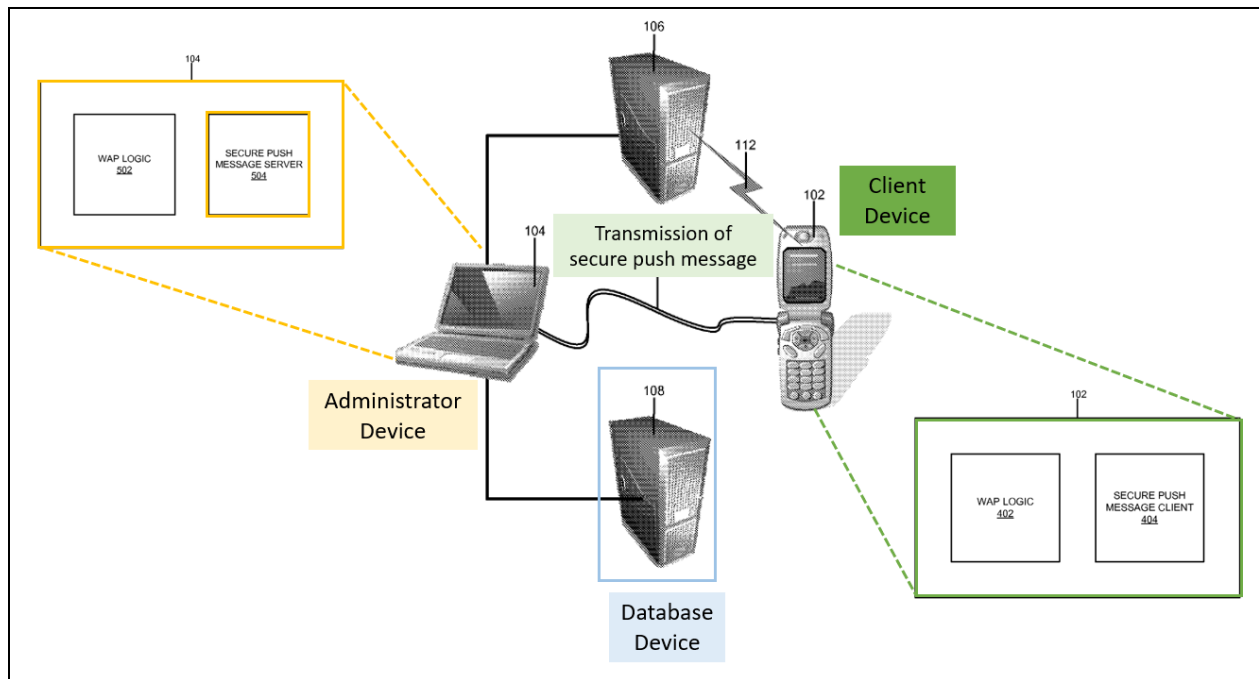
changing the session ID to prevent the threat of attacks stemming from key reuse.”
SAMSUNG-1014, [0060].

78. In this manner, Shen’s “invention ... provides a MMS system supporting both telecom network and wireless LAN and having improved security,” including “provi[sion] [of] an end-to-end security solution for MMS applications,” which can favorably be implemented for use with “enterprise applications.” SAMSUNG-1014, [0017], [0021] (describing additional advantages stemming from Shen’s invention).

C. Rakic (SAMSUNG-1015)

79. Rakic relates to secure push messages. SAMSUNG-1015, Title. Rakic discloses that, in the context of push communications, “a sender may send a secure push message to a client device,” where the secure push message may bear an electronic signature that validates (e.g., authenticate) the secure push message at the client device. *Id.*, [0040].

80. Rakic discloses a networked environment in which an administrator device 104, which includes the secure push message server 504, communicates secure push messages over a network to a client device 102:



SAMSUNG-1015, FIG. 1, 4-5 (annotated)

81. As shown, the environment additionally includes a database device 108 that stores information related to client devices (e.g., codes obtained from a client device such as IMEI or a component in the client device (e.g., removable memory)). SAMSUNG-1015, [0047]-[0049].

82. The secure push message server 504 generates a key, an electronic signature, and a secure push message for a message from a particular user/administrator. *Id.* Specifically, the secure push message server generates the key based on the above-described code(s) (e.g., IMEI) that are obtained from client device 102 or a component thereof, and subsequently, generate an electronic signature based on the key and the message. *Id.*, [66]-[67]. In particular, the electronic signature is created by “generat[ing] a first key by combining an

administrator code, a client device identifier that identifies a client device, and subscriber information that is associated with a service to which a user subscribes.”

Id.

83. The sending device then “hash[es] the first key to generate a second key, and use the second key to sign a data block within the secure push message to produce an electronic signature.” *Id.* The secure push message server 504 generates and sends the secure push message to client device 102, e.g., by appending the electronic signature to the message and/or other information (e.g., a timestamp). *Id.*, [67], FIG. 8.

84. At the client device 102, a secure push message client 404 can validate the message and the sender by generating a key and then generating a signature using this generated key and the data block of the received message. [0109]-[0112]. If the signature portion of the received message matches the signature portion of the received message, the secure push message is determined to be valid. *Id.*

85. In this manner, Rakic discloses that the secure push message server generates signatures for inclusion in push messages, and these signatures are then used by the (1) client device to “authenticate the sender” and (2) “verify that an intended recipient of the secure push message is the client device, and that the client device includes a correct component.” *Id.*, [0041].

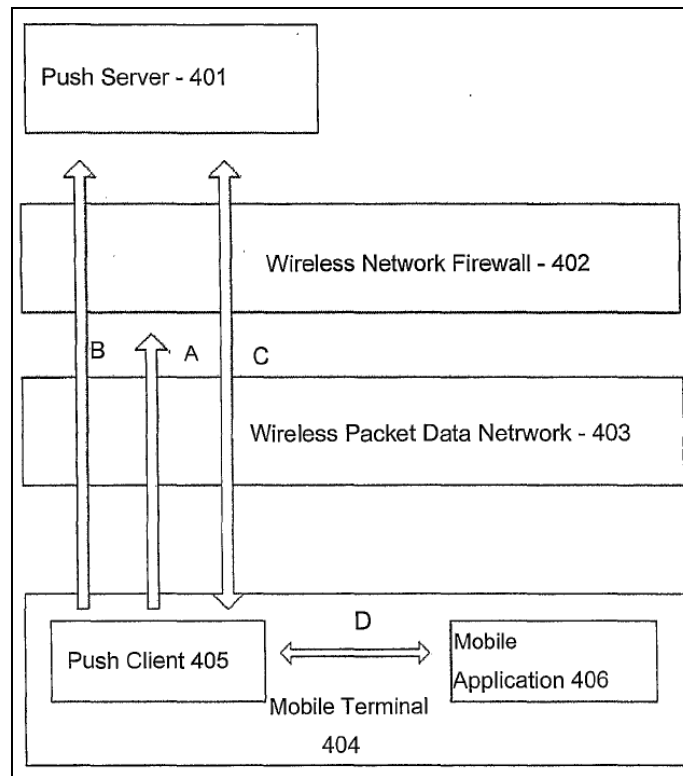
D. Houghton (SAMSUNG-1007)

86. Houghton discloses “technique[s] for providing a push-type command of applications on a mobile terminal by a remote server computer.” SAMSUNG-1007, 11.¹

87. In some embodiments, Houghton seeks to push, from a server and to applications on a mobile terminal, messages/commands that “initiate control of one or more applications on an individual’s mobile terminal.” *Id.*, 10-11. Based on the pushed commands, the server can deliver to the mobile applications executing on mobile terminals “real time information, interactivity and rich media content, regardless of which mobile operator is being used and where the mobile user happens to be at that moment.” *Id.*

88. One example embodiment “for pushing mobile application launch commands to a mobile terminal” is shown in FIG. 4, which is reproduced below:

¹ All references to page numbers in Houghton (SAMSUNG-1007) are to the page numbers printed on the document rather than to page numbers that will be stamped on the exhibit.



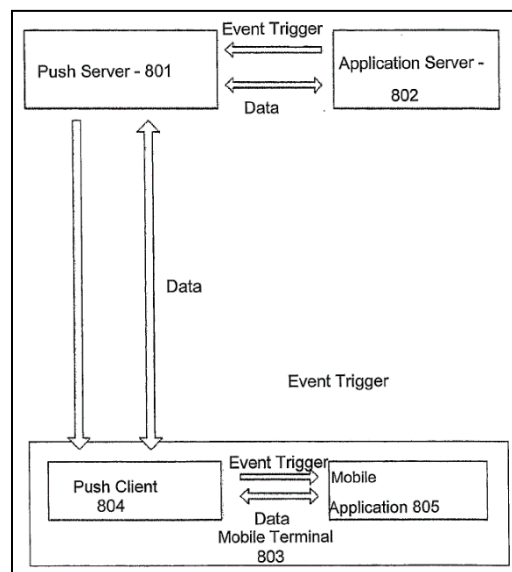
SAMSUNG-1006, 15, Figure 4

89. In this embodiment, after a “connection is established between the client 405 and the server 401,” push client 405 pushes “a message to the” push server 401 or vice versa, as illustrated by arrow C. *Id.*, 20

90. The push message includes “application-specific” information “specifying which mobile application 406 from a plurality of such applications” is the target of the push message. *Id.*, 21. When the mobile terminal receives this message, “appropriate automated actions and user interface media capabilities of the mobile terminal as specified in the command C from the server 401 are executed by the push client 405.” *Id.* “The most common case is playing a sound, asking the

user if the mobile application should be launched, and if affirmative[,] passing the command details (arrow D) to the specified mobile application.” *Id.*

91. Figure 8 shows another embodiment that depicts the above-described communication between the mobile terminal and the push server 801, and further depicts “a data connection between application server 802 and mobile application 805,” which is “established with the aid of the persistent managed, tested and configured data connection in Figure 4, arrow C between push server 401 /801 and push client 405/804.” *Id.*, 23.



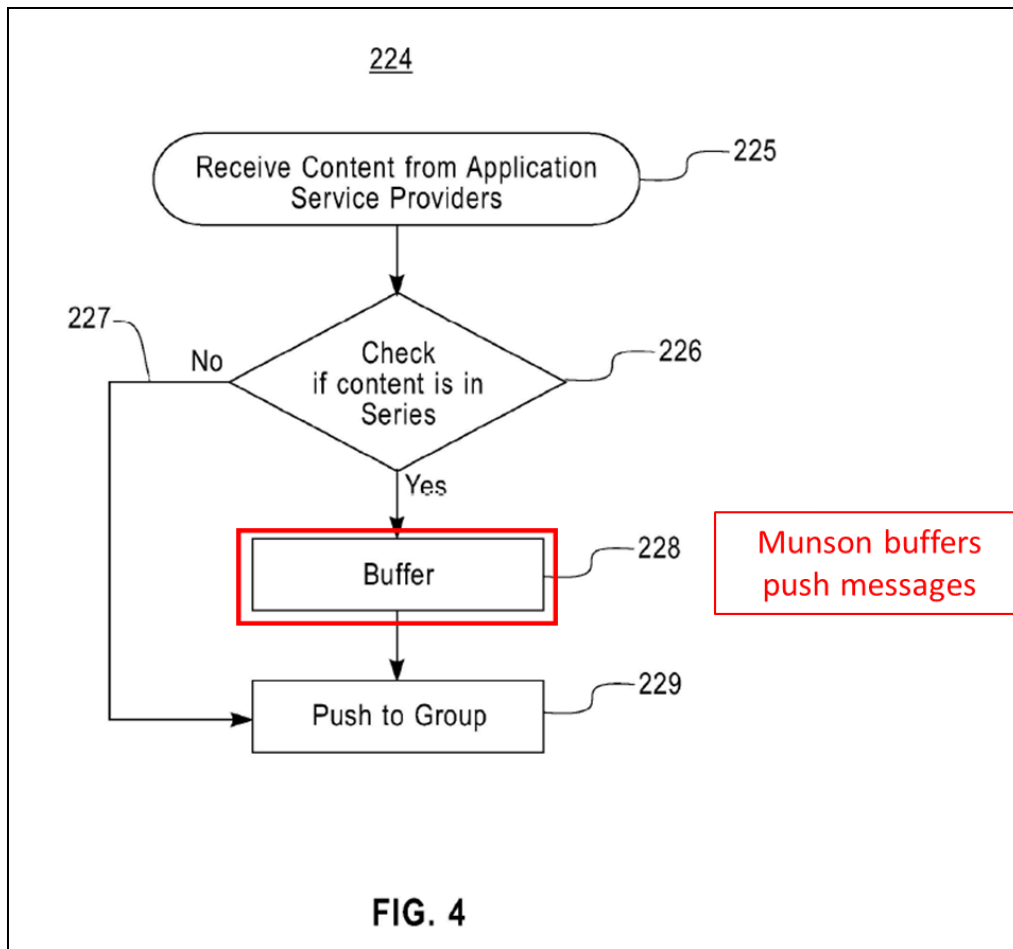
SAMSUNG-1007, Figure 8

92. Houghton further discloses using a “connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS” to facilitate a secure communication link between the push server and mobile terminal. *SAMSUNG-1007*, 20. Houghton also

contemplates “exchanging security keys” between push client and push server.
SAMSUNG-1007, 29.

E. Munson (SAMSUNG-1017)

93. Munson discloses a method of “pushing contents to client devices.”
SAMSUNG-1007, Abstract. Munson discloses “group pushes” where content is buffered and sent to multiple devices simultaneously, “serializ[ing]” content such that a series of messages are delivered to a particular device simultaneously, and various message queuing policies. SAMSUNG-1017, 3:7-67, 4:1-56, FIGS. 1-5. Illustrated below is an example process according to Munson, showing buffering of received push messages:



SAMSUNG-1007, FIG. 4.

F. Ellison (SAMSUNG-1019)

94. Ellison discloses a “method and apparatus to protect a subset of a software environment.” SAMSUNG-1019, Abstract. This includes a plurality of “operating system nub key[s] (OSNK)” which are “unique to an operating system (OS) nub.” *Id.* A “usage protector” uses the OSNK to “protect usage of [a] subset of the software environment.” *Id.*

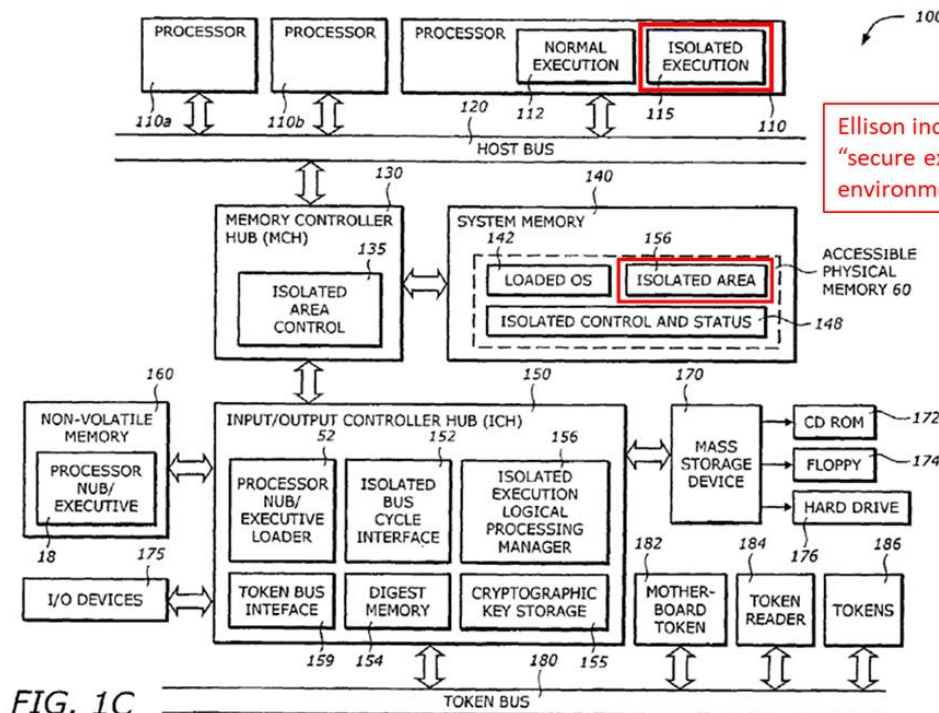


FIG. 1C

SAMSUNG-1019, FIG. 1C.

VIII. GROUND 1: CLAIMS 1, 5, 6, 7, 9, 11-13, AND 15 ARE RENDERED OBVIOUS BY TS-23.140²

A. Claim Analysis

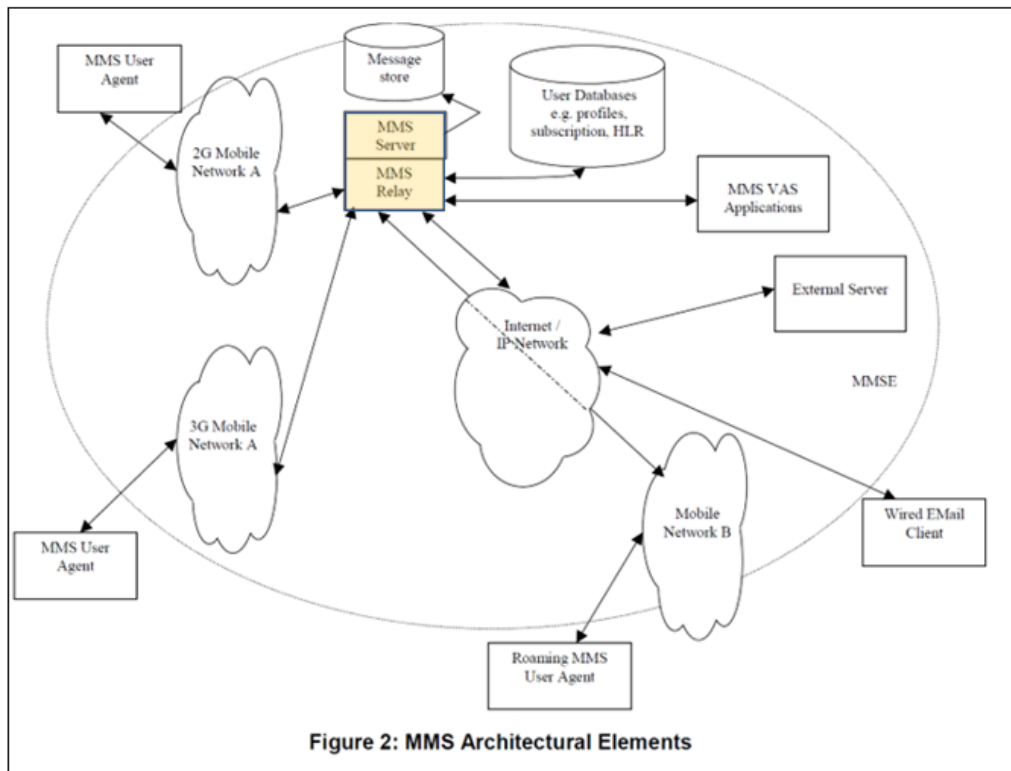
1. Claims 1 and 15

(a) [1pre]: A message link server comprising / [15pre]: A method of operating a message link server, comprising:

² I note that the above descriptions of the references and the below provided combinations and analysis with respect to those combinations are incorporated into each mapping that includes citations to them. Additionally, all emphasis included in this declaration has been added unless otherwise indicated.

95. If the preamble is limiting, TS-23.140 renders obvious a message link server (e.g., the MMS Server/Relay) and a method for operating the same.

96. As described below, TS-23.140 discloses a MMS Relay/Server that facilitates delivery of messages between different entities over a network (as shown below):



SAMSUNG-1004, §4.2, p. 17, fig. 2 (providing an MMS Relay/Server that facilitates delivery of messages between user devices and other elements in the network).

97. The MMS Relay/Server “is responsible for storage and handling of incoming and outgoing messages and for the transfer of messages between different messaging systems.” *Id.* (noting that this server may be a single logical element or

may be separated into MMS Relay and MMS Server elements; describing generation of charging data records (CDR) when retrieving or delivering multimedia messages (MM) and for Value Added Service Provider (VASP) operations).

98. As shown above, MMS communications are facilitated between MMS User Agents (i.e., between two user equipment (UEs)) or between MMS User Agents and MMS VAS Applications.

99. An MMS User Agent is an application is an application residing on a UE, whereas an MMS VAS Application is an application that provides Value Added Services (e.g., news service or weather forecasts) to MMS Users and resides on a separate network entity, as shown in the above figure. *Id.*, 14.

100. MMS can also be “used to transport data specific to applications.” *Id.*, p. 54. Such applications can be downloaded “to a mobile phone” and can register with an MMS User Agent of the UE (user equipment) or an MMS VAS application. *Id.* Once registered, MMS can be used “to transport data specific to applications between two MMS User Agents or an MMS User Agent and an MMS VAS Application (or vice versa).” *Id.*, 54-55.

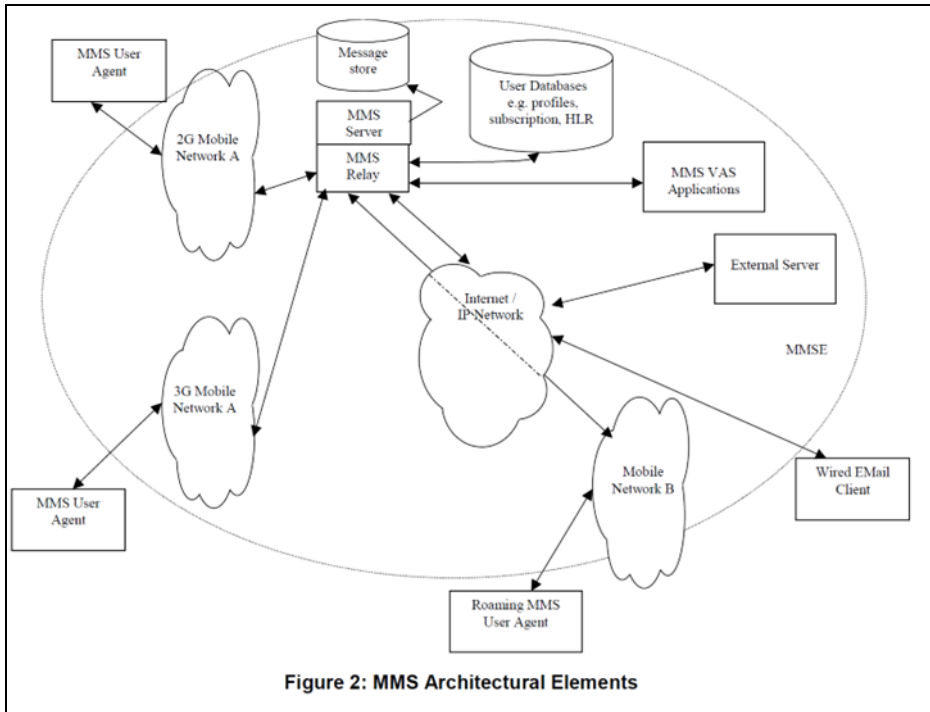
101. Additional details regarding the MMS Relay/Server and the method for operating it are provided below with reference to [1a]/[15a]-[1d4]/[15d3] *infra*.

102. For the above reasons, TS-23.140 renders obvious [1pre]/[15pre].

- (b) *[1a]: a transport services stack to maintain a respective secure message link through an Internet network between the message link server and a respective device link agent on each of a plurality of wireless end-user devices, / [15a]: maintaining a respective secure message link through an Internet network between the message link server and a respective device link agent on each of a plurality of wireless end-user devices,*

103. TS-23.140 renders obvious a transport services stack (e.g., a transport layer security (TLS)-based transport protocol) to maintain a respective secure message link (e.g., secure links) through an Internet network between the message link server (e.g., Message Relay/Server) and a respective device link agent (e.g., MMS User Agent) on each of a plurality of wireless end-user devices (e.g., wireless client devices).

104. In more detail, and as shown in FIG. 2 (reproduced below), the MMS Relay/Server communicates via an Internet network with MMS User Agents that execute on each UE (wireless end-user device).



SAMSUNG-1004, p. 17 (“Figure 2 shows that multimedia messaging may encompass many different network types ... provided by the Internet protocol and its associated set of messaging protocols” that “enables messaging in 2G and 3G wireless networks to be compatible with messaging systems found on the Internet.”).

105. Similarly, as disclosed in SAMSUNG-1010, FIG. 3 depicts a similar MMS communication between an MMS client and the MMS Relay/Server over the Internet and a wireless network.

106. The MMS User Agent is a device link agent because it facilitates transmission and receipt of multimedia messages (MMs) between different devices (e.g., MMS User Agents executing on other UEs, MMS Relay/Server, MMS VAS

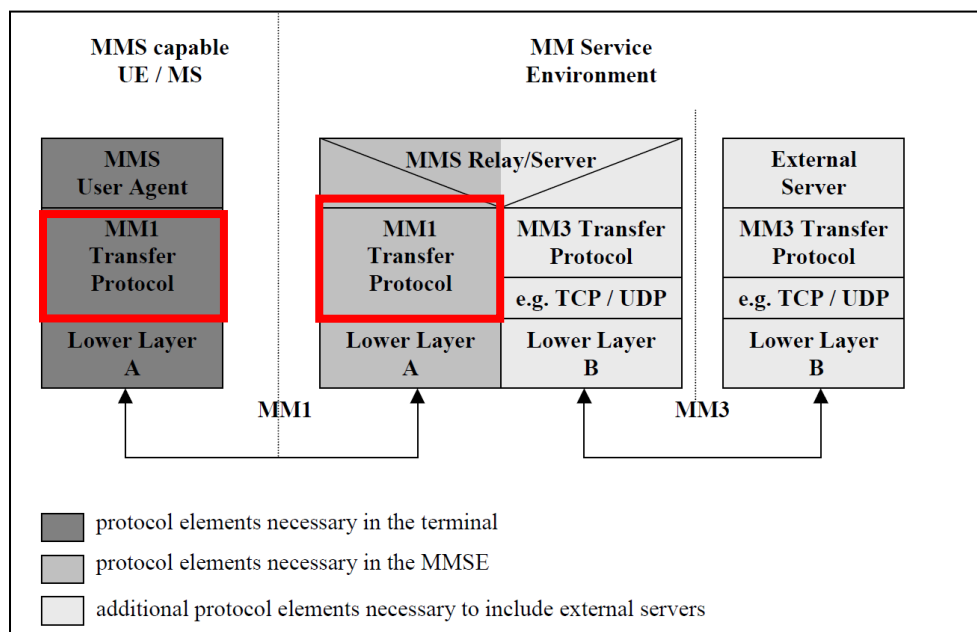
Applications) that are linked together over a communication network (as shown in FIG. 2 above). SAMSUNG-1004, pp. 19-20, §5.1.1 (describing operations of MMS User Agent); *see also* SAMSUNG-1036, 2, Fig. 1 (showing multiple MMS User Agents in communication with MMS Relay/Servers).

107. The '192 patent does not specifically describe the structure of the transport services stack that facilitate the claimed communication link, and instead, functionally describes this claim element. For example, the '192 specification states that the “transport services stack” facilitates communications between the client and server devices and can be “implemented with standard secure or open Internet networking protocols, such as TLS or TCP.” *See* SAMSUNG-1001, 89:24-41, 90:34-50.

108. A POSITA would have understood TLS to refer to Transport Layer Security, and TCP as referring to Transmission Control Protocol. *See* SAMSUNG-1027³, SAMSUNG-1028.

³ SAMSUNG-1027 refers to IETF RFC 793, Transmission Control Protocol (Sept. 1981) (available at <https://www.ietf.org/rfc/rfc793.txt>) and SAMSUNG-1028 refers to The TLS Protocol Version v 1.0 (Jan. 1999) (available at <https://datatracker.ietf.org/doc/html/rfc2246>).

109. Consistent with this disclosure, TS-23.140 also discloses or renders obvious a transport services stack in the form of a transfer protocol and associated functionality (that uses TCP and TLS) that maintains a secure message link through an Internet Network between the message link server (e.g., MMS Relay/Server) and wireless end-user devices (e.g., UEs using the same protocol). For example, TS-23.140 discloses that MMS shall “make use of the protocol framework outlined in figure 4” pursuant to which the MMS User Agent communicates with MMS Relay/Server. SAMSUNG-1004, §6.2, pp. 24-25.



SAMSUNG-1004, §6.2, pp. 24-25, FIG. 4 (providing protocol framework to provide MMS; listing MM1 transfer protocol as part of MMS Relay/Server).

110. As shown, TS-23.140 discloses that the MM1 Transfer Protocol is implemented both at the MMS Relay/Server and the MMS UE. *See id.* A POSITA

would have understood that the MM1 Transfer Protocol is a transport services stack that facilitates transmission/transport of network communications between the MMS Relay/Server and network elements (e.g., the MMS User Agent, MMS VAS Applications).

111. A POSITA would have understood that the MM1 Transfer Protocol is transport services stack given its well-known description in the art as the MM1 Transfer “protocol **stack**” that provides a “transport mechanism” using the HTTP/TCP/IP. SAMSUNG-1013.

112. Indeed, consistent with this understanding, TS-23.140 contemplates “WAP/OMA implementation” for the “MM1 Transfer Protocol” and incorporates by reference Open Mobile Alliance specifications, such as those provided in SAMSUNG-1011. See SAMSUNG-1004, pp. 13, 162. This incorporated document explains that “[s]ince MMS is an application-level protocol framework to enable messaging transactions, a device implementing OMA MMS must have either of WAP WSP stack or HTTP/TCP/IP stack.” SAMSUNG-1011, pp. 11.

113. Further still, the same suite of OMA specifications related to MMS explains that “TLS” or “transport layer security protocol provides for secure data transmission between the MMS Client and the MMS Proxy-Relay in architectural configurations that employ HTTP based protocol stacks for MMSM implementation.” SAMSUNG-1010, p. 22.

114. Given these disclosures and the well-known use of TLS to secure/encrypt network communications, a POSITA would have understood or found obvious that the transport services stack in the MMS Relay/Server would use TLS for securing the communication link between the MMS User Agent and the MMS Relay/Server. This is further corroborated by SAMSUNG-1010, which contemplates “encrypt[ing] the communication channel between the client and the backend MMS server with some security protocols, e.g., WTLS (Wireless Transport Layer Security.” SAMSUNG-1014, [0017].

115. A POSITA would have further understood or found obvious that in the MMS environment, multiple MMS User Agents (and their associated UEs) are in communication with the MMS Relay/Server, such that a separate secure TLS-based link is maintained between each MMS User Agent and the MMS Relay/Server. This is further corroborated by multiple references. As a starting point, a POSITA would have known that, in the relevant timeframe, TLS or SSL-based communications are expensive to form and use keep-alive and session-resumption methods to maintain already-formed/established connections (and thereby avoid cost and time to re-establish such connections). Indeed, this is corroborated by Houghton, which discloses connections between push servers, akin to MMS Relay/Server, and push clients, akin to MMS User Agents, being “persistent managed, tested and configured” and use periodic messages to prevent “time expir[ing]” the connection. SAMSUNG-

1007, 23, 17. For example, Mostafa discloses and shows (as reproduced below) communication links formed and maintained between MMS Relay/Server and multiple terminals (and their respective MMS User Agents):

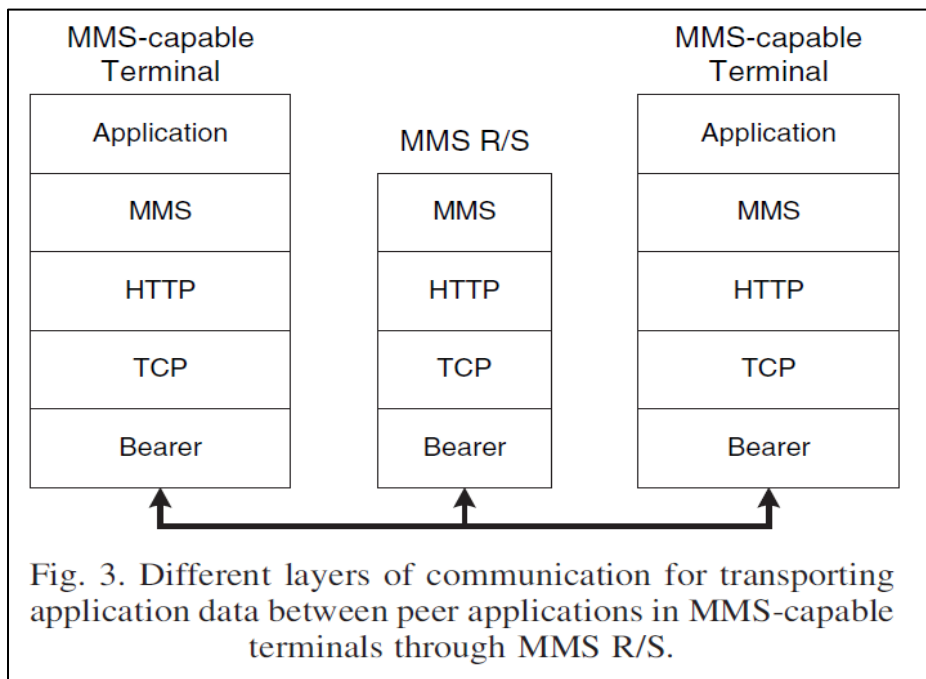


Fig. 3. Different layers of communication for transporting application data between peer applications in MMS-capable terminals through MMS R/S.

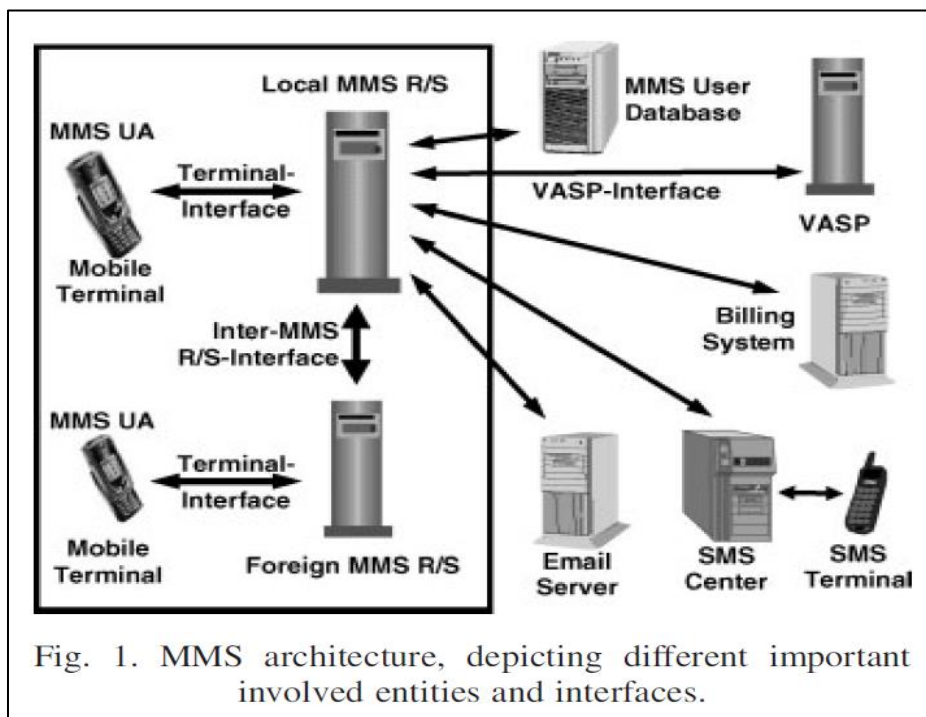
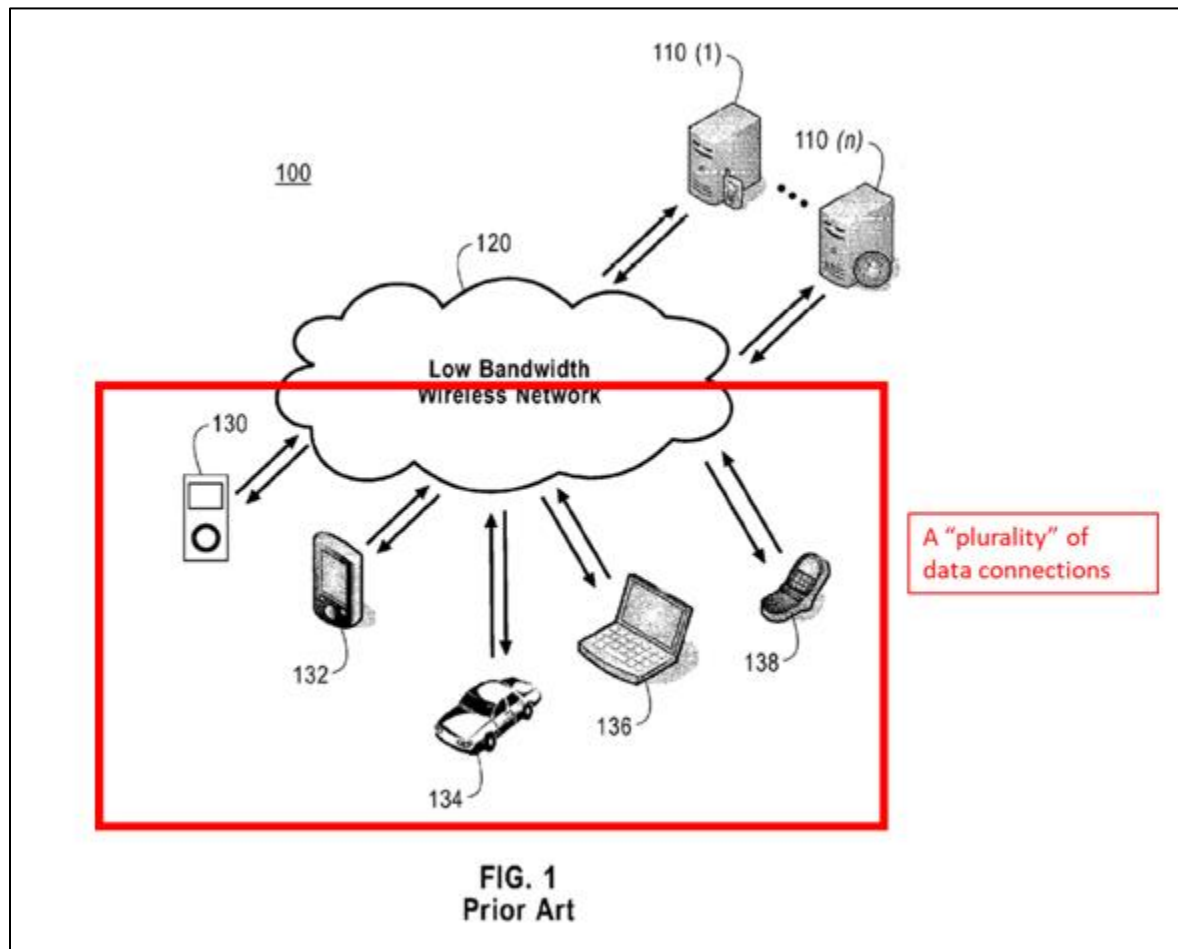


Fig. 1. MMS architecture, depicting different important involved entities and interfaces.

SAMSUNG-1036, FIGS. 1-3, pp. 2-4. Munson further corroborates that such persistently, maintained communication links are formed between multiple devices and the servers, as shown below. SAMSUNG-1017, FIG. 1, [0007]-[0008] (showing plurality of data connections with multiple end-user devices).



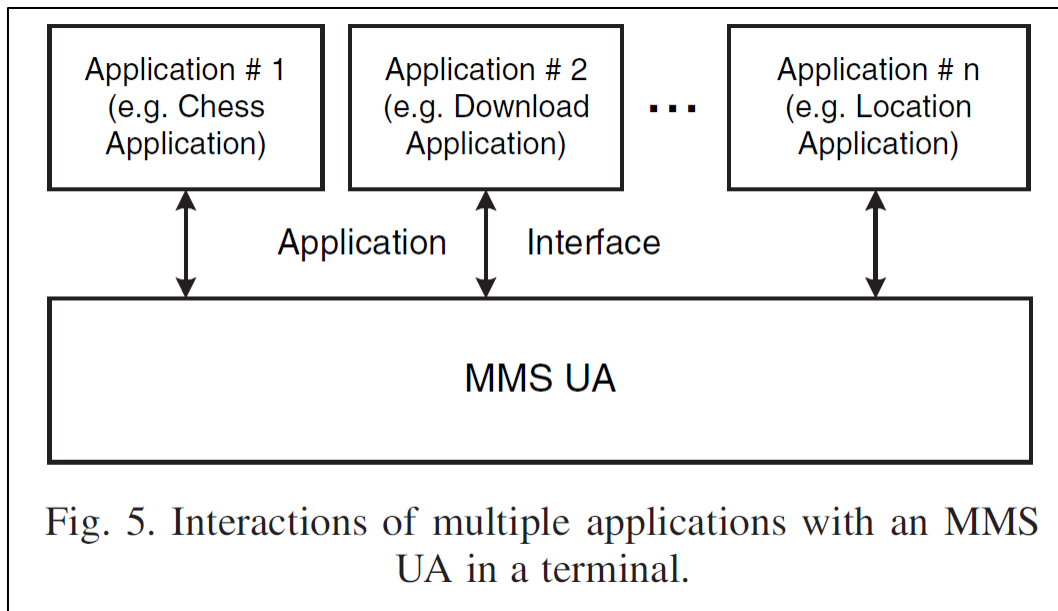
116. For the above reasons, TS-23.140 renders obvious [1a]/[15a].

- (c) [1b]: each of the wireless end-user devices comprising multiple software components authorized to receive and process data from secure message link messages received via a device link agent on that device; / [15b]: each of the wireless end-user devices comprising multiple

software components authorized to receive and process data from secure message link messages received via a device link agent on that device;

117. TS-23.140 renders obvious that each of the wireless end-user devices (e.g., wireless client devices) include multiple software components (e.g., multiple applications) authorized to receive and process data (e.g., application-specific data intended for the application identified by the identifier and assuming user agent supports transport of application-specific data) from secure message link messages received via a device link agent on that device.

118. In more detail, TS-23.140 discloses that applications, such as those that are downloaded on a mobile phone, can use MMS to “transport data specific to ***applications***” and MMS communication can include “transport data specific to applications between two MMS User Agents or an MMS User Agent and an MMS VAS Application.” SAMSUNG-1004, §7.1.18, pp. 54-55. Thus, TS-23.140 discloses that each UE (wireless end-user device) includes multiple software applications (software components). SAMSUNG-1036, 2-4 (showing and listing multiple applications in UEs that receive application data via MMS), FIG. 5 below.



119. A POSITA would have understood or found obvious that each such application has to be authorized to receive and process messages via MMS. TS-23.140 discloses that “[a]pplications that want to transport data specific to applications other than MMS will initially ***need to register with the appropriate MMS User Agent or MMS VAS Application.***” SAMSUNG-1004, §7.1.18, pp. 54-55. “During this registration process the application provisions an MMS User Agent or an MMS VAS Application with its application identification value and may negotiate with the MMS User Agent or MMS VAS Application the details (amount and format) of information to be exchanged between the two entities.” *Id.*

120. Additionally, the MMS User Agent can indicate the “Terminal Capability” toward its recipient MMS Relay/Server by transmitting “a set of information describing the terminal’s capabilities, including a binary flag indicating

whether or not the terminal supports Application data.” SAMSUNG-1004, §7.1.3.1, p. 30.

121. Subsequently, a message including application data for an intended application is only delivered to that application upon determining that the MMS User Agent has the capability of supporting application data and the terminal includes the intended application for the receipt of such application data. *Id.*, pp. 30, 54-55 (“If the MMS Relay/Server finds from the recipient MMS User Agent’s capability indication ... that the recipient MMS User Agent supports transport of application data, the MMS Relay/Server ... shall pass on the destination application identifier, the 'reply-path' identifier (if present) and the additional application/implementation specific control information (if present) unaltered.”; “If the destination application does not reside on the receiving MMS User Agent or on the USIM or MMS VAS Application, the MMS User Agent or MMS VAS Application shall discard the corresponding abstract message.”). From these disclosures a POSITA would have understood that such applications are **authorized** to receive messages including application data.

122. A POSITA would have further understood that application data received by a particular destination application would be processed by that application to, e.g., facilitate display of that data or perform operations with respect

to that data. SAMSUNG-1004, § 7.1.18.2.2, p. 56 (contemplating “further handling and processing of the” received application data “by the destination application”).

123. Further still, because the above-described communication is facilitated between MMS User Agents and/or an MMS User Agent and an MMS VAS Application, and such communications occur via the MMS Relay/Server (as described above in [1a]/[15a]), a POSITA would have understood that the messages including the application data are received via the MMS User Agent (device link agent) on the particular terminal/device. SAMSUNG-1004, §7.1.18, pp. 54-55 (“Based on the negotiated details upon application registration process an application may trigger an MMS User Agent or an MMS VAS Application to submit certain abstract messages. Upon triggering an MMS User Agent or an MMS VAS Application to send an abstract message the MMS User Agent or MMS VAS Application may receive information from the application. ***The MMS User Agent or MMS VAS Application may insert this information in both the information elements and/or payload (if present) of the abstract message.... Abstract messages that are sent by an MMS User Agent or an MMS VAS Application on behalf of an originating application shall contain a destination application identifier.***”); *id.*, 55 (“Upon reception of an abstract message containing a destination application identifier (it can either be the MM1_notification.REQ, MM1_retrieve.RES or MM7_deliver.REQ transactions), ***the receiving MMS User Agent or MMS VAS***

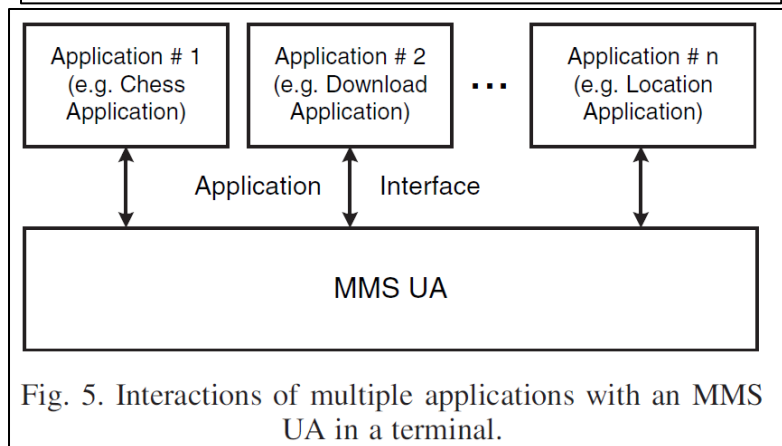
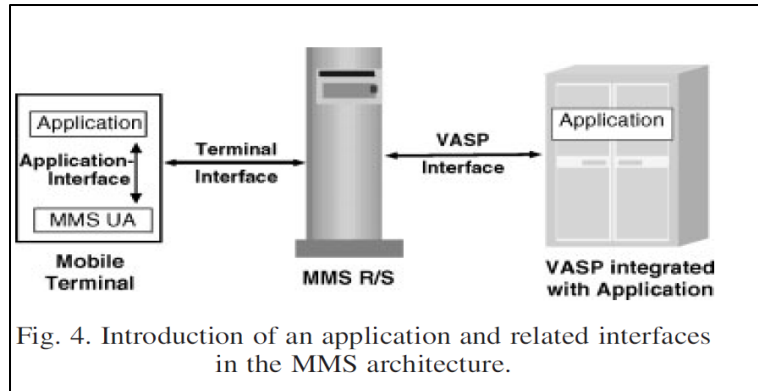
Application shall first check if the destination application resides on it” and if so, the MMS User Agent shall immediately route the received MMS information on to the destination application that is referred to from the destination application identifier”).

124. Additionally, because a secure message link is used for the communication between the MMS Relay/Server and the MMS User Agent (as described with reference to [1a]/[15a] (which I have incorporated by reference here), a POSITA would have understood that the messages including the application data are “secure message link messages.”

125. A POSITA would have found it obvious, in view of TS-23.140’s disclosures, that the above-described procedure of delivery of application-specific data to an application could be performed for multiple applications that are intended to send or receive application-specific data. For example, TS-23.140 provides examples of applications (e.g., chess application, messaging/discussion applications, news applications, weather applications) that use MMS for application specific messaging. See, e.g., SAMSUNG-1004, p. 55 (“The additional application/implementation specific control information shall be used for all future needs that are not supported by the application identifier of the destination application and the identifier of the originating application, such as specifying a particular logical channel in the application addressing method (e.g., ‘discussion

thread #05') or distinguishing between multiple instances of the same application (e.g., 'chess application #02'); p. 14 (explaining that VAS application are "applications providing value added services (e.g. news service or weather forecasts) to MMS users"), pp. 54-55 ("Applications that ... transport application specific data using MMS ... reside on an MMS User Agent.... Applications that want to transport data specific to applications other than MMS will initially need to register with the appropriate MMS User Agent"; describing different types of applications, e.g., chess and messaging applications"). Indeed, MMS provided a well-established infrastructure for application messaging and as described above, applications would naturally use MMS messaging to facilitate mobile content delivery (e.g., delivery of news, weather, or application related messaging).

126. Further still, it was well-known for mobile/MMS-capable user devices/terminals to include multiple applications and for such applications to send and receive application-specific data using MMS. For example, Mostafa discloses and depicts (below) multiple applications, e.g., chess, download, and location applications, that interface with the MMS User Agent (UA) to facilitate sending and receiving of application data via MMS. *See SAMSUNG-1036*, 3-4 (pp. 732-733, FIGS. 3-5).



See also *id.* (“As for example, a chess application within a terminal may offer its user to play chess with a user of a different terminal having the same application. If both the terminals are also MMS-capable, the application may use an MM to convey the information about each move made by a user on the chessboard to the other user. Other interactive games (e.g., golf, snooker) in a mobile terminal can also use MMS as the transport similarly. A location service could also be based on this new MMS feature, so that a location server, acting as an application over a VASP, can provide location specific information to a terminal. Moreover, the feature is also useful for downloading any content (e.g., a media object, a new game, a new plug-in of an existing game, an application, media codec) from a VASP. Furthermore, a

configuration application could use the feature to get a new or an updated configuration of a terminal (e.g., connectivity information for any network service like MMS, browsing) from a network. These are just few examples, and the feature can be used for many other purposes.”).

127. Thus, a POSITA would have understood or found obvious that the UEs (i.e., the wireless end-user devices) would include multiple applications (i.e., software components) that are capable of receiving and processing application data received from the MMS Server/Relay.

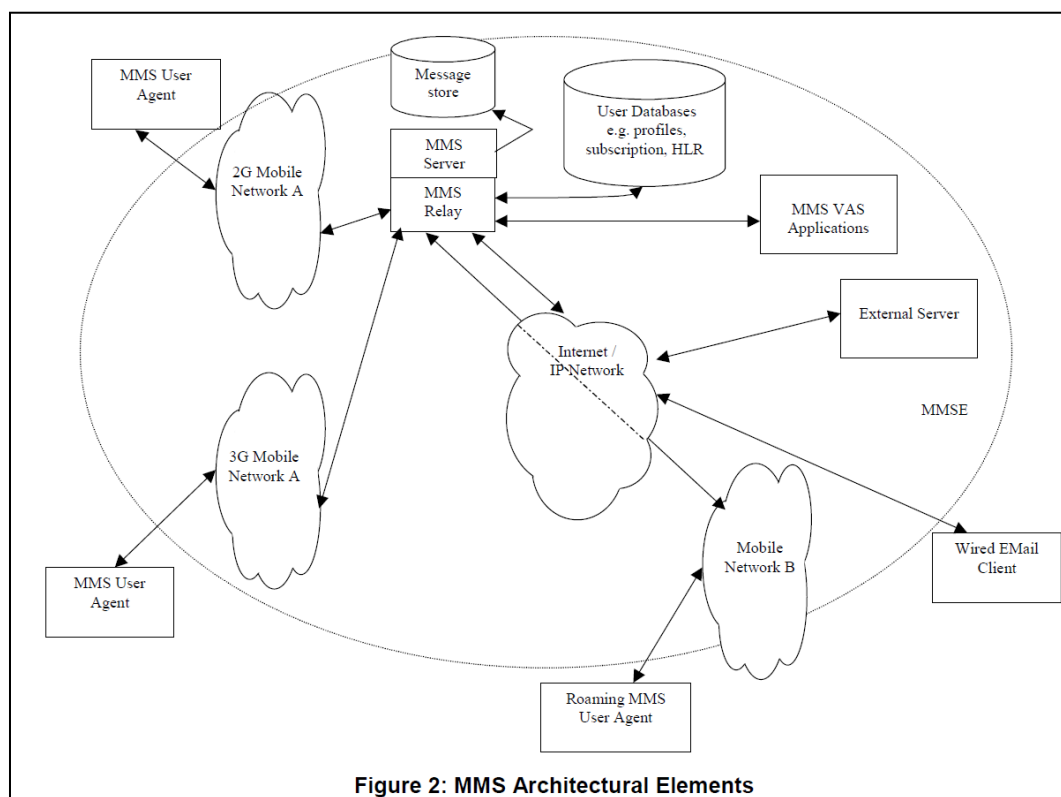
128. For the above reasons, TS-23.140 renders obvious [1b]/[15b].

(d) *[1c1]: an interface to a network to receive network element messages from a plurality of network elements, / [15c1]: receiving network element messages from a plurality of network elements,*

129. TS-23.140 discloses or renders obvious an interface (e.g., multiple interfaces including MM1, MM3, MM7) to a network to receive network element messages (e.g., messages including application-specific data and other addressing data) from a plurality of network elements (e.g., MMS User Agents resident on MMS client devices, MMS VAS Applications).

130. In more detail, the following figure shows an MMS network environment in which communications are facilitated between various MMS User

Agents, the MMS Server/Relay, MMS VAS Applications, and one or more external servers over various networks.

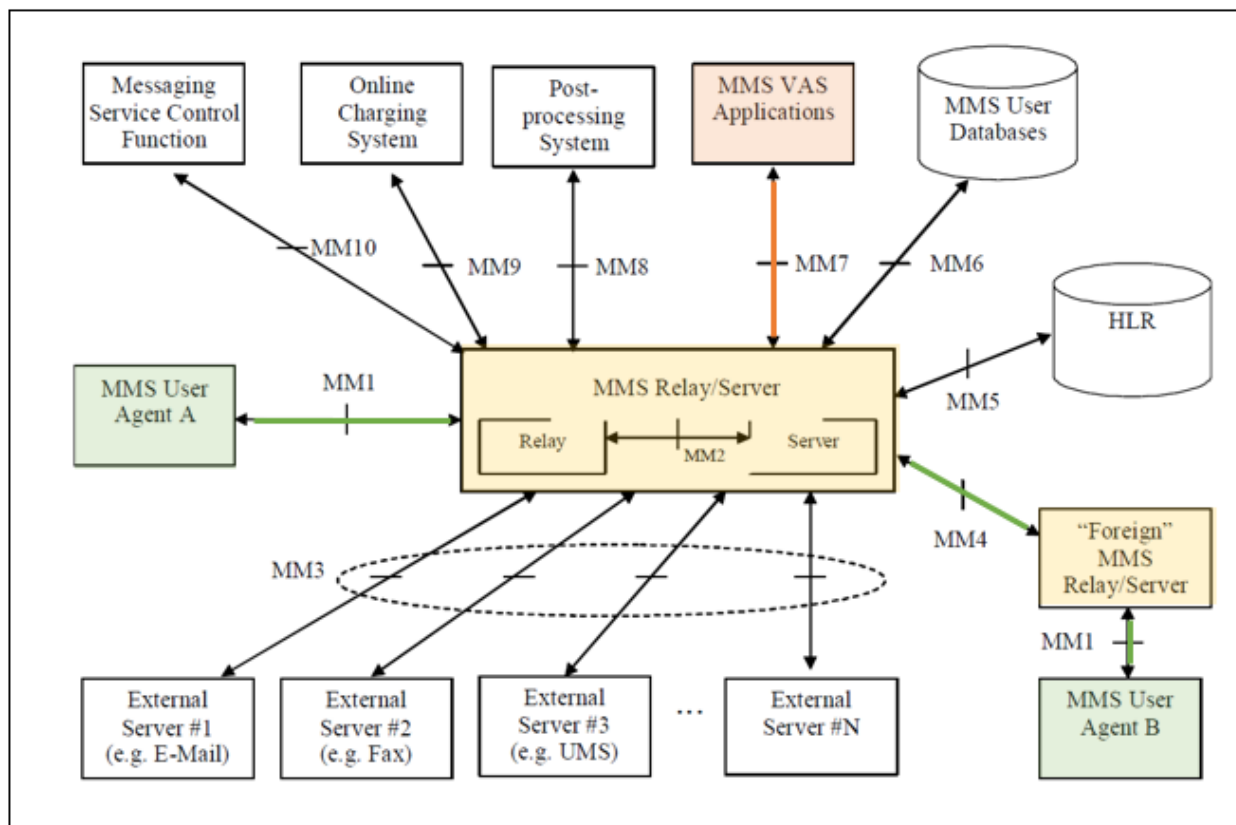


SAMSUNG-1004, § 4.2, Fig. 2, p. 17 (describing the above MMS environment as “a collection of MMS-specific network elements.”).

131. And, as described with reference to [1a]/[15a] and [1b]/[15b], the MMS Relay/Server receives messages from one or more of these network elements (i.e., network element messages), such as one or more MMS User Agents and MMS VAS Applications.

132. TS-23.140 further discloses various “interfaces” facilitating network communication of such network element messages. SAMSUNG-1004,

§6.1, pp. 23-24. For example, the following figure shows the various interfaces facilitating communications between MMS User Agents, MMS VAS Applications, and MMS Relay/Server, including the MM1, MM2, MM7, and MM4 interfaces:



SAMSUNG-1004, §6.1, Fig. 3, pp. 23-24.

133. Specifically, TS-23.140 that (1) the MM1 interface is the “reference point between the MMS User Agent and the MMS Relay/Server,” (2) the MM4 interface is the “reference point between the MMS Relay/Server and another MMS Relay/Server that is within another MMSE,” and (3) MM7 interface is “the reference point between the MMS Relay/Server and MMS VAS Applications.” SAMSUNG-1004, §6.1, pp. 23-24.

134. As shown in the above Figures, the interfaces are to various networks, including 2G/3G mobile networks, and IP/internet networks. SAMSUNG-1004, Fig. 2.

135. Additionally, as described with reference to [1b]/[15b], applications that are registered with MMS User Agents or MMS VAS application can transmit application data to applications registered with other MMS user agents and VAS applications, in which case, the communications are facilitated via the MMS Server/Relay. See SAMSUNG-1004, §7.1.18, pp. 54-55 (contemplating delivery of application specification data to an application via the user agent, using that application's identifier).

136. For the above reasons, TS-23.140 renders obvious [1c1]/[15c1].

(e) *[1c2]: the received network element messages comprising respective message content and requests for delivery of the respective message content to respective wireless end-user devices, the respective message content including data for, and an identification of, a respective one of the authorized software components; / [15c2]: the received network element messages comprising respective message content and requests for delivery of the respective message content to respective wireless end-user devices, the respective message content including data for, and an identification of, a respective one of the authorized software components;*

137. TS-23.140 renders obvious that the received network element messages (e.g., abstract messages received from applications resident on the MMS User Agent

or the MMS VAS Applications) comprise respective message content, including data for (e.g., application data and/or other control information) and identification of a respective one of the authorized software (e.g., identification of destination application(s)), and requests for delivery of the respective message content to respective wireless end-user devices.

138. In more detail, and as described with reference to [1a]/[15a] and [1b]/[15b], TS-23.140 contemplates use of MMS to facilitate transport of application data from one device (e.g., server or terminal) and its associated agent (MMS User Agent or MMS VAS application) to another device (server or terminal) and its associated agent (MMS User Agent or MMS VAS Application). SAMSUNG-1004, §7.1.18.1, pp. 54-55.

139. Specifically, TS-23.140 discloses that the transmission of application data is facilitated by an originating application triggering the MMS User Agent or MMS VAS Application to which it is registered to send an abstract message (that includes this application data) to a destination application. SAMSUNG-1004, §7.1.18.1, pp. 54-55 (“Based on the negotiated details upon application registration process *an application may trigger an MMS User Agent or an MMS VAS Application to submit certain abstract messages*. Upon triggering an MMS User Agent or an MMS VAS Application to send an abstract message *the MMS User Agent or MMS VAS Application may receive information from the application*.”).

The MMS User Agent or MMS VAS Application *may insert this information in both the information elements and/or payload (if present) of the abstract message.*”), pp. 55-56 (application data can include “application/implementation specific control information”); §7.1.18, pp. 54-55, §3.1, p. 14 (defining abstract message as “information which is transferred between two MMS entities used to convey an MM and/or associated control information between these two entities”).

140. The abstract message further includes the destination application’s application identifier. *Id.* (“**Abstract messages that are sent by an MMS User Agent** or an MMS VAS Application on behalf of an originating application **shall contain a destination application identifier.**”).

141. Additionally, given that abstract messages are sent by an originating MMS User Agent to MMS Relay/Server for delivery to an MMS User Agent with which the destination application is registered, a POSITA would have understood or found obvious that the transmitted message comprises a request for delivery of the respective message content to a respective wireless end-user device (including the destination application resident on that device). SAMSUNG-1004, §7.1.18, pp. 54-55 (“If the MMS Relay/Server finds from the recipient MMS User Agent’s capability indication (see clause 7.1.3.1) that the recipient MMS User Agent supports transport of application data, **the MMS Relay/Server . . . shall pass on the destination application identifier, the 'reply-path' identifier (if present) and the**

additional application/implementation specific control information (if present)
unaltered. Upon reception of an abstract message containing a destination application identifier (it can either be the ***MM1_notification.REQ***, ***MM1_retrieve.RES*** or ***MM7_deliver.REQ*** transactions), the receiving MMS User Agent or MMS VAS Application shall first check if the destination application resides on it.”); FIG. 6 (showing typical message exchange between MMS User Agents via MMS Server/Relay); TS-23.140 (SAMSUNG-1004), §7.1.2.1, p. 28; TS-23.140, § 7.1.3, p. 29; *see also* [1c1] above (further explaining and describing receipt of network element messages, including above-described messages exchanged between applications).

142. A POSITA would have understood or found obvious that the recipient/destination application would be an authorized/registered application (*see* [1b] *supra*) because applications intending to send/receive application data “need to register with the appropriate MMS User Agent” using their “application identification value” and then, upon message receipt, the recipient application would then need to be identified by MMS User Agent using its identification value as being resident on the user device before the message is transmitted to this application. *See* SAMSUNG-1004, 55-56.

143. Moreover, a POSITA would have understood in view of TS-23.140’s disclosures (*see* § VII.A *supra* and [1pre]-[1b] discussion above) MMS Relay/Server

can interface with multiple devices and their respective MMS User Agents, such that it transmits messages to multiple MMS User Agents/devices for delivery to their respective applications). As such, a POSITA would have understood or found obvious that MMS Relay/Server receives requests for delivery of message content to multiple wireless end-user devices in the MMS environment.

144. For the above reasons, TS-23.140 renders obvious [1c2]/[15c2].

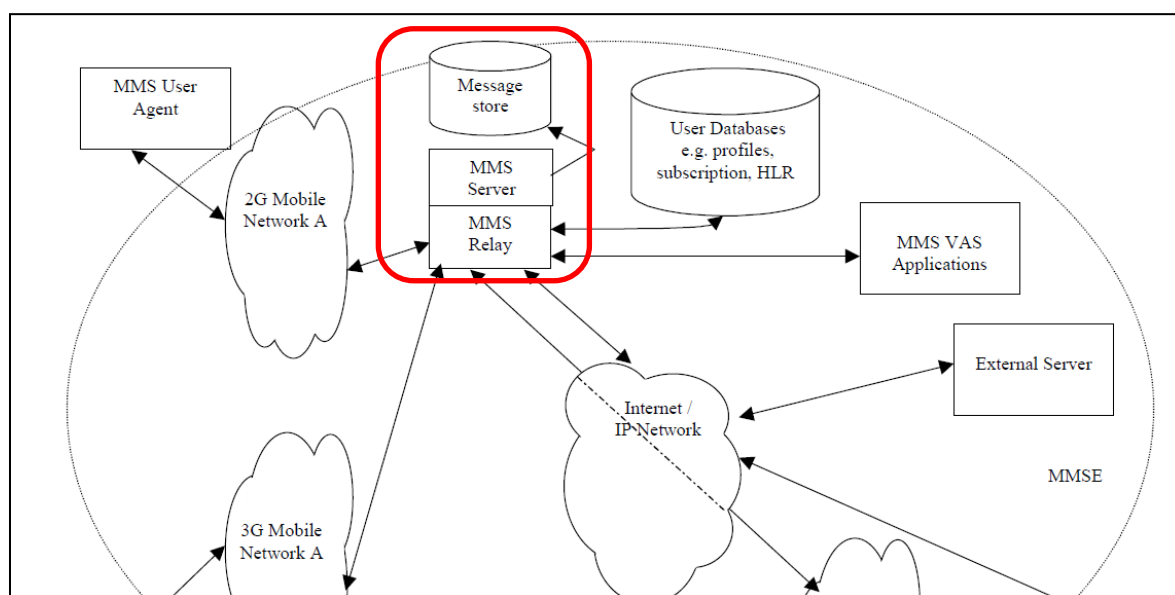
- (f) *[1d1]-[1d2]: a message buffer system including a memory and logic, the memory to buffer content from the received network element messages for which delivery is requested to a given one of the wireless end-user devices, / [15d1]: buffering content from the received network element messages for which delivery is requested to a given one of the wireless end-user devices;*

145. TS-23.140 renders obvious a message buffer system including a memory that buffers content from the received network element messages (e.g., message store for storing the received message from, e.g., an application associated with the originating MMS User Agent or VAS application, for which delivery is requested to a given one of the wireless end-user devices (e.g., an application running on a client device associated with the MMS User Agent for that device).

146. As described with reference to [1c1]-[1c2] (which I have incorporated by reference here), the network element messages are received from one or more network elements by the MMS Relay/Server, and delivery of these message is

requested to an application resident on a user device (and associated with its MMS User Agent).

147. Moreover, the “**MMS Relay/Server is responsible for storage and handling of incoming and outgoing messages** and for the transfer of messages between different messaging systems.” SAMSUNG-1004, § 4.2, p. 17. Fig. 2 of TS.123.140 shows a message store coupled to the MMS Server/Relay:



See also SAMSUNG-1004, § 5.2, p. 21 (describing “temporary storage of messages” received and send by MMS User Agents; facilitating transport of application data).

148. “Upon reception of an MM from an originator MMS User Agent,” the “originator MMS Relay/Server . . . **shall retain the MM until the earliest desired time of delivery**, if the optional feature of earliest time of delivery is supported by the originator MMS Relay/Server. . . .” SAMSUNG-1004, § 7.1.1, 7.1.2, pp. 26-28.

149. Similarly, “[u]pon reception of an MM the recipient MMS Relay/Server [which can be the same as the originator MMS User Agent],” this MMS Relay /Server can (1) **store the MM at least until one of** “the associated time of expiry is reached, the MM is delivered, the recipient MMS User Agent requests the MM to be routed forward or the MM is rejected,” and (2) “shall generate a notification to the recipient MMS User Agent.” *Id.*

150. TS-23.140 further describes “persistent storage of messages” in a “Persistent Network-Based Storage” (called MMBox) **which is associated with “the MMS Relay/Server and into which Multimedia Messages (MMs) may be stored, retrieved, and deleted.”** SAMSUNG-1004, § 5.2.1, pp. 21-22. As such, when a multimedia message is received from a MMS User Agent, the MMS Relay/Server **“may store the MM into an MMBox . . .”** *Id.*, pp. 26-28.

151. Additionally, as described with reference to [1d3]-[1d4] (which I have incorporated by reference here), the MMS Relay/Server includes a message buffer system with logic that facilitate delivery of messages upon occurrence of one or more message delivery triggers.

152. For the above reasons, TS-23.140 renders obvious [1d1]-[1d2]/[15d1].

- (g) [1d3]: the logic to determine when one of a plurality of message delivery triggers for the given one of the wireless end-user devices has occurred, wherein for at least some of the received network element messages, the

receipt of such a message by the message buffer system is not a message delivery trigger, and for at least one of the message delivery triggers, the trigger is an occurrence of an asynchronous event with time-critical messaging needs, and / [15d2]: determining when one of a plurality of message delivery triggers for the given one of the wireless end-user devices has occurred, wherein for at least some of the received network element messages, the receipt of such a message is not a message delivery trigger, and for at least one of the message delivery triggers, the trigger is an occurrence of an asynchronous event with time-critical messaging needs; and

153. As explained below, TS-23.140 renders obvious the message buffer system including logic that determines when one of a plurality of message delivery triggers for the given one of the wireless end-user devices has occurred (e.g., user request for a message, expiration of a message expiry period, determination of user settings or recommendations specifying retrieval mode to use for received message, conformity of message to MMS User Agent specification of a size restriction).

154. Additionally, and as further explained below, TS-23.140 renders obvious that the receipt of a message is not a message delivery trigger and one of the message delivery triggers is an occurrence of an asynchronous event with time-critical messaging needs (e.g., a user request for a received message).

155. In more detail, TS-23.140 discloses multiple message delivery triggers for transmitting the stored message data to a MMS User Agent for a UE that includes the destination application. For example, the message is not delivered by the MMS

Relay/Server to a destination MMS User Agent (resident on a user device) until one or more of the following triggers have occurred:

- The MMS Relay/Server has sent a notification to the recipient User Agent (TS-23.140, §7.1.2 (“Upon reception of an MM the recipient ***MMS Relay/Server ... shall generate a notification to the recipient MMS User Agent.***”)).
- The recipient MMS User Agent requests retrieval of the message upon receipt of the notification from the MMS Relay/Server—within a message expiry period. See SAMSUNG-1004, §7.1.2.1 (“In a response to the notification the MMS User Agent shall be able to ... retrieve the MM, either immediately or at a later time,” and either automatically or manually, where the “retrieval mode” is based on “the user settings in the terminal or the recommendation carried in the MM notification”).
- The specified deferred delivery period is met (as in the case where the ***recipient MMS User Agent requests deferred delivery of the message in the retrieve message request***). See SAMSUNG-1004, §7.1.2.1 (“With the MM notification the recipient MMS User Agent shall receive a message reference that can be used for retrieving the MM from the recipient MMS Relay/Server. ***The message reference that is***

conveyed in a notification shall at least be valid throughout the message expiry period, till the successful retrieval of the MM or until the MM was rejected.”).

- Recipient MMS User Agent is available (i.e., in network coverage). See SAMSUNG-1004, §7.1.3 (“Upon retrieval request the recipient MMS Relay/Server shall ... store messages in the network until the *recipient MMS User Agent becomes reachable (e.g. user moves back into coverage, switches MMS User Agent on)* or until the MM expires.”).
- The message conforms to the message size limitations specified in the message retrieval request. See TS-23.140, §7.1.3 (“Within a retrieval request *the recipient MMS User Agent may indicate a size restriction of the returned MM (i.e., maximum size) that the MMS Relay/Server is to use in processing the retrieval request*”; the message should not be delivered “unless the size restriction set by the MMS User Agent is met”).

156. Because the MMS Relay/Server does not deliver the message to the recipient MMS User Agent (corresponding to a particular user terminal) until the above conditions/triggers are met, a POSITA would have understood or found obvious that the MMS Relay/Server includes logic that is configured to determine

when one (or more) of these multiple message delivery triggers for the particular terminal (given one of the wireless end-user devices) has occurred. For example, a POSITA would have recognized that the MMS Relay/Server would be programmed with logic that would monitor the occurrence of these conditions and upon detecting the occurrence of one or more such conditions, prepare and send the stored messages.

157. Moreover, given the above disclosures of the different message delivery triggers, a POSITA would have found obvious that the receipt of a message alone would not trigger message delivery, particularly considering the other conditions/triggers that can be implemented and which would then need to be satisfied before the message is deliver.

158. Also, in view of the above disclosures about the different message delivery triggers, a POSITA would have understood or found obvious that one or more of the above-recited triggers is an occurrence of “an asynchronous event with time-critical messaging needs,” consistent with how this phrase is described in the ’192 patent’s specification. Specifically, the ’192 specification states in relevant part as follows:

In some embodiments, the transmission trigger is based on waiting for an amount of service usage, and also including a minimum transmission rate that triggers a transmission according to one or more of the following parameters: a maximum time between transmissions

*clock to keep the service processor 115 in communication with the service controller 122 when little or no service usage is occurring, a polling request of some kind from the service controller 122, a response to a service controller heartbeat, a transmission generated by a service verification error event, **or a transmission generated by some other asynchronous event with time critical service processor 115 (or service controller 122) messaging needs, such as a transaction or service billing event or a user request.***

SAMSUNG-1001, 38:50-63. Thus, a POSITA would have understood that in the context of the '192 Patent, a trigger that is “an occurrence of asynchronous event with time-critical messaging needs” encompasses a “user request” for message delivery. *See id.*

159. Consistent with this disclosure, a POSITA would have understood that TS-23.140 discloses that the MMS User Agent can request delivery of the message received and stored by the recipient MMS User Agent, and that such requested delivery can be manual—i.e., based on user request. See, e.g., SAMSUNG-1004, §7.1.2.1, pp. 28-29 (“the MMS User Agent shall be able to ... retrieve the MM, either immediately or at a later time ..., **either manually or automatically** ...”); §8.1.5.1, p. 69 (“The recipient MMS User Agent shall issue an MM1_retrieve.REQ to the recipient MMS Relay/Server to initiate the retrieval process. The MMS

Relay/Server shall respond with an MM1_retrieve.RES, which contains MMs control information and the MM content.”); §5.1.1.1, p. 20 (in “manual mode,” the “user is made aware of the MM notification and ***is allowed to make a decision whether to download the MM or not***”).

160. For the above reasons, TS-23.140 renders obvious [1d3]/[15d2].

- (h) *[1d4]: upon determining that one of the message delivery triggers has occurred, the logic further to supply one or more messages comprising the buffered content to the transport services stack for delivery on the secure message link maintained between the transport services stack and a device link agent on the given one of the wireless end-user devices. / [15d3]: upon determining that one of the message delivery triggers has occurred, supplying one or more messages comprising the buffered content for delivery on the secure message link maintained between the message link server and a device link agent on the given one of the wireless end-user devices.*

161. TS-23.140 renders obvious that, upon determining that one of the message delivery triggers has occurred, the message buffer system includes logic for supplying one or more messages comprising the buffered content (e.g., stored message in the message store or MMSBox) to the transport services stack for delivery on the secure message link maintained between the transport services stack and a device link agent (e.g., MMS User Agent) on the given one of the wireless end-user devices.

162. As described in [1c2]-[1c3], TS-23.140 contemplates delivery of the stored message upon the occurrence of one or more of the above-described delivery triggers. See above analysis/evidence with reference to [1d1]-[1d3]/[15d1]-[15d2]; *e.g.*, SAMSUNG-1004, pp. 28-31, §7.1.3, 7.1.2.1.

163. By way of example, TS-23.140 discloses that “[u]pon retrieval request[,] the recipient MMS Relay/Server ... shall deliver the MM to the recipient MMS User Agent.” *Id.*, p. 29; *see also id.* (using size restriction in message retrieval request in process the request; further noting that “MMS User Agent shall be able to request retrieval” of the message “from the user’s MMBox”).

164. As described above ([1d1]-[1d3]/[15d1]-[15d2], which I have incorporated here), MMS Relay/Server delivers stored messages (in MMBox or another temporary storage) upon the occurrence of one or more of the above-described delivery triggers. SAMSUNG-1004, 28-31. A POSITA would have found obvious that MMS Relay/Server includes logic for performing such message delivery upon detecting occurrence of one or more of the above-described delivery triggers (again, as described above with reference to [1d3]/[15d2], which I have incorporated here).

165. Additionally, as described in [1a] (which I have incorporated by reference here), TS-23.140 discloses message delivery via the MMS Relay/Server’s transport services stack (*e.g.*, MM1 Transfer Protocol) that maintains secure

message link (e.g., TLS-based link) to a device link agent (e.g., a recipient MMS user agent) of a particular wireless end-user device (e.g., a UE/terminal corresponding to the recipient MMS User Agent). *See, e.g.*, SAMSUNG-1004, §6.2, p. 24, Fig. 4 (providing protocol framework to provide MMS; listing MM1 Transfer Protocol as part of MMS Relay/Server); SAMSUNG-1010, p. 22 (describing TLS-based link for data transmission between MMS User Agent and MMS Proxy-Relay in architectural configurations that employ HTTP based protocol stacks for MMSM implementation).

166. For the above reasons, TS-23.140 renders obvious [1d4]/[15d3].

2. Claim 5

167. TS-23.140 renders obvious the message link server (as described with reference to claims 1 and 15 above) and further renders obvious [5a]-[5c], as described below.

- (i) *[5a]: wherein the transport services stack is further to receive, over each of the respective secure message links, upload messages forwarded by the respective device link agents from at least a subset of the device software components,*

168. TS-23.140 renders obvious that the transport services stack is further to receive, over each of the respective secure message links, upload messages forwarded by the respective device link agents (e.g., one or more MMS User Agents)

from at least a subset of the device software components (e.g., one or more applications).

169. In more detail, TS-23.140 discloses that an application executing on a user device/UE (i.e., a device software component) can forward a message including application data (i.e., upload message) to the UE's MMS User Agent (i.e., the device link agent), which in turn can submit the message to the MMS Relay/Server. SAMSUNG-1004, §7.1.18, pp. 54-55; § 3.1, p. 14.

170. A POSITA would have understood that an application receiving application data using MMS can also send application data to an application resident on another device (e.g., another device with an MMS User Agent or a MMS VAS Application).

171. For example, TS-23.140 discloses that “an application may trigger an MMS User Agent or an MMS VAS Application to submit certain abstract messages. Upon triggering an MMS User Agent or an MMS VAS Application to send an abstract message the MMS User Agent or MMS VAS Application may receive information from the application. The MMS User Agent or MMS VAS Application may insert this information in both the information elements and/or payload (if present) of the abstract message.” *Id.*

172. Both the MMS User Agent and the MMS Relay/Server facilitate “transport of application data” (SAMSUNG-1004, §§5.1.1 and 5.1.2, pp. 19-21, 35

(describing transport of application data by MMS User Agent to another MMS User Agent via the MMS Relay/Server), 55 (transporting data between applications of two MMS User Agents). Moreover, as described with reference to [1a]/[15a] (which I have incorporated by reference here), the MMS Relay/Server includes the MM1 Transfer Protocol that facilitate receipt/transport of application data from the MMS capable UE/MS to the MMS Relay/Server. SAMSUNG-1004, §6.2, p. 24, Fig. 4. From these disclosures, a POSITA would have understood or found obvious that the transport stack of the MMS Relay/Server receives message including application data from an application via a particular MMS User Agent.

173. Moreover, in the MMS Environment contemplated by TS-23.140 (see FIGS. 1 and 2), a POSITA would have understood that multiple different devices, including end-user devices/terminal, would be able to use an MMS Relay/Server for transmission of data to multiple other devices/server/network elements.

174. In such an environment, a POSITA would have found it obvious for the MMS Relay/Server to receive upload messages (including application data) from multiple MMS User Agents and their respective applications requesting delivery of such data, and such communications would happen over the dedicated TLS-based links between each such MMS User Agent and the MMS Relay/Server (as described with reference to [1a]-[1b]/[15a]-[15b]).

- (j) *[5b]: each of the upload messages identifying a corresponding one of the network elements to which the device respective software component has requested delivery,*

175. TS-23.140 renders obvious each of the upload messages identifying a corresponding one of the network elements (e.g., recipient MMS User Agents on UEs) to which the device respective software component (e.g., originating application on a sending UE) has requested delivery.

176. As described in [5a] above, the application data is included in the upload message (i.e., the abstract message) from an originating MMS Agent and its associated device, and this abstract message further includes the “application identifier of the destination application.” SAMSUNG-1004, §7.1.18, pp. 54-55 (“***If the MMS Relay/Server finds from the recipient MMS User Agent’s capability indication (see clause 7.1.3.1) that the recipient MMS User Agent supports transport of application data, the MMS Relay/Server . . . shall pass on the destination application identifier,*** the ‘reply-path’ identifier (if present) and the additional application/implementation specific control information (if present) unaltered.”). Because the message includes an identifier of a destination application resident on a particular device/terminal, a POSITA would have understood that upload message identifies the corresponding device/terminal (i.e., the network element).

177. Additionally/alternatively, as with sending any multimedia messages (MMS), including MMs that include application data, the sender MMS User Agent “indicate[s] the address of the MM recipient.” *See* SAMSUNG-1004, TS-23.140, § 7.1.1, p. 26; *see also* p. 190 (showing data types included in messages sent over MM1 interface between MMS User Agent and the MMS Relay/Server, including “Recipient address”).

178. Indeed, because messages are transmitted from a MMS User Agent to a recipient MMS User Agent (or MMS VAS Application) (on another device/server), a POSITA would have found it obvious for the message to include an identifier or address for the device/server to which the message is being transmitted.

179. A POSITA would have understood or found obvious that routing messages to particular devices would be facilitated when including some device identification to facilitate that routing over a network. Indeed, TS-23.140 discloses that for MM1 communications (between MMS Relay/Server and MMS User Agents), the message must be addressed using three addresses: “the address of the MMS Relay/Server, the address of the recipient and the address of the originator.” SAMSUNG-1004, p. 57. “The recipient’s address can be a user’s address, a user’s terminal address, or a short code.” *Id.* Additional details about message addressing and formats are provided below (*id.*, p. 57):

The MMS User Agent and MMS Relay/Server shall support both E.164 (MSISDN) and RFC2822 addressing formats. The reference point MM1 should support a way to indicate the used address type to enable future extension. The encoding of the addressing is up to the corresponding MM1 implementation (cf. Annex B).

E.g. the originator MMS User Agent may specify each of the address fields in one of the following formats:

- 1) RFC 2822 address (FQDN or unqualified)
- 2) PLMN address: ["+" | "*" | "#"] [digit | "*" | "#"] ... ["/TYPE= PLMN"]
- 3) Other "/TYPE= "

The "/TYPE= " field specifies the address type. When PLMN format is used the type is optional. The "/TYPE= " convention provides flexibility for future enhancements.

When the "/TYPE=" qualifier is absent, the MMS Relay/Server should resolve potential ambiguities by applying the following logic to the address in the following order:

1. *if it contains the "@" character, the address should be interpreted as an FQDN RFC2822 address*
2. *if it is completely numeric, except possibly including "+", "*", or "#", it should be interpreted as "/TYPE= PLMN", e.g. an E.164 address, a local telephone number, or a numeric short code,*
3. *otherwise, it should be interpreted as an unqualified RFC2822 address (alphanumeric short code)*

180. Thus, a POSITA would have found obvious to include such addressing in the upload message to facilitate transmission of data between devices of a network. Thus, given TS-23.140's disclosures and well-known device addressing aspects for

network communications, a POSITA would have found obvious to include such addressing in the upload message for transmitting data between network devices.

(k) *[5c]: the network server system using the interface to a network to deliver content from the upload messages to the respective identified network elements.*

181. As described in [1a] and [1c] above, the communications between the MMS Relay/Server and the recipient MMS User Agents would happen using the MM1 and MM4 interfaces, which facilitate the network communications between network elements via one or more MMS Relay/Servers, where the network elements would be the MMS User Agents residing on MMS client devices/UEs.⁴

182. For the above reasons, TS-23.140 renders obvious [5a]-[5c].

3. Claim 6

(l) *[6]: The message link server of claim 1, wherein at least one of the one or more messages for delivery by the*

⁴ Note that claim 1 does not recite a “network server system” and thus there is no antecedent basis for “the network server system” in claim 5. For purposes of claim interpretation, I have interpreted “network server system” as referring to the message link server of claim 1 and its interface to the network, as recited in claim 1.

transport services stack comprises multiple identifier/data pairs.

183. As described below, TS-23.140 renders obvious that at least one of the one or more messages for delivery by the transport services stack comprises multiple identifier/data pairs.

184. As described with reference to [1c1]-[1c2] and [1d3]-[1d4] (which I have incorporated by reference here), an originating MMS User Agent (or an originating MMS VAS Application) sends, via the MMS Relay/Server, application data as well as the identifier of the destination application to the destination application corresponding to a destination MMS User Agent. Thus, such a message includes an identifier/data pair in the form of the application data and the destination application identifier. SAMSUNG-1004, 54-56.

185. Moreover, a POSITA would have recognized that consolidating application data and corresponding application identifiers in a message would have been straight-forward and readily implementable, and doing so would achieve network efficiencies from using a single communication/message to transmit data/identifiers for multiple applications (instead of providing separate transmissions with separate messages for each recipient application (including application data and identification for the same)).

186. In other words, a POSITA would have found obvious (from a network efficiency standpoint) for messages directed to multiple such applications to be condensed into one communication/message.

187. Indeed, TS-23.140 discloses message delivery being triggered when the MMS User Agent becomes available/reachable (i.e., moves to coverage or switches on device) (SAMSUNG-1004, §7.1.2.1, §7.1.3) and a POSITA would have found obvious for all queued/buffered message content intended for multiple applications associated with the particular MMS User Agent to be transmitted to the device and the particular MMS User Agent in a single message/communication that includes application data and application identifier for each such application (i.e., multiple identifier/data pairs).

188. Also, applicable are the concepts of broadcast (<https://csrc.nist.gov/glossary/term/broadcast>) and multicast (Tom Bachert, IPv4 Multicast Security: A Network Perspective, SANS Institute. <https://sansorg.egnyte.com/dl/Y0z5h8Z2F4>) messaging, wherein a single address can be used to send messages to many or all devices. As such, a POSITA would have understood or found obvious that there were multiple known techniques for delivering messages to multiple recipients (thereby providing messages with multiple identifier/data pairs that are intended for applications—where data for

application is routed to that application using its identifier included in the message(s)).

189. For the above reasons, TS-23.140 renders obvious claim 6.

4. Claim 7

(m) [7]: *The message link server of claim 1, the device messaging agent on at least one of the wireless end-user devices further to initiate the respective secure Internet data message link to the transport services stack.*

190. As described in [1a] (which I have incorporated by reference here), in MMS, TLS is used for secure messaging between the MMS User Agent and the MMS Relay/Server, and the MMS User Agent is responsible for sending messages by using such network protocols. Thus, a POSITA would have found it obvious for the TLS-based communication link between the UE/client device and the MMS Relay/Server to be initiated by the client device.

191. Indeed, before the Critical Date, it was well-known for TLS-based communications between a UE/client device and a server to be initiated by the client device. See, e.g., SAMSUNG-1012, 29:31-30:24 (describing that TLS requires a handshake phase and a data transfer phase, and in the handshake phase, the “client side initiates a TLS handshake by sending the server a ClientHello message”).

192. Given that the MMS User Agent sends messages to the MMS Relay/Server (at the user terminal/client device) via a TLS-based connection and that TLS communications are initiated by a client device, a POSITA would have

obvious that the MMS User Agent (i.e., the device messaging agent on at least one of the wireless end-user devices) initiates the respective secure Internet Data message link to the transport services stack of the MMS Proxy/Relay server.

193. Additionally, as described with reference to claim 1 (limitations 1d2-1d4), TS-23.140 discloses that MMS User Agent may be unavailable and subsequently, when it becomes available, a POSITA would have found obvious that the MMS User Agent may initiate and maintain a communication with the MMS Relay/Server to retrieve any messages stored therein and intended for MMS User Agent. Thus, this further confirms the obviousness of the MMS User Agent/device initiating the TLS-based secure Internet data message link to the MMS Relay/Server (and its associated network communication and transport protocol stock).

194. For the above reasons, TS-23.140 renders obvious claim 7.

5. Claim 9

(n) [9]: *The message link server of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.*

195. TS 123.140 renders obvious that one of the message delivery triggers is the expiration of a periodic timer.

196. In more detail, TS-23.140 discloses that the originating MMS User Agent can set the desired time of delivery of the MM, in which case the MM is stored

by the MMS Relay/Server until the ***earliest desired time of delivery***, before the message is delivered. See SAMSUNG-1004, § 7.1.1, p. 27.

197. A POSITA would have understood or found obvious that specifying the desired time of delivery triggers a periodic timer upon the receipt of a message that expires when the delivery time is reached, which then triggers the delivery of the message. Such a timer would be periodic in that each time a message is received, the timer is triggered and continues for a particular period at the expiration of which, the message is delivered to the intended recipient.

198. Additionally, or alternatively, MMS User Agent uses periodic polling to identify new messages on an external server, and subsequently retrieves those messages—which also demonstrates another example a message delivery trigger that is the expiration of a periodic timer (as indicated by periodic polling). See SAMSUNG-1004, §8.3.3, pp. 90-91 (explaining that discovery of new messages on external servers can be done by “***periodic polling*** for messages on External Server, followed by retrieval by the MMS User Agent via the MMS/Relay Server”); §3.1, p. 14 (explaining that an external server is a “network entity/application of an external system such as Internet email, unified messaging system or facsimile to which MMs may be sent to and/or from which MMs may be received by an MMS User Agent via an MMS service provider”).

199. A POSITA would have understood that (1) the above-described periodic polling for messages is akin to the claimed feature of a periodic timer expiring because the periodic polling would be done at periodic intervals (i.e., after a certain time period has elapsed/expired) and (2) this periodic polling is a message delivery trigger because, in response to receipt of such polling, messages from the respective servers would be provided/delivered to the MMS User Agent via the MMS Relay/Server.

200. For the above reasons, TS-23.140 renders obvious claim 7.

6. Claims 11-12

- (o) *[11]: The message link server of claim 1, wherein one of the message delivery triggers is the receipt of a transmission on the respective secure message link from the device link agent of the given one of the wireless end-user devices, or a response generated to a transmission received from that device link agent. / [12]: The message link server of claim 11, wherein the transmission is a heartbeat message generated by the given device link agent, or a request received from the given device link agent.*

201. As described in [1d2] (which I have incorporated by reference here), TS-23.140 discloses that one of the message delivery triggers is “the receipt of a transmission on the respective secure message link from the device link agent of the given one of the wireless end-user devices,” where the transmission is “a request received from the given device link agent.”

202. Specifically, TS-23.140 discloses that the MMS User Agent can send a retrieval request to the MMS Relay/Server to retrieve the message from the MMS Relay/Server, which then triggers delivery of the message to the MMS User Agent. See SAMSUNG-1004, §7.1.2.1 (describing MMS Relay/Server sending “a notification” to the recipient MMS User Agent and in response, the MMS User Agent “retriev[ing]” the message at the manual request of the user for such retrieval).

203. For the above reasons, TS-23.140 renders obvious claims 11-12.

7. Claim 13

(p) *[13]: The message link server of claim 1, wherein one of the message delivery triggers is the receipt of a particular network element message from one of the network elements.*

204. TS-23.140 discloses that one of the message delivery triggers is the receipt of a particular network element message from one of the network elements (e.g., a message without any indication of earliest desired time of delivery, a message with application-specific data where the recipient MMS User Agent’s capability indication indicates supports for transmission of application data).

205. As one example, TS-23.140 explains that if a multimedia message is received without an “earliest desired time of delivery” or if the originator MMS Relay/Server does not support this feature, the multimedia message is “immediately routed forward” to the intended MMS User Agent. See SAMSUNG-1004, § 7.1.1, pp. 27, 32.

206. As another example, TS-23.140 explains that “[i]f the MMS Relay/Server finds from the recipient MMS User Agent’s capability indication (see clause 7.1.3.1) that the recipient MMS User Agent supports transport of application data, the MMS Relay/Server shall not perform any type of content adaptation to a multimedia message (MM) that may be contained in the payload of an abstract message that contains a destination application identifier” and “shall pass on the destination application identifier, the “reply-path” identifier (if present) and the additional application/implementation specific control information (if present) unaltered.” *See id.*, § 7.1.18.2.2, pp. 55-56. Thus, the receipt of this particular type of message with application data by the MMS Relay/Server constitutes a message delivery trigger that causes deliver of the message by the MMS Relay/Server to the MMS User Agent (for delivery to one or more device applications). *Id.*

207. In this manner, a POSITA would have understood that TS-23.140 contemplates message delivery triggers that include receipt of certain types of messages from one or more network elements (originating MMS User Agents operating of UEs/client devices).

208. For the above reasons, TS-23.140 renders obvious claim 13.

IX. GROUND 2: CLAIMS 2 AND 3 ARE RENDERED OBVIOUS BY TS-23.140 AND SHEN

A. Combination of TS-23.140 and Shen

209. Both TS-23.140 and Shen disclose a system in which an MMS Relay/Server receives and stores messages before transmitting those messages to MMS User Agents. SAMSUNG-1004, pp. 16-18, 54-56; SAMSUNG-1014, [29]-[34], FIGS. 1-2, 5. Shen, however, explains that the store and forward functionality of the MMS Relay/Server exposes a “potential security problem” given that such systems are “not an end-to-end solution” SAMSUNG-1014, [0004].

210. To overcome this “security problem,” Shen’s invention in one embodiment discloses that “an authentication and key management module 502 is added in the MMS gateway to distribute symmetric keys to users and a certificate management module 504 is added in the user device,” as shown below:

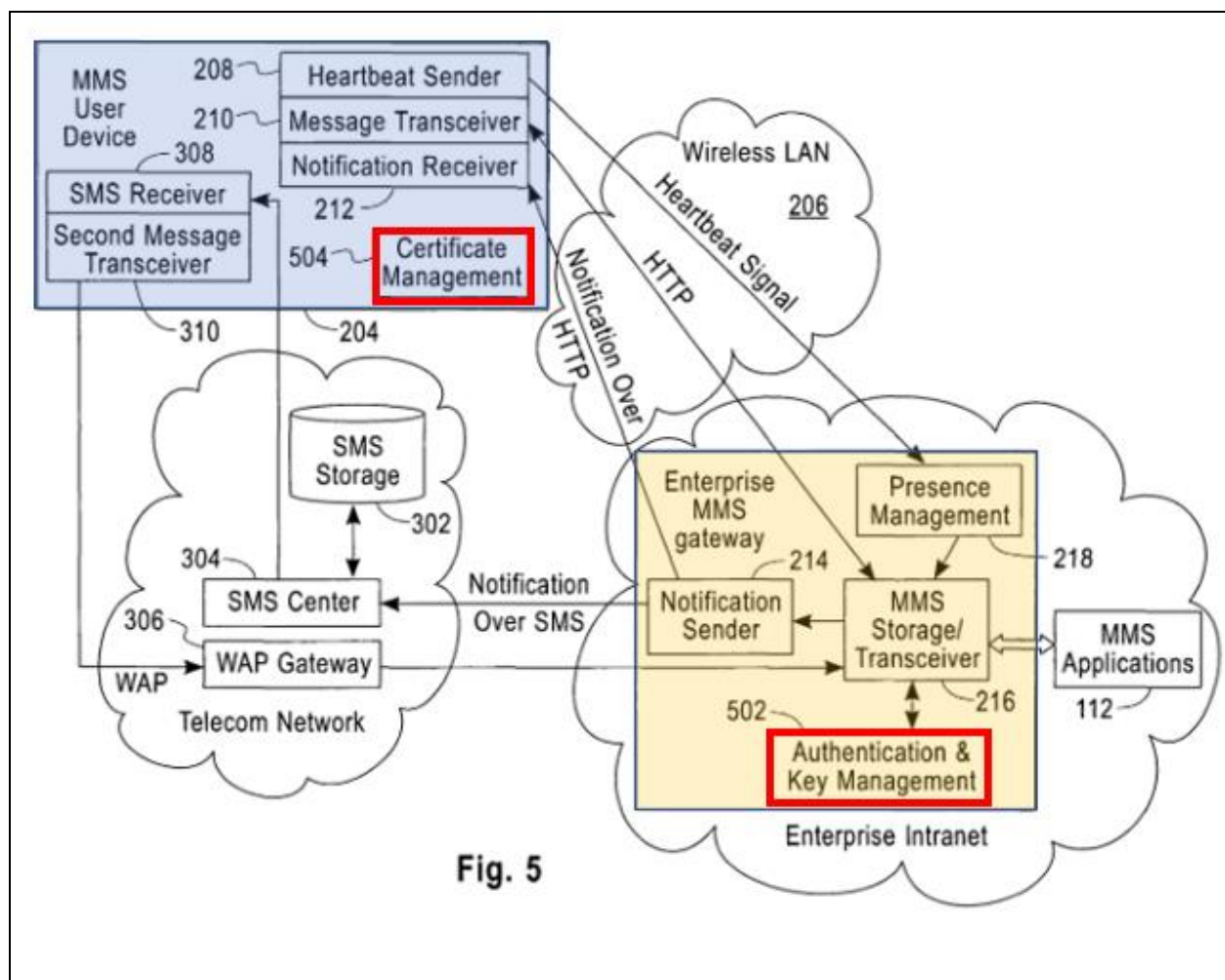


Fig. 5

211. Using these modules, Shen discloses generation of a symmetric key that is then used by the MMS Relay/Server to encrypt data communications transmitted to the MMS User Device/Agent of user devices. SAMSUNG-1014, [55]-[60].

212. Combining the above-described teaching of Shen with those of TS-23.140 would have been obvious to a POSITA at least because such a combination would have merely involved (1) combining prior art elements according to known methods to yield predictable results and (2) the use of known technique to improve similar devices (methods, or products) in the same way.

213. Indeed, TS-23.140 and Shen provide similar MMS architectures whereby MMS User Devices/Agents communicate with an MMS Relay/Server. And given the “security problems” in data communications occurring in such environments, a POSITA would have been motivated to implement known techniques that are disclosed in Shen to solve these problems—namely implementing an “authentication and key management module 502” at the MMS Relay/Server that generates and “distribute[s] symmetric keys to users” and using this symmetric key to encrypt data transmitted to “the user device,” as contemplated by Shen. SAMSUNG-1014, [0054]-[0060].

214. A POSITA would have been motivated to so combine the teachings of these references to achieve an MMS system “having improved security,” resulting from “provi[sion] [of] an end-to-end security solution for MMS applications,” which is particularly beneficial in the context of “enterprise applications.” SAMSUNG-1014, [0017], [0021] (describing additional advantages stemming from Shen’s invention). Moreover, it was well known before the Critical Date for push serves to encrypt content of messages sent to push clients/client devices. Chou is one example of a push message system that uses a “WAP gateway” (e.g., a network message server) that encrypts content received from a “content server” for transmission to a “mobile client” that then “decodes the response.” SAMSUNG-1009, ¶¶[0054]-[0060]. Rakic and Shen provide additional examples of a push server encrypting

message traffic. SAMSUNG-1015, ¶¶[0065]-[0068]; SAMSUNG-1014, ¶¶[0054]-[0060].

215. So configuring the TS-23.140 system to implement Shen's above-described modules and associated encryption key generation functionality to facilitate encryption of messages (as described above) would have amounted to the use of a known technique to improve similar devices in the same way, and combining prior art elements according to known methods to yield predictable results.

216. And, given that TS-23.140's system described message delivery in a similar MMS environment (as in Shen) involving a similar MMS communication between MMS User Agents/Device and the MMS Relay/Server, a POSITA would have found it straightforward to modify and/or program the MMS Relay/Server (and the MMS User/Agent)—with a reasonable expectation of success—to implement Shen's above-described modules and associated functionality for generating an encryption key and encrypting communications between the MMS User Agent and the MMS Relay/Server using that key.

217. Additionally, the resulting system's elements would each perform functions they were known to perform prior to the combination—i.e., TS-23.140's system would continue to facilitate communication of messages between MMS Relay/Server and MMS User Agents, and Shen's teachings (as implemented in

combination) would facilitate encryption of messages by the MMS Relay/Server that are then sent to the MMS User Agent.

218. As described below, the combination of TS-23.140 and Shen (referred to herein as TS-23.140-Shen) renders obvious claims 2 and 3.

B. Claim Analysis

1. Claim 2

- (a) [2]: The message link server of claim 1, further comprising an encrypt function to encrypt the one or more messages supplied to the transport services stack for delivery on the secure message link maintained between the message link server and the device link agent on the given one of the wireless end-user devices*

219. TS-23.140-Shen renders obvious an encrypt function to encrypt the one or more messages supplied to the transport services stack for delivery on the secure message link maintained between the message link server and the device link agent on the given one of the wireless end-user devices.

220. As described with reference to [1a] and [1d4] (which I have incorporated by reference here), TS-23.140 renders obvious that messages are supplied to the MMS Relay/Server's transport services stack for delivery on the secure message link between the MMS Relay/Server and an MMS User Agent (resident on a particular user device/terminal).

221. Additionally, as described in §IX.A and VII.B (which I have incorporated by reference here), the combination would have implemented Shin's

teachings of authentication and key management module at the MMS Relay/Server, which would have facilitated generation of a shared key between the user device and the MMS Relay/Server, where such shared key can be used to “encrypt[] MMS messages with the shared secret key.” SAMSUNG-1014, [0059].

222. In this manner, TS-23.140-Shen combination would have implemented an encryption function that encrypts the messages sent by the MMS Relay/Server to the MMS User/Agent (where such message would be delivered by the transport services stack on the secure link to the MMS User/Agent, as described above). *Id.*

223. For the above reasons, TS-23.140-Shen renders obvious claim 2.

2. Claim 3

(a) [3]: The message link server of claim 2, wherein the encrypted one or more messages are transported to the device link agent on the given one of the wireless end-user devices using one or more of encryption on the transport services stack, IP (Internet Protocol) layer encryption, and tunneling.

224. TS-23.140-Shen renders obvious the encrypted one or more messages are transported to the device link agent on the given one of the wireless end-user devices using one or more of encryption on the transport services stack, IP (Internet Protocol) layer encryption, and tunneling.

225. As described with respect to claim 2 (which I have incorporated by reference here), TS-23.140-Shen renders obvious encrypting messages and transporting the encrypted messages to a particular MMS User Agent on a UE/terminal.

226. Additionally, as explained with reference to [1a] (which I have incorporated by reference here), the combination renders obvious encryption on the transport services stack in view of the above-described teachings of transport layer security (TLS) on the transport/communication link between the MMS Relay/Server and MMS User Agent (i.e., encryption on the transport services stack).

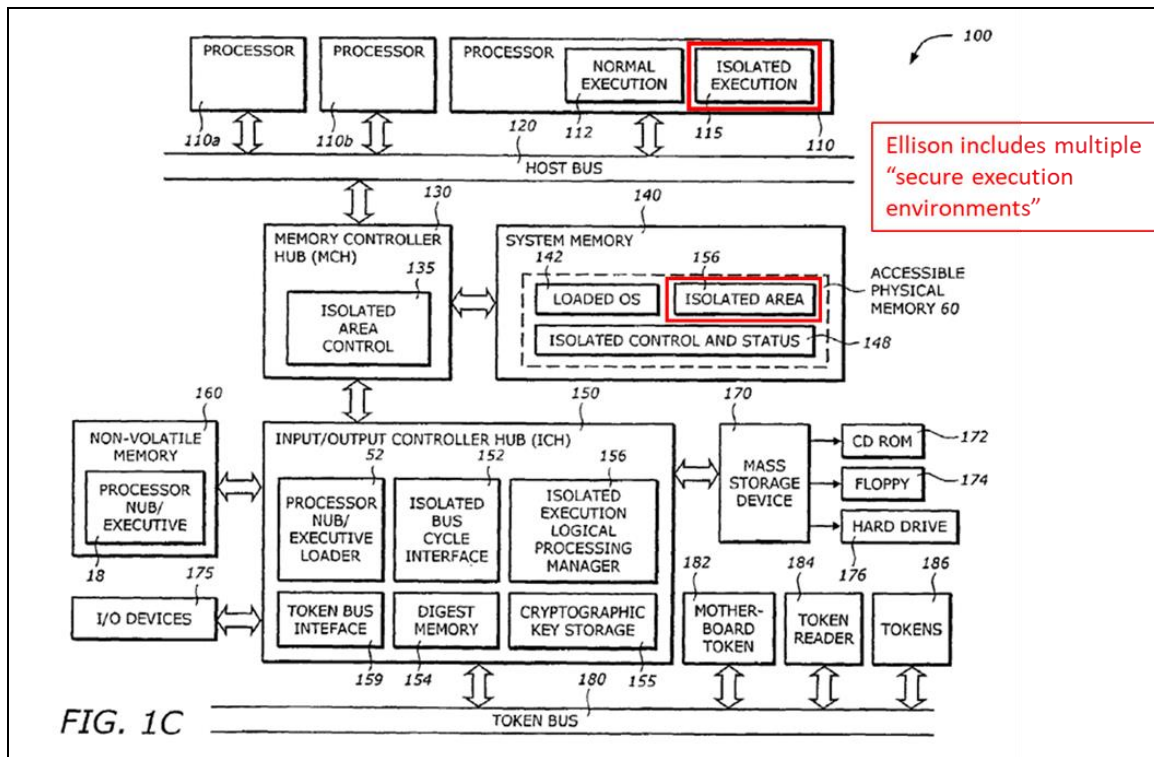
227. For the above reasons, TS-23.140-Shen renders obvious claim 3.

x. GROUND 3: CLAIMS 4 IS RENDERED OBVIOUS BY TS-23.140 AND ELLISON

A. Ellison

228. Ellison discloses a techniques for “protect[ing] a subset of a software environment.” SAMSUNG-1019, Abstract. This includes multiple “operating system nub key[s] (OSNK)” which are “unique to an operating system (OS) nub.” *Id.*

229. A “usage protector” uses the OSNK to “protect usage of [a] subset of the software environment.” *Id.*



SAMSUNG-1019, FIG. 1C.

B. Combination of TS-23.140 and Ellison

230. A POSITA would have found obvious to combine these references such that Ellison's secure platform would have been incorporated into TS-23.140's UE/terminals. SAMSUNG-1019, 8:25-32, 8:66-9:62, FIG. 2. A POSITA would have combined the teachings of these references for the following reasons.

231. First, a POSITA would have recognized that Ellison's secure platform would have improved the security of TS-23.140's MMS system by improving device security. SAMSUNG-1019, 8:25-32, 8:66-9:62, FIG. 2.

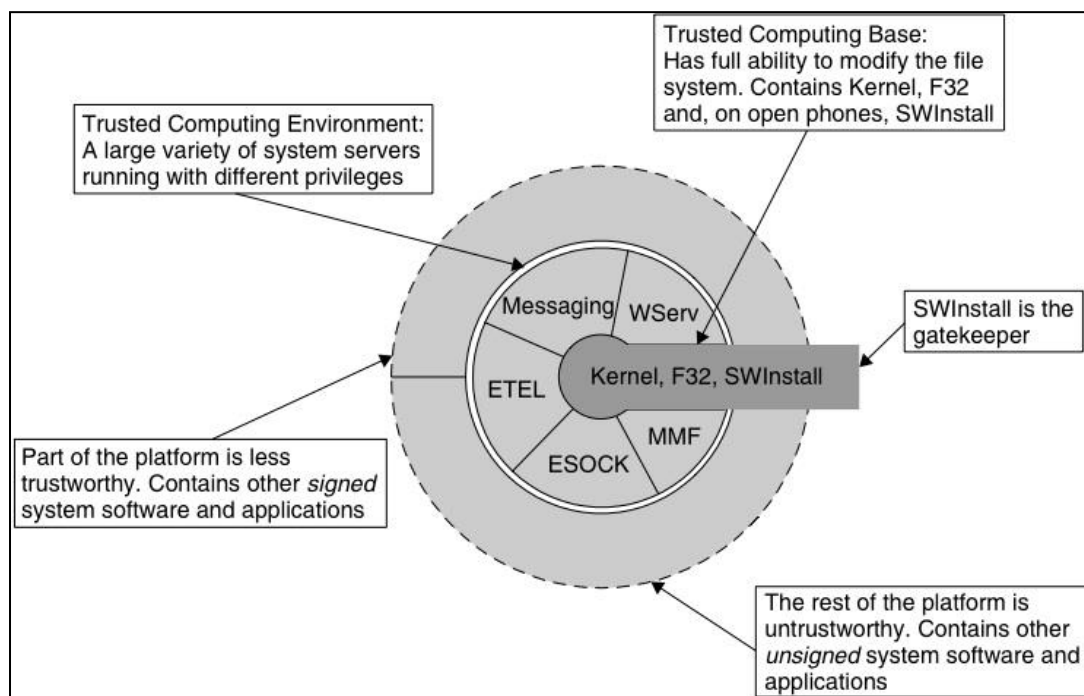
232. Second and relatedly, such increased confidence in device and/or network security provided by Ellison would have enabled a POSITA to TS-23.140's MMS services to additional UEs/servers (VAS Applications) *Id.*

233. Indeed, a POSITA would have understood that device level security improvements are an efficient, beneficial option for improving network security, which would have been a natural solution for TS-23.140's system, which contemplates secure/encrypted messaging in the MMS environments.

234. As explained below, combining the teachings of TS-23.140 and Ellison would have been obvious because such a combination would have merely involved combining prior art elements according to known methods to yield predictable results. Incorporating Ellison's security techniques into TS-23.140's mobile terminals would have been predictable and foreseeable with a reasonable expectation of success because Ellison describes that its techniques can be implemented in "computer system[s]" (such as TS-23.140's UEs) that would include a "processor." SAMSUNG-1019, 5:11-16. Indeed, a POSITA would have recognized or found obvious that UEs/terminals in TS-23.140 would include a "processor" capable of implementing Ellison's techniques.

235. Third, in the context of mobile devices and their associated operating systems, it was well-known before the Critical Date to implement more secure systems, with rings or tiers of protection around different device components. For

example, in the early 2000s, the Symbian mobile operating system (B. Li, E. Reshetova, T. Aura, “Symbian OS platform security model”, LOGIN: Magazine, August 2010; Craig Heath, Symbian OS Platform Security: Software Development Using the Symbian OS Security Architecture (Wiley, 2006)) separated processes into three “Trust Tiers”: The Trusted Computing Base (TCB), The Trusted Computing Environment (TCE), and Applications, as illustrated below:



236. The first two layers/tiers shown were primarily dedicated to components of the Operating System with relatively higher operating privileged, followed by the Applications tier with applications having relatively lower privileges. The first two tiers included the Trusted Computing Base and the Trusted Computing Environment. As shown, the Trusted Computing Environment includes, among other services, a Messaging service. The Application Tier could be further

subdivided into Signed and Unsigned applications. Signed applications could interact with the TCE and have this layer execute higher privileged requests, whereas Unsigned applications could not access that same functionality.

237. Thus, it was well-known for operating systems to include the messaging services and clients/agents responsible for provision of those services, within the trusted computing environment, and for applications (including those that would be intended recipients of those messages) to be tiered/organized outside this environment. Given that TS-23.140 contemplates mobile terminals with an MMS User Agent that communicates with applications on the device, the well-known need to protect computing environments (including components providing essential services such as messaging) from other applications/components of the environment, and Ellison's own disclosures confirming the same, a POSITA would have been motivated to implement Ellison's teaching of secure execution environment that enable such improved security within the MMS environment, such that the MMS User Agent is organized as part of the trusted computing environment (i.e., the secure execution environment) and enables communications with applications residing outside this trusted/secure environment.

238. In an example of the combined system, Ellison's secure platform would have been incorporated into TS-23.140's mobile terminals, to secure the MMS User

Agent, and further have a usage protector protecting messaging operations of the MMS User Agent. SAMSUNG-1019, 8:25-32, 8:66-9:62, FIG. 2.

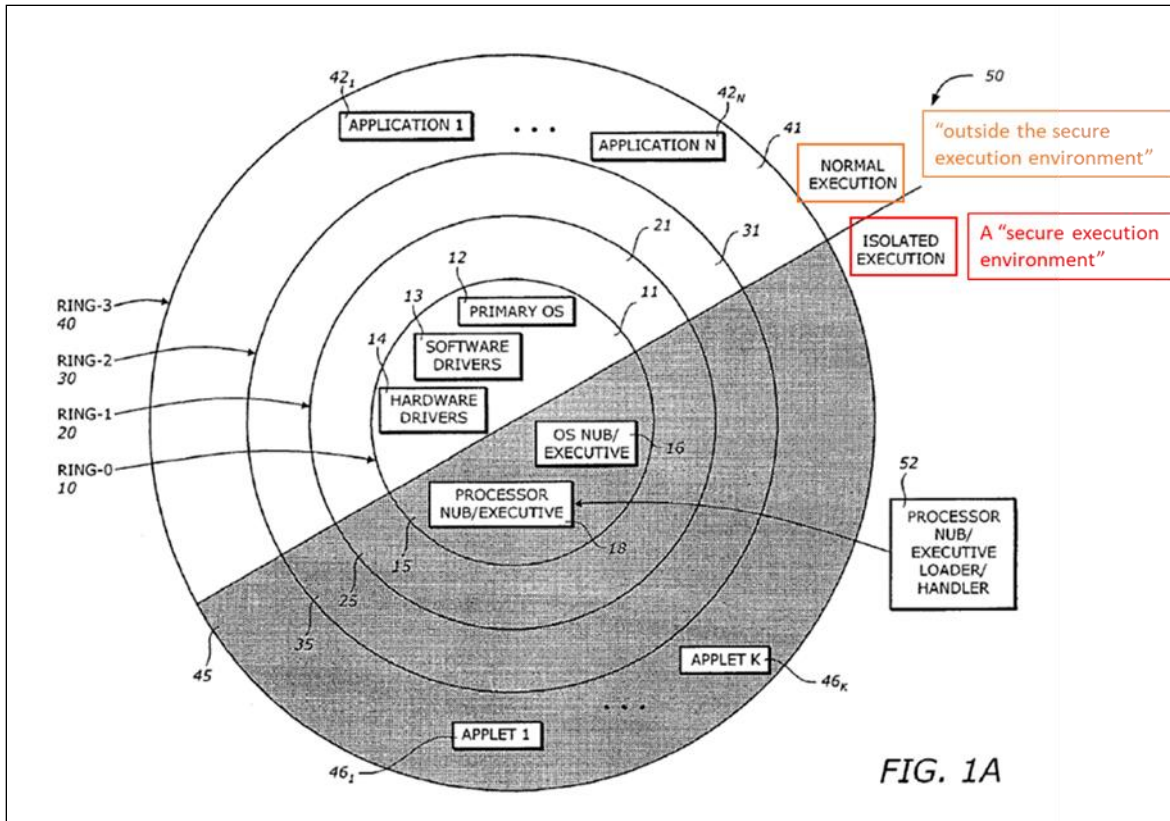
C. Claim Analysis

1. Claim 4

- (a) *[4]: The message link server of claim 1, wherein the device link agent executes in a secure execution environment on at least one of the devices, and at least one of the software components executes outside of the secure execution environment on that device.*

239. Ellison discloses both an “isolated execution mode” where “access ... is restricted” and a “normal execution mode” that “operates in a non-secure environment, or a normal environment without the security features provided by the isolated execution mode.” SAMSUNG-1019, 4-65-67, 5:1, 6:1-26, 8:25-32, FIGS. 1A, 1C.

240. Additionally, Ellison’s teachings enable “normal” execution, for instance, for applications “***outside of the secure execution environment on that device.***” SAMSUNG-1019, 4-65-67, 5:1, 6:1-26, FIGS. 1A, 1C.

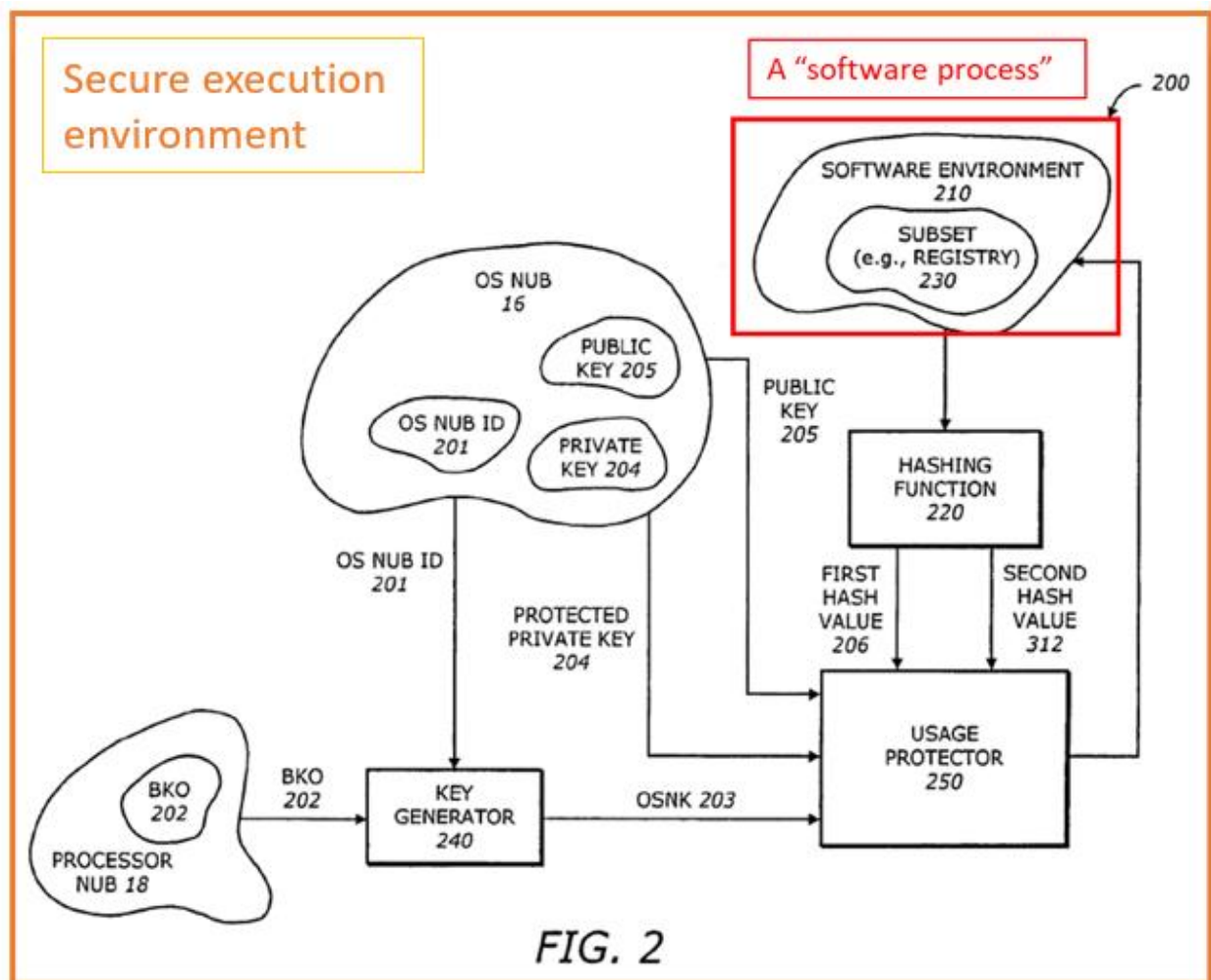


SAMSUNG-1019, FIG. 1A.

241. Ellison further discloses an “isolated area 70” which is “[a] memory area that is defined by the processor 110 when operating in [an] isolated execution mode.” SAMSUNG-1019, 6:1-26, 8:25-32, FIGS. 1A-1C, 2. Ellison refers to this technique, illustrated below in Figure 2, as a “secure platform.” SAMSUNG-1019, 6:1-26, 8:25-32, 8:66-9:62, FIGS. 1A-1C, 2.

242. Ellison further teaches that the secure platform includes a “key generator 240” that “generates a key operating system nub key (OSNK) 203” which is then supplied to “trusted agents.” *Id.*, 8:66-9:6, FIG. 2. One example of a trusted agent is a “usage protector 250” that “uses the OSNK 203 to protect the usage of [a]

subset 230.” *Id.*, 9:28-40, FIG. 2. Ellison’s usage protector 250 uses a hashing function with the subset 203’s ONSK 203 to determine if the subset 203 has been altered following changes (e.g., reads or writes to the subset 203), thereby providing “protection against unauthorized reads, and detection of intrusion, tampering or unauthorized modification” *Id.*, 9:47-62.



SAMSUNG-1019, FIG. 2.

243. In implementing the TS-23.140-Ellison combination, Ellison’s above-described teachings of a usage protector and secure platform, which operate within

the secure, isolated area (secure execution environment) would be incorporated into TS-23.140's MMS User Agent executing on UEs/terminals, which in turn would provide secure access to applications residing outside the secure execution environment.

244. A POSITA would have been motivated to so combine these teachings because (1) a POSITA would have recognized that Ellison's secure platform would have improved the security of TS-23.140's MMS system by improving device security (SAMSUNG-1019, 8:25-32, 8:66-9:62, FIG. 2); and (2) such device level security improvements are an efficient, beneficial option for improving network security, which would have been a natural solution for TS-23.140's system, which contemplates secure/encrypted messaging in the MMS environments. *Id.*

245. In other words, the combined system would enable more secure transmission of MMS messages to applications on the mobile terminal, and prevent potential bad actors from sending malware to the mobile terminal in the form of such messages or from modifying or otherwise gaining unauthorized access to the messages. *Id.*; SAMSUNG-1019, 8:25-32, 8:66-9:62, FIG. 2.

246. Moreover, combining TS-23.140 and Ellison's teachings would have been obvious because such a combination would have merely involved combining prior art elements according to known methods to yield predictable results. Incorporating Ellison's security techniques into TS-23.140's mobile terminals

would also have been predictable and foreseeable with a reasonable expectation of success because Ellison describes that its techniques can be implemented in “computer system[s]” (such as TS-23.140’s UEs) that would include a “processor.” SAMSUNG-1019, 5:11-16. Indeed, a POSITA would have recognized or found obvious that UEs/terminals in TS-23.140 would include a “processor” capable of implementing Ellison’s techniques.

247. Moreover, as noted above, in §§X.B and V, it was well-known before the Critical Date for operating systems to include the messaging services and clients/agents responsible for provisions of those services, within the trusted computing environment, and for applications (including those that would be intended recipients of those messages) to be tiered/organized outside this environment.

248. Given that TS-23.140 contemplates mobile terminals with an MMS User Agent that communicates with applications on the device, the well-known need to protect computing environments (including components providing essential services such as messaging) from other applications/components of the environment, and Ellison’s own disclosures confirming the same and providing such tiered, security architecture within the device, a POSITA would have been motivated to similar enable such improved security within the MMS environment by implementing the MMS User Agent as part of the trusted computing environment

(i.e., the secure execution environment) and providing the applications that communicate with this agent outside this trusted/secure environment.

249. For the above reasons, TS-23.140 renders obvious that the MMS User Agent (device link agent) executes in a secure execution environment and one or more of the applications to which push messages are routed (at least one of the software components) executes outside of the secure execution environment.

XI. GROUND 4: CLAIM 8 IS RENDERED OBVIOUS BY TS-23.140 AND RAKIC

A. Claim Analysis

1. Claim 8

- (a) [8]: The message link server of claim 1, further comprising a secure server to provide secure authorization signatures to the given one of the wireless end-user devices, the secure authorization signatures indicating the authorized software components that are allowed to receive data from se-cure message link messages via the message link server.*

250. For the reasons described below, TS-23.140 in view of Rakic renders claim 8 obvious that message link server (e.g., the MMS Relay/Server) comprises a secure server (e.g., a secure push message server) to provide secure authorization signatures to the given one of the wireless end-user devices (e.g., UE/terminal), the secure authorization signatures indicating the authorized software components (e.g., applications receiving application data that have previously undergone an

application registration process) that are to receive data from secure message link messages via the message link server.

251. In more detail, TS-23.140 discloses an “application registration process” that registers an application with the MMS User Agent and allows it to use MMS to transport application-specific data. SAMSUNG-1004, §7.1.18, p. 54.

252. As a result, when an application is registered and subsequently receives an application-specific message, that message is routed to the application based on the application identifier of the application as included in the message. *Id.*, pp. 54-56 (“If the destination application resides on a receiving MMS VAS Application, the MMS VAS Application shall immediately route the received MMS information on to the destination application that is referred to by the destination application identifier **(based on the negotiated details upon application registration process)**; *id.* (“**If the MMS Relay/Server finds from the recipient MMS User Agent’s capability indication (see clause 7.1.3.1) that the recipient MMS User Agent supports transport of application data**, the MMS Relay/Server shall not perform any type of content adaptation to a multimedia message (MM) that may be contained in the payload of an abstract message that contains a destination application identifier; ... shall pass on the destination application identifier, the 'reply-path' identifier (if present) and the additional application/implementation specific control information (if present) unaltered”).

253. Specifically, if the destination application resides on a receiving MMS User Agent and is registered, “the MMS User Agent shall immediately route the received MMS information on to the destination application that is referred to from the destination application identifier ***(based on the negotiated details upon application registration process)*** without presentation to the user.” *Id.*

254. In this manner, the MMS User Agent includes data indicating the authorized/registered applications (i.e., software components) that are allowed to receive data from secure message link message via the MMS Server (i.e., message link server).

255. TS-23.140 additionally discloses that, if the destination application identifier is present in the message, the message is directly routed to the destination application. *Id.* However, TS-23.140 does not expressly describe validation or authentication of the message that verifies the identity of the sender of the message.

256. Without such verification, a POSITA would have understood that the device and/or the destination application can be vulnerable to attacks by unknown third party senders. Nor does TS-23.140 verify that the message is indeed intended for the correct application, which too can inadvertently result in confidential or sensitive data or information leaking to an incorrect device or an application.

257. MMS communications are a type of push communications. SAMSUNG-1004, p. 14 (“MMS uses a number of technologies to realise [sic] the

requirements of the stage 1 description (3G TS 22.140) [1]. The present document describes how the service requirements are realised with the selected technologies. As far as possible existing protocols (e.g. WAP, SMTP, ESMTP as transfer protocols; lower layers to provide push, pull, notification) and existing message formats (e.g. SMIL, MIME) shall be used for the realisation [sic] of the Multimedia Messaging Service.”).

258. And, in the context of push communications, Rakic provides one well-known solution to address the above-described computer security concerns. As described in §VII.C, Rakic discloses a database device 108 that stores information related to client devices (e.g., codes obtained from a client device such as IMEI or a component in the client device (e.g., removable memory), and a secure push message server 504 uses stored information in the database device 108 to generate a key, an electronic signature, and a secure push message for a push message intended for a particular client device. SAMSUNG-1015, [0047]-[0049], [65].

259. The secure push message server generates the key based on the above-described code(s) (e.g., IMEI) that are obtained from client device 102 or a component thereof, and subsequently, generate an electronic signature based on the key and the message. *Id.*, [0066].

260. The secure push message server 504 generates and sends the secure push message to client device 102, e.g., by appending the electronic signature to the

message and/or other information (e.g., a timestamp). *Id.*, [67]. At the client device, a secure push message client can validate the message and the sender by generating a key and then generating a signature using this generated key and the data block of the received message. *Id.*, [0109]-[0112]. If the signature portion of the received message matches the signature portion of the received message, the secure push message is determined to be valid. *Id.*

261. A POSITA would have been motivated to implement the above-described techniques in Rakic within the MMS Relay/Server to address the above-described security problems. Indeed, a POSITA would have been motivated to do so because, in view of Rakic, the above-described techniques enable (1) the client device to “authenticate the sender” and (2) “verify that an intended recipient of the secure push message is the client device, and that the client device includes a correct component.” SAMSUNG-1015, [0041].

262. In doing so, a POSITA would have recognized the improvements to data and computer security for both the client device and the underlying applications to which the data is being transmitted. *Id.* Similar modifications to email, in the form of the Secure/Multipurposed Internet Mail Extensions (S/MIME) (B. Ramsdell, S/MIME Version 3 Message Specification, IETF RFC 2633, June 1999—SAMSUNG-1035) and Pretty Good Privacy (PGP) (Philip Zimmermann, Pretty Good Privacy: RSA Public Key Cryptography for the Masses. PGP User’s Guide.

Version 1.0, June 1991—SAMSUNG-1034), were long known in the art as a means of providing cryptographic guarantees of confidentiality and authenticity in messaging.

263. In view of the above motivations, a POSITA would have implemented Rakic's secure push message server 504 and database device 108, as well as their above-described functionality, as part of the MMS Relay/Server in TS-23.140.

264. So, configuring the TS-23.140 system to implement Rakic's teachings would have amounted to the use of a known technique to improve similar devices in the same way, and combining prior art elements according to known methods to yield predictable results. Indeed, given that TS-23.140 and Rakic both contemplate push messaging with a similar push architectures/systems involving a push server (e.g., MMS Relay/Server and Secure Push Message Server) that communicates with a client/user device, a POSITA would have found it straightforward to implement—with a reasonable expectation of success—the signature generation functionality of Rakic's secure push messaging server as well as its database storing the client device information, as part of the MMS Relay/Server in TS-23.140.

265. Additionally, the resulting system's elements would each perform functions they were known to perform prior to the combination—i.e., TS-23.140's system would continue to facilitate communication of messages between MMS Relay/Server and MMS User Agents, and Rakic's teachings (as implemented in

combination) would facilitate provision of secure signatures (along with the messages) to the message receiving device(s).

266. In the resulting TS-23.140-Rakic combination, Rakic's secure push message server and its associated signature generation functionality would be implemented in the MMS Relay/Server and would be used to generate signatures for messages intended for different UE/terminals and their respective MMS User Agents.

267. A POSITA would have understood that the signatures are authorization signatures because they are used for authentication of the sender and/or verify that the intended recipient of the message is the client device and the client device includes a correct component. SAMSUNG-1015, [0041].

268. Additionally, as Rakic explains, the signature generating functionality "verif[ies] that an intended recipient of the secure push message is the client device, and that the client device includes a correct component." *See id.* A POSITA would have found obvious that one such component could be an application of the device. In other words, a POSITA would have understood that Rakic does not limit its teachings of verification of component to hardware components, and in the context of push communications that are intended to be delivered to specific software components/applications, a POSITA would have found obvious that such components could include software components.

269. As such, in view of Rakic as applied in the context of TS-23.140, a POSITA would have found obvious that the above-described secure authorization signatures indicate the authorized application (i.e., software components) that are allowed/authorized to receive messages including application data via the MMS Relay/Server.

XII. GROUND 5: CLAIMS 1, 5, 6, 7, 9, 11-13, AND 15 ARE RENDERED OBVIOUS BY HOUGHTON AND MUNSON

A. Combination of Houghton and Munson

270. A POSITA would have found obvious to combine the teachings of Houghton and Munson such that Munson's method of buffering and pushing messages would have been incorporated into Houghton's push messaging system of Houghton. *See* §§VII.D and VII.E.

271. A POSITA would have combined Houghton and Munson's teachings for multiple reasons, which are described below.

272. A POSITA would have recognized or found obvious that a mobile terminal that is an intended recipient of a message may not have network connectivity, such that a push server may maintain or store/buffer the message until the mobile terminal has network connectivity and the message can be delivered.

Indeed, Houghton contemplates such "store and forward messaging systems" and further discusses that messages may not deliverable while the user is

out of network coverage. SAMUNG-1007, 3, 7 (“These server computers such as a telecommunications operator SIP Server or WAP Gateway is used to control and deliver messages and ***store messages when the mobile terminal is unavailable and deliver them on the sender[']s behalf, possibly later then the terminal becomes available***”). Munson provides a conventional solution involving storage of messages for later delivery to mobile terminals that would have been efficient and straightforward to implement within the existing setup of Houghton’s push messaging system. SAMUNG-1007, 3, 7, FIGS. 3-4; SAMSUNG-1017, 3:66-67, 4:1-11, 5:22-33, FIG. 3. Moreover, it was well-known for store-and-forward push systems to include buffers to store messages “when the mobile terminal is not connected to the mobile internet service network” and “re-transmitting the stored push information when the mobile terminal is connected to the mobile internet service network.” SAMSUNG-1006, claim 14, FIG. 1, ¶34.

273. A POSITA would have been motivated to implement Munson’s message buffering (and association functionality and component) within Houghton’s system to avoid losing or discarding of messages (e.g., when the mobile terminal is outside network range or connectivity and the message is therefore undeliverable).

274. Additionally, like Munson, Houghton already contemplates “store and forward messaging systems” as well as the above-described scenario where a mobile terminal may be outside network range/coverage. SAMAUNG-1007, 3.

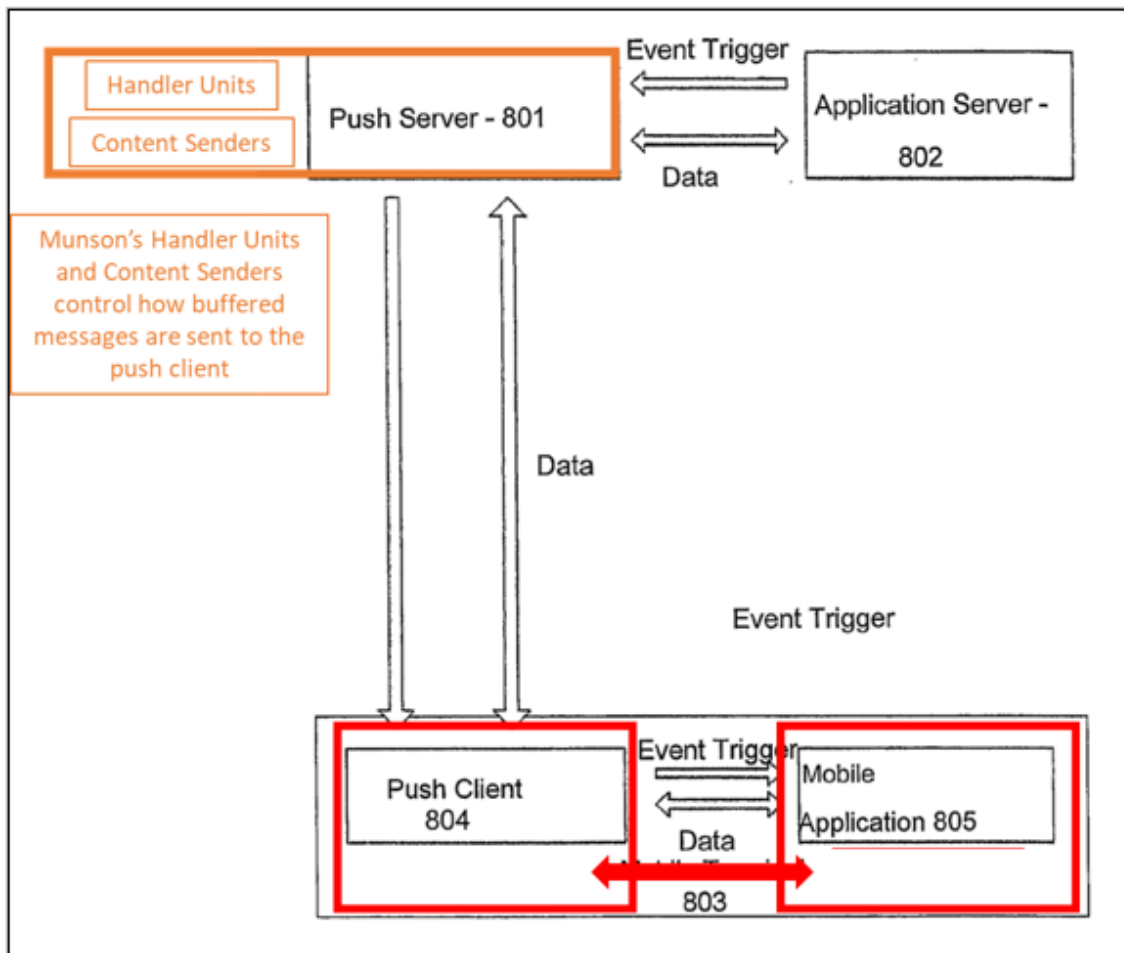
275. Thus, combining Houghton and Munson’s teachings would have been obvious and straightforward because such a combination would have amounted to using known prior art techniques (message buffering) to improve similar devices/systems (push messaging systems, as contemplated by Houghton and Munson) in the same way that yields predictable results (push messaging systems that can store and forward messages and subsequently deliver them, as contemplated by Houghton and as expressly described by Munson).

276. Incorporating Munson’s method of buffering and later transmission of messages (e.g., upon occurrence of a trigger) into Houghton’s push messaging system would also have been predictable and foreseeable with a reasonable expectation of success because (1) Houghton and Munson disclose push messaging systems whereby push messages are transmitted from a push server to a mobile terminal upon occurrence of a trigger event, (2) Houghton contemplates storage and forward systems, which are described in additional detail in Munson, and (3) Munson adds implementation details regarding such message storage and forwarding that enable storage of received messages and subsequent delivery upon

occurrence of a delivery trigger. SAMUNG-1007, 3, 7, FIGS. 3-4; SAMSUNG-1017, 1:11-26.

277. Thus, in the combination, each of Houghton and Munson's systems would continue to operate the way they did prior to combination, e.g., Houghton's system would continue to push messages from a push server to a push client, and Munson's described techniques (as implemented in combination) would facilitate buffering of push messages in a buffer/memory of Houghton's push server and delivering the stored messages when certain conditions arise (e.g., user becomes available in network range).

278. In an example of the resulting combination (referred to as Houghton-Munson) and as illustrated below, Munson's handler units and content senders would have been incorporated into Houghton's push server such that they would buffer push messages based on the policies (as further described by Munson) and subsequently deliver those messages upon occurrence of certain conditions. SAMUNG-1007, 3, 7, FIGS. 3-4; SAMSUNG-1017, 3:66-67, 4:1-11, 5:22-33, FIG. 3.



SAMSUNG-1007, FIG. 8 (modified to incorporate Munson).

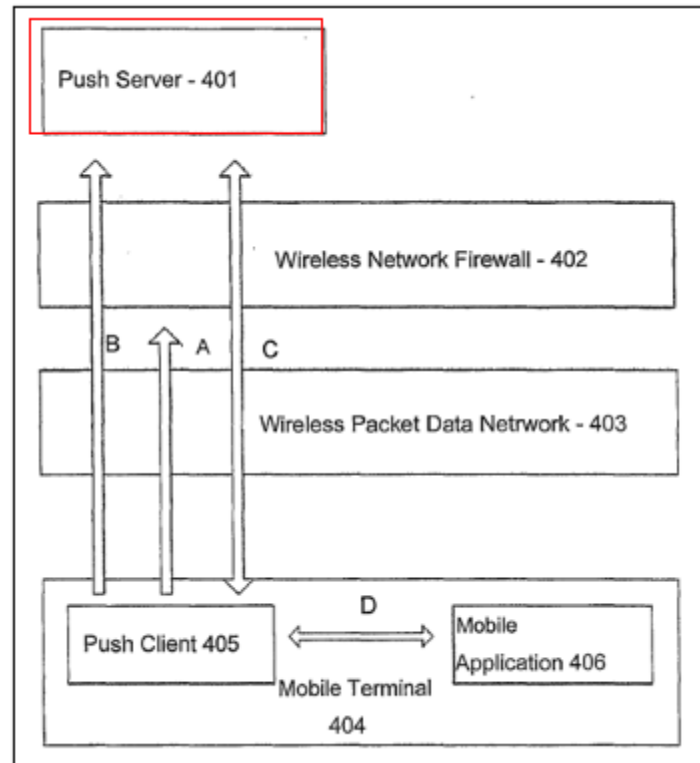
B. Claim Analysis

1. Claims 1 and 15

- (a) [1pre]: A message link server comprising / [15pre]: A method of operating a message link server, comprising:

279. If the preamble is limiting, Houghton-Munson renders obvious a message link server (e.g., push server 401) and a method for operating the same. As described below, and shown with reference to FIG. 4, Houghton's push server 401

sends push messages over communication links formed over a network to mobile terminals.



SAMSUNG-1007, FIG. 4 (“illustrating an exemplary implementation of the system for pushing mobile application launch commands to a mobile terminal based on server side events in accordance with one embodiment of the present invention”).

280. “Figure 4 illustrates a Push Client basic functionality 405 implemented in a mobile terminal 404 **as part of a message passing system in cooperation with a Push Server 401**. The push client 405 is preferably implemented in the form of a software run in a processor of the mobile terminal 404. The push client software 405 runs in the background and attempts to log into the push server 401 over a data

network 403, normally a wireless network capable of transmitting Internet Protocol (IP) packets.” SAMSUNG-1007, 16-17; *see id.*, Abstract (“A server outside a wireless network security entity is adapted to push messages to a mobile terminal located in the wireless network and thereby cause programs to start to a specified operating state or, if already started, change operating state to the state indicated by the message thereby presenting a new application view to the user of the mobile terminal. ***This mobile-initiated permanent connection allows the server to pass or push messages from outside the operator’s firewall to the mobile terminal at any time.***”).

281. Additional details regarding Houghton-Munson’s push server and the method for operating this server are described with reference to [1a]-[1d4]/[15pre]-[15d3] (which I have incorporated by reference here).

282. For the above reasons, Houghton-Munson renders obvious [1pre]/[15pre].

- (b) [1a]: a transport services stack to maintain a respective secure message link through an Internet network between the message link server and a respective device link agent on each of a plurality of wireless end-user devices, / [15a]: maintaining a respective secure message link through an Internet network between the message link server and a respective device link agent on each of a plurality of wireless end-user devices,

283. Houghton-Munson renders this limitation obvious consistent with how this limitation is described in the context of the '192 Patent's specification. Specifically, the '192 Patent explains that the "transport services stack" facilitates communications between the server (service controller 120) and the client device (service processor 115) over a communication link (service control link 1653). SAMSUNG-1001, 90:33-38.

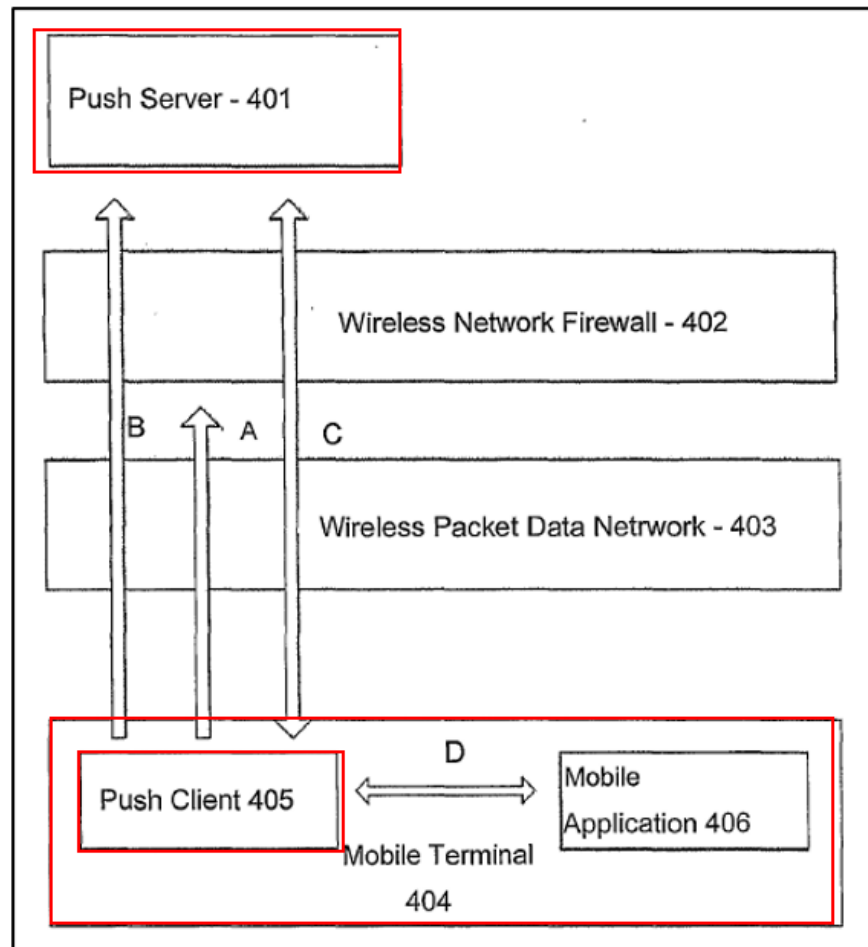
284. The patent further discloses use of transport protocols for such server-client communications, including, e.g., TCP, IP, UDP (*id.*, 90:50-61, 99:8-32) and notes that "there are many other approaches for implementing a secure control channel over the Internet including, for example, one or more of various packet encryption protocols running at or just below the application layer, running TCP Transport Layer Security (TLS), and running IP level security or secure tunnels" (*id.*, 17:13-22).

285. Consistent with these disclosures, Houghton-Munson renders obvious a transport services stack that maintains a secure message link through an Internet network between the message link server (e.g., push server 401) and a device link agent (e.g., push client 405) on respective client devices (e.g., mobile terminals 404).

286. For example, and as explained below, Houghton discloses communications over the Internet network between the push server 401 and push

client 405, using various transport protocols that have additional security to facilitate secure links between the push server 401 and the push client 405.

287. Specifically, Houghton's push client software 405 on mobile terminal 404 "runs in the background and attempts to log into the push server 401 over a data network 403, normally a wireless network capable of transmitting Internet Protocol (IP) packets. The arrow A in Figure 4 illustrates an optional attempt by the push client 405 to communicate with the wireless network firewall 402 or other similar security or gateway device in order to open a connection for the push server 401 to directly contact the push client 401 as in FIG 3. If successful, operations can continue as described in FIG 3 or as below. Arrow B illustrates a login attempt. " SAMSUNG-1007, 17, FIG. 4 (reproduced below; showing communication between push client 405 and push server 401 over a wireless network).



288. Push client 405 constitutes the device link agent because it is software running on the mobile terminal device that initiates and maintains a link using “Internet technologies” between the mobile terminal 404 and the push server 401, and using this link, push server 401 pushes messages to the mobile terminal. See, *e.g.*, SAMSUNG-1007, 11, Abstract (“A server outside a wireless network security entity is adapted to push messages to a mobile terminal located in the wireless network and thereby cause programs to start to a specified operating state or, if already started, change operating state to the state indicated by the message thereby

presenting a new application view to the user of the mobile terminal. ***The mobile terminal is provided with means, such as software, which initiates and maintains contact with the server using Internet technologies.*** This mobile-initiated permanent connection allows the server to pass or push messages from outside the operator's firewall to the mobile terminal at any time.”), 20 (describing a data structure stored in mobile terminal including preferred communication protocol and other network connection parameters for facilitating connection between push client and push server), 20 (“At any time after a connection is established between the client 405 and the server 401, the client may push a message to the server or the server may send a push message to the client. This is illustrated by means of the arrow C in Figure 4. ***These messages through a previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS arrive quite rapidly.... A connectionless protocol, such as UDP/IP, may also used in cases where the connection-oriented protocol is not possible or not desirable because of the push server resources used by every static instance of a connection-oriented protocol.*** Connectionless communication is also used when a message needs to get through to the server 401 quickly, such as when occurs when the client 405 notifies the server 401 of an terminal event, user interface event or application event but the connection-oriented protocol connection is not connected or is already busy

transferring information....”); SAMSUNG-1001, 43:18-50 (interchangeably using agent and software (such as an application) running on a device).

289. For this connection, Houghton contemplates use of known transport protocols, including, e.g., “a connection-oriented protocol suitable to the wireless network such as TCP/IP,” “a connectionless protocol suitable to the wireless network 403, such as UDP/IP,” “an alternate connection-oriented protocol such as HTTP suitable for mobile push operation through more strict firewalls and proxy Internet connections,” or “a secure protocol such as HTTPS, IP-Sec, secure IP6 or a proprietary security protocol to identify the communicating parties, prevent message interception by 3rd parties, and prevent message modification by 3rd parties.” SAMSUNG-1007, 18-20; see id., claim 10 (“method as claimed in claim 9, wherein said trying of one or more other techniques and/network parameters include one or more of using an alternate push server network address which may be more available or more accessible at the present time and from the present location, using one or more alternative mobile terminal network connection settings, or variation of the parameters within the settings, using an alternate connection route through a second wireless or fixed terminal accessible to the wireless terminal over a wireless network and capable of communication with the push server, using a TCP connection, using a UDP connection, using an SSL connection, using an HTTP or tunneling HTTP

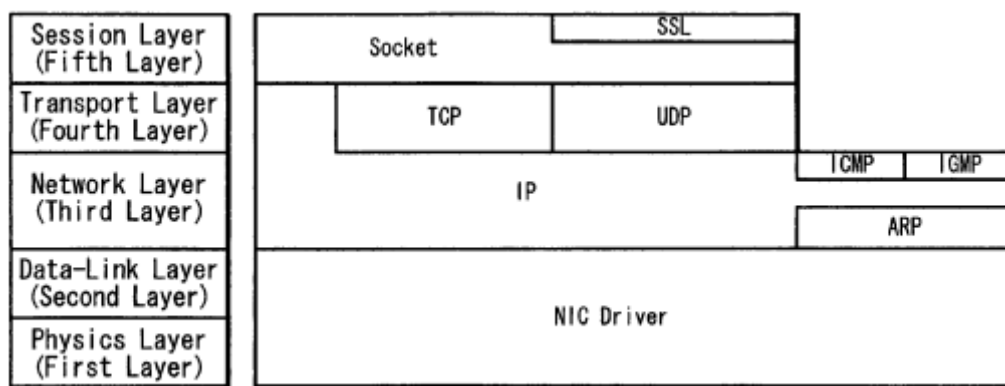
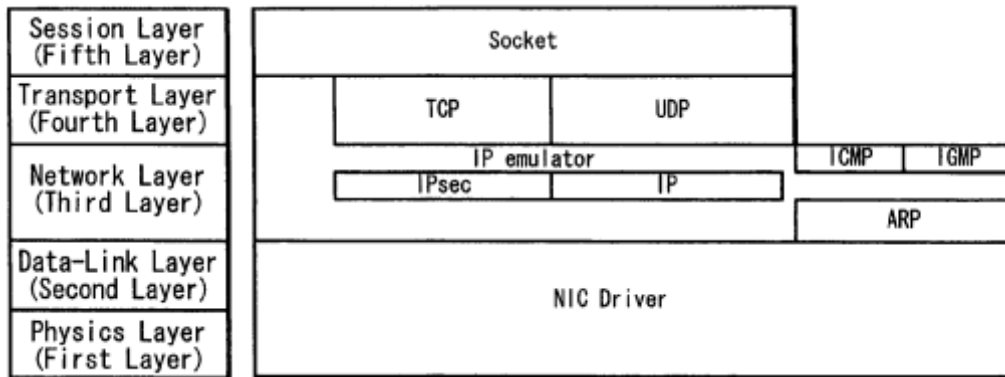
connection, using an SHTTP or tunneling HTTPS connection, using an XML-based connection over other transport protocol.”).

290. “At any time after a connection is established between the client 405 and the server 401, the client may push a message to the server or the server may send a push message to the client,” which is illustrated using arrow C in Figure 4. *Id.*, 19 (explaining using of “a previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS” for communication between push client and push server).

291. The ’192 Patent does not explicitly describe the structure of the “transport services stack” and instead only describes it functionally in terms of it facilitating transport/communication of data packets over an Internet network and between a server and a client. In some embodiments, the ’192 patent describes the communication as being secured using encryption, TLS/SSL, HTTPS, IP-Sec, or another security mechanism. SAMSUNG-1001, 90:33-61, 100:8-32, 17:13-22. The above descriptions in Houghton similarly describe the server-client communications using such secure transport protocols. From these disclosures, a POSITA would have understood or found obvious that Houghton’s push server would include software and associated hardware components/drivers that implement and facilitate use of such transport protocols to facilitate its secure communications over an Internet network with the mobile terminal 404.

292. Indeed, before the Critical Date, it was well-known for a device implementing transport protocols (as already contemplated by Houghton) to implement a transport services/protocol stack that facilitates secure network communications with other network elements (e.g., a mobile terminal).

293. For example, a prior art reference, Ozaki, describes the state of the art for TCP/IP protocol stacks that are implemented in a device's "software or firmware" and that are made up of multiple layers, including a physical layer, a data link layer, a network layer, a transport layer, a session layer, a presentation layer, and an application layer. SAMSUNG-1005, ¶¶12-14. The physical and data link layer are typically implemented using a device's network interface card (NIC) and an associated driver, and enable connecting the device to the network. *Id.*, ¶¶15-18. The network layer "defines a communication method between two ... nodes" and in the "case of TCP/IP" this layer corresponds to an IP layer, and the transport layer corresponds to the TCP layer. *Id.*, ¶¶19-21. IPsec (Internet Protocol Security) is used to encrypt communications in the network layer. *Id.*, ¶¶19-21, 8-10 (describing use of symmetric/common key encryption for IPsec encryption). The session layer "defines a procedure of a session (from start to end of a communication" and enabling a connection between applications," and Secure Socket Layer (SSL) is "conventional representative encryption communication technology" for facilitating encryption of communication in this layer. *Id.*, ¶22.



SAMSUNG-1005, FIGS. 25-26.

294. Given that Houghton already contemplates TCP/IP-based transport protocols for facilitating transport of data/messages between network entities, a POSITA would have been motivated to seek out and implement well-known TCP/IP transport stacks in the software and hardware of the push server (as disclosed in Ozaki) to enable such data transport over a network. Implementing these well-known TCP/IP transport stacks in the push server, which already supports and facilitates TCP/IP protocols and associated communications, would therefore have

amounted to implementing known techniques within known devices to achieve predictable results.

295. Moreover, as described above, a communication link is established between the push client/mobile terminal and the push server 401, and this link constitutes a secure message link because it facilitates sending of push messages between the push client 405 and push server 401 using secure/encryption protocols such as IPsec, HTTPS, and/or SSL. SAMSUNG-1007, 11, 18-20, claim 10; SAMSUNG-1001, 90:33-61, 00:8-32, 17:13-22 (similarly describing use of TLS/SSL, IPsec to secure network communications between network entities).

296. Given these disclosures in Houghton about using secure protocols such as SSL, HTTPS, and IPsec, and the well-known use of such protocols to secure TCP/IP-based network communications, a POSITA would have understood or found obvious that the transport services stack that would be implemented in Houghton's push server would use such protocols for securing the TCP/IP based communication link between the push client 405 and the push server 401.

297. A POSITA would have further understood that, in push messaging environments such as those provided in Houghton, multiple mobile terminals (and their associated push clients) would be in communication with push server 401, such that a separate secure SSL or IPsec-based TCP/IP communication link would be between the push server and each mobile terminal and its respective push client. *See*

also SAMSUNG-1007, 18, 19 (disclosing multiple “terminals” in Houghton’s environment); SAMSUNG-1037, 2-3 (describing and showing multiple user devices in communication with a push/MMS server).

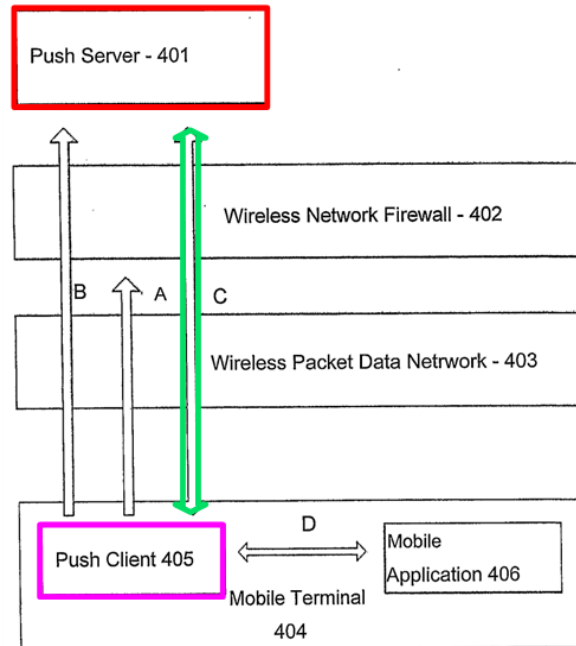
298. For the above reasons, Houghton-Munson renders obvious [1a]/[15a].

(c) *[1b]: each of the wireless end-user devices comprising multiple software components authorized to receive and process data from secure message link messages received via a device link agent on that device; / [15b]: each of the wireless end-user devices comprising multiple software components authorized to receive and process data from secure message link messages received via a device link agent on that device;*

299. Houghton-Munson renders obvious that each of the wireless end-user devices (e.g., mobile terminals 404) include multiple software components (e.g., mobile applications 406) authorized to receive and process data (e.g., application-specific data intended for the application identified by the identifier and assuming user agent supports transport of application-specific data) from secure message link messages received via a device link agent on that device.

300. As shown below and described further in Houghton, wireless end-user device (mobile terminal 404) includes multiple software components (e.g., mobile application(s) 406), and messages (e.g., command push or application command messages) are provided to the respective applications via a device link agent (e.g.,

push client 405) on the device/mobile terminal 404. SAMSUNG-1007, FIGS. 4, 8, 21-22 (“plurality of such applications), 24-25



301. Push client 405 is “implemented in the form of a software run in a processor of the mobile terminal 404” and operates “as part of a message passing system in cooperation with a Push Server 401.” SAMSUNG-1007, 16-17; *see id.* (“The push client software 405 preferably launches automatically when the mobile terminal 404 is turned on. The push client software 405 runs in the background and attempts to log into the push server 401 over a data network 403, normally a wireless network capable of transmitting Internet Protocol (IP) packets. The arrow A in Figure 4 illustrates an optional attempt by the push client 405 to communicate with the wireless network firewall 402 or other similar security or gateway device in order to open a connection for the push server 401 to directly contact the push client 401

as in FIG 3. If successful, operations can continue as described in FIG 3 or as below. Arrow B illustrates a login attempt.”); *see also* SAMSUNG-1001, 43:18-31, 11:51-57 (describing the device agent as software executing in a device processor).

302. Once a connection is established between push client 405 and push server 401, “the *server may send a push message*” to push client 405, as illustrated using arrow C above. SAMSUNG-1007, 20. Such a push message is transmitted to the push client (for delivery to a mobile application) through the “previously established, connection-oriented protocol such as TCP/IP, SSL, HTTP or HTTPS.” *Id.*; *see id.*, 34, claim 33 (“wherein *the push client* and push server establish a secure connection and *unwrap such security* on behalf of the mobile application”). Because the communication link uses SSL to secure communications transmitted over the communication link between push server 401 and mobile terminal 404, the received push message from push server constitutes a secure message link message.

303. Additionally, a POSITA would have understood or found obvious that push server 401 may communicate and facilitate transmission of push messages to multiple mobile terminals, each including its respective push client 405 and mobile applications 406. This is confirmed and corroborated by the following disclosure in Houghton: “The push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406 and direct such to the push server

401, possibly for redirection through push server API to other servers, ***mobile terminals, push clients and computing devices.***” SAMSUNG-1007, 21.

304. The received push message include “application-specific data,” such as “application commands,” “updates,” or “data messages” that are “***directed to a push application or dynamically loaded module or user interface feature from among a plurality of such applications, modules and features by means of said client-initiated or client-opened and client-maintained permanent connection.***” SAMSUNG-1007, claim 1.

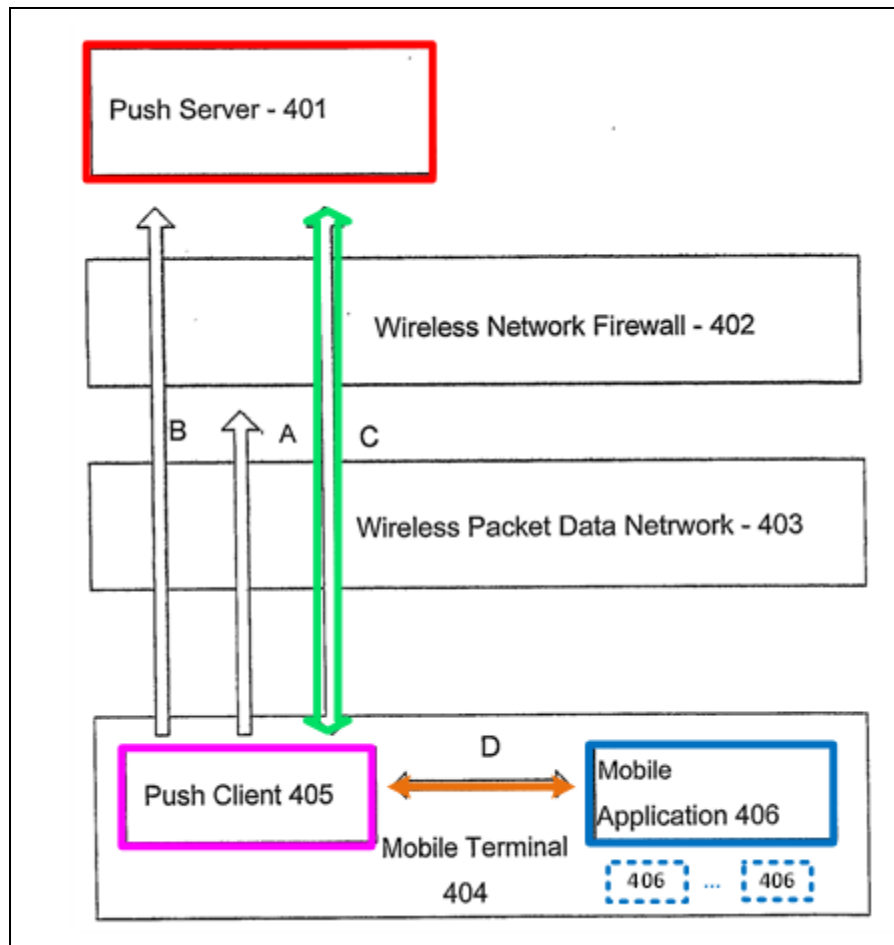
305. The received data is then processed by the receiving application to, e.g., present certain data or operate the application in a particular state. For example, “the push messages may include, for example, a message stimulating, triggering or ***commanding the launch, display or shift to foreground of a user interface feature of the terminal such as an application*** which is part of the basic terminal software or and an additional installed application beyond the base software supplied with the mobile terminal.” SAMSUNG-1007, 11.

306. In response, the “mobile terminal may spontaneously open the application, dynamically loaded module or user interface feature of the terminal thereof to a designed application view or location, present new information of the availability of new information, a new feature or operating state.” *Id.*, 12. And, if already operating, “the action of such a message would instead stimulate the

application or user interface feature to present itself to the foreground of the user interface and change state as specified in the application-specific message received from the server.” Id.

307. The push message also includes “a data packet” with “binary or text information *specifying which mobile application 406 from a plurality of such applications*” and additional “binary or text information to be passed to the ... specified mobile application.” SAMSUNG-1007, 21, 22 (“The potential applications of the COMMAND PUSH include but are not limited to mobile applications for enterprise connectivity, back office integration, database access, scientific and industrial process and equipment monitoring and control, pictures transfer, mobile gaming, application data transfer, mobile electronic commerce, application alarms, reminders, multimedia interactivity, corporate calendaring, personal messaging and group messaging, etc.”).

308. The push message is subsequently directed from push client 405 to the specified mobile application 406 via the communication path shown using arrow D in Figure 4 below:



SAMSUNG-1007, FIG. 4 (annotated), 21, 24.

309. Moreover, because the push command messages are specifically routed to a particular application from among a plurality of applications running on the terminal, and the destination application then operates on the received message data (e.g., launch application, present content, etc.), a POSITA would have understood or found obvious that the applications executing on the mobile terminal are authorized to receive and process the data included in the command push/application command messages.

310. Additionally, or alternatively, even if it is determined that Houghton does not explicitly disclose that the multiple software components (e.g., mobile applications 406) are **authorized** to receive and process data from secure message link messages, a POSITA would have found that to be obvious in view of the relevant state of the art, as explained below.

311. Before the Critical Date, it was well-known for applications that are intended recipients of push messages to register with a push server and/or push client before they can receive messages transmitted over such push frameworks. For example, in the relevant context of push communications and push environments, Lee discloses that “various push services provided to *each of applications are managed* in an integrated fashion through the integrated push service server 100 and the integrated push service agent 250” and for this reason, specification of a recipient application (similar to Houghton’s disclosures) are required in push messages to facilitate appropriate transmission of the push message to the application. SAMSUNG-1006, ¶23.

312. Similarly, Shenfield discloses another push environment in which push client includes “an application registration application provider interface adapted to register applications to said push client and further adapted to receive an application manifest for said applications.” SAMSUNG-1018, ¶¶17, 63, 106-109 (application

“registers with push client 510, providing application manifest 918 to establish a channel to a content provider for servicing” the application).

313. A POSITA would have found obvious to implement the above-described and well-known authorization or registration processes in Houghton’s system to enable registration of a particular application with a push client and the push server with which the push client interacts.

314. A POSITA would have been motivated to do so for multiple reasons, including to enable dynamic content delivery “to have information or data pushed” to devices without users of such devices having to “go and seek out that data.” *See, e.g., SAMSUNG-1018 (Shenfield)*, ¶¶3, 6.

315. A POSITA would have been further motivated to so implement application registration in Houghton’s system to ensure that data and messages are routed to the correct application and simultaneously ensure that applications have opted into receiving push data/messages (thus avoiding inundating unregistered applications with content that they have not requested or opted into receiving). As an example, different applications likely expect incoming data in a variety of formats. Whereas weather data may come in one format (a series of numbers to be parsed and organized by the application), news applications are likely to send full text sentences to be read. Neither format in this example would make sense to the other application, and therefore delivering messages to them would result in confusion. As such, it is

clear that a POSITA would have been motivated to ensure that information was routed to the correct application.

316. Moreover, because Houghton already contemplates that push client and push server coordinate on message delivery to and from applications resident on mobile terminals, implementing the well-known application registration/management teachings in push environments, within Houghton's similar push environment, would have been straightforward and a POSITA would have had a reasonable expectation of success in combining the teachings in the above-described manner.

317. Therefore, a POSITA would have found obvious that the push messages in Houghton are routed by push client 405 to applications that are authorized/registered to receive such messages.

318. For the above reasons, Houghton-Munson renders obvious [1b]/[15b].

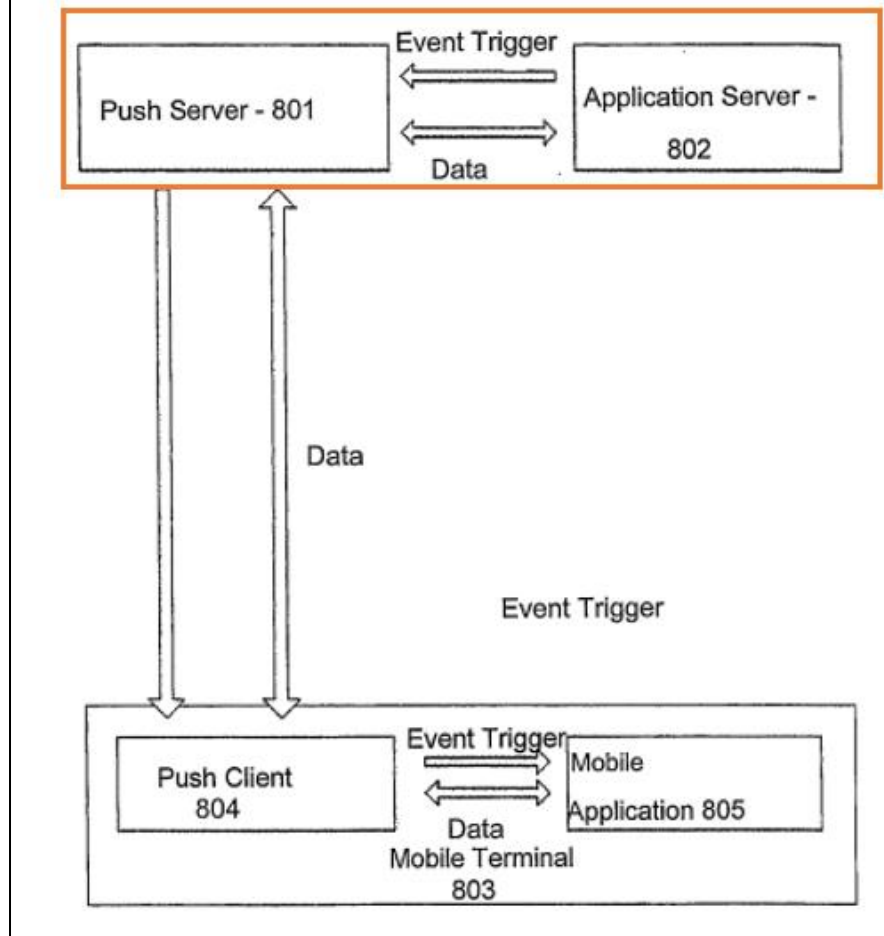
(d) *[1c1]: an interface to a network to receive network element messages from a plurality of network elements, / [15c1]: receiving network element messages from a plurality of network elements,*

319. Houghton-Munson render obvious an interface to a network to receive network element messages (e.g., an application programming interface (API)) from a plurality of network elements (e.g., application server(s) 702/802).

320. In more detail, Houghton's Figure 8 (reproduced below) illustrates an example of a "COMMAND PUSH" procedure where the push server 701 is triggered by a trigger event "*from an application server 702*" to "*push an application command message to the push client 704 and thereby initiate a mobile terminal client trigger event in a mobile application 705 from a plurality of such applications, program modules and user interface features on the terminal 705.*" SAMSUNG-1007, 21-22.

321. Fig. 8 of Houghton is shown below (annotated):

Figure 8



322. Thereafter, “a data connection between application server 802 and mobile application 805 is established with the aid of the persistent managed, tested and configured data connection in Figure 4, arrow C between push server 401/801 and push client 405/804” and “[a]n *IP or other API (application programming interface) connection between application server 802 and push server 801 simplifies the work and cost of creating application server 802 by taking care of*

network details already established for the COMMAND PUSH procedure.” Id., 22-23.

323. Houghton further discloses that, “[i]n some cases a separate server such as a telecommunication service server, gaming environment server, and original equipment manufacturer device or business information system may notify the server directly of events or messages to be sent using ***an external application programming interface (API)***. Such a device ***connected by external API*** may also receive information including the results and responses of messages which the user has replied to by terminal software or web page interaction.” SAMSUNG-1007, 14; *see id.*, 21 (“The push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406 ***and direct such to the push server 401 , possibly for redirection through push server API*** to other servers, mobile terminals, push clients and computing devices.”).

324. Moreover, the application server communicates over a network using the API or IP connection with push server 401. *See* SAMSUNG-1007, 23 (“***An IP or other API (application programming interface) connection between application server 802 and push server 801*** simplifies the work and cost of creating application server 802 ***by taking care of network*** details already established for the COMMAND PUSH procedure”).

325. Indeed, before the Critical Date, it was well known for application servers and push servers to be distributed across a network, and for such servers to communicate over a network using an interface such as an API).

326. Additionally, a POSITA would have understood or found obvious that the push environment contemplated by Houghton would include multiple different application servers, each separately communicating with the push server to exchange data/messages with one or more mobile terminals that are connected to push server 401.

327. This further corroborated by Houghton's following disclosure: "The push client 405 may in turn at any time accept data D from each of a plurality of mobile applications 406 and direct such to the push server 401, possibly ***for redirection through push server API to other servers***, mobile terminals, push clients and ***computing devices***." SAMSUNG-1007, 21.

328. Therefore, a POSITA would have understood or found obvious that Houghton's push server includes an interface to a network in the form of one or more APIs via which application-specific messages and data are received, from one or more network elements such as application servers. Moreover, a POSITA would have understood that application servers are network elements and hence messages received from such servers would constitute network element messages.

329. For the above reasons, Houghton-Munson renders obvious [1c1]/[15c1].

- (e) *[1c2]: the received network element messages comprising respective message content and requests for delivery of the respective message content to respective wireless end-user devices, the respective message content including data for, and an identification of, a respective one of the authorized software components; / [15c2]: the received network element messages comprising respective message content and requests for delivery of the respective message content to respective wireless end-user devices, the respective message content including data for, and an identification of, a respective one of the authorized software components;*

330. Houghton-Munson renders obvious that the received network element messages comprise respective message content, including data for (e.g., application-specific data and/or data related to the state in which to launch the server) and identification of a respective one of the authorized software (e.g., a specification of the application from among multiple applications to which to direct messages), and requests for delivery of the respective message content to respective wireless end-user devices (mobile terminals).

331. In more detail, in Houghton, “the push server 701 is triggered by a trigger event as for example from an application server 702 to push an application command message to the push client 704 and thereby initiate a mobile terminal client trigger event in a mobile application 705 from a plurality of such applications, program modules and user interface features on the terminal 705.” SAMSUNG-1007, 21-22.

332. Upon occurrence of this trigger, “a data connection between application server 802 and mobile application 805 is established with the aid of the persistent managed, tested and configured data connection in Figure 4, arrow C between push server 401/801 and push client 405/804.” *Id.*, 23.

333. Houghton further explains that the message pushed from push server 401/801 to push client 405/805 includes “a data packet” along with “[a]dditional binary or text information *specifying which mobile application 406 from a plurality of such applications may optionally be* included” as well as “*binary or text information to be passed to the default or specified mobile application*” and “*additional binary or text information specifying how upon receiving such a message the client 405* will notify and optionally ask the user if the action should be performed may optionally be included.” *Id.*, 21; *see id.*, claim 1 (“*pushing commands and data to a wireless terminal* from a push server” to a push client”), 8 (“When a remote program such as a push server 202 sitting inside the operator's firewall 201 and with permission to contact ports on the mobile terminal 203 sends an IP packet to the terminal (A), *the IP terminal software 204 routes the packet with it's commands or application-specific data to the mobile application 205 (B), launching the application if it is not already running.*”), 21 (“Upon receiving such a message, *appropriate automated actions and user interface media capabilities of the mobile terminal as specified in the command C from the server 401* are

executed by the push client 405. The most common case is playing a sound, asking the user if the mobile application should be launched, and if affirmative passing the command details (arrow D) to the specified mobile application, module or additional feature within the same application suite, launching the application, module or feature if needed. Such action may include optionally proceed without audio, with network downloaded audio, or without first requesting confirmation from the user as specified by user preferences and command details.”).

334. In view of the above disclosures, a POSITA would have understood that Houghton discloses or suggest that the received network element messages (push messages from application server(s) 702/802) include message content with data for the destination mobile application, as well as an identification of this application (based on the information “specifying which mobile application 407 from a plurality of such applications”).

335. However, if the claim limitation at issue requires an application identifier for the application to which the message is directed, Houghton does not expressly disclose that. Houghton, however, states that the message “specif[ies] which mobile application from a plurality of such application” to direct the message to. SAMSUNG-1007, 21.

336. Before the Critical Date, it was well-known for such information specifying/identifying a particular application to be in the form an application

identifier. SAMSUNG-1008, ¶22. For example, in the relevant context of push communications and associated environments, Lee discloses that the push message from a push server includes an “app-ID” that specifies a target application on the mobile terminal that is the destination for the message. SAMSUNG-1006, ¶¶13, 22. Similarly, TS-23.140 also contemplates that each destination application to which the push/MMS message is routed is identified using an application identifier in the message. SAMSUNG-1004, 54-56.

337. Given that Houghton routes messages to an application based on the “information specifying ... [that] application” and given the common and well-known use of application identifiers to facilitate such identification, a POSITA would have found obvious to implement Houghton’s teaching of the “information specifying” the mobile application as the application identifier for that application.

338. Implementing Houghton’s system in this manner therefore would have amounted to an implementation of a known technique (use of application identifier) to a known system (a push messaging system of Houghton) to achieve predictable results (routing messages to applications based on their identifiers, as suggested by Houghton and as was well-known in the art before the Critical Date).

339. Moreover, a POSITA would have found obvious that the received push message requests delivery of the message content to one or more mobile terminals. Specifically, and as explained above, “push server 701 is triggered by a trigger event

as for example from an application server 702 to push an application command message ***to the push client 704 and thereby initiate a mobile terminal client trigger event in a mobile application 705*** from a plurality of such applications, program modules and user interface features on the terminal 705.” SAMSUNG-1007, 21-22.

340. “The trigger mechanism may be between separate parts of a larger application or application suite, or may be an IP-triggered application launch and data transfer as in the case of mobile application launched by contact to a specified IP port.” *Id.* The “push server 901 is triggered by a trigger event of which the application server 902 is aware to send push message to the mobile terminal 905,” which causes “pushing application launch commands, lazy application updates, and lazy application data updates through push clients 904 and 905 to mobile application 907 through local command and data path D (also Figure 5, arrow A).” *Id.*, 24.

341. From these disclosures, a POSITA would have understood or found obvious that the received push messages from the application server request delivery of message content to particular mobile terminals, and in particular, to the applications resident on those mobile terminals.

342. For the above reasons, Houghton-Munson renders obvious [1c2]/[15c2].

- (f) [1d1]-[1d2]: a message buffer system including a memory and logic, the memory to buffer content from the received network element messages for which delivery is requested to a given one of the wireless end-user devices,

/ [15d1]: buffering content from the received network element messages for which delivery is requested to a given one of the wireless end-user devices;

343. Houghton-Munson render obvious a message buffer system including a memory that buffers content from the received network element messages for which delivery is requested to a given one of the wireless end-user devices.

344. In more detail, and as explained with reference to [1c1]-[1c2] (which I have incorporated by reference here), Houghton-Munson render obvious that the push server receives network element messages in the form of push messages from application servers, where delivery of such messages is requested to one or more mobile terminals/wireless end-user devices.

345. The push server further includes a message buffer system that includes logic (as described with reference to [1d3] and [1d4] below) and in view of Munson, would have also included memory that buffers content from the received push messages, as explained in §XII.A above and as further explained below.

346. Indeed, Houghton discloses that push messaging services, like “SMS (Short Message Service)” and “MMS (Multi-media Messaging Service)” “store and forward” push messages. SAMSUNG-1007, 3, FIGS. 3-4. Houghton further discloses that servers (such as push servers) “control and deliver messages and store messages when the mobile terminal is unavailable” and then later deliver those stored messages when “the terminal becomes available.” SAMSUNG-1007, 7.

347. Given (1) messages may be received by push server when the client device is unavailable and (2) Houghton’s express disclosure of storing such messages until the user becomes available, a POSITA would have understood or found obvious that the push server would include a buffer/memory for storing received messages.

348. Indeed, in the similar context of push communications, it was well-known, as taught in Lee, to store push messages “when the mobile terminal is not connected to the mobile internet service network” and “re-transmitting the stored push information when the mobile terminal is connected to the mobile internet service network” See SAMSUNG-1006, claim 14, FIG. 1 (showing a push information storage device 112 for storing push messages). Lee also contemplates that a scheduler 114 can “schedule and transmit push information push_info or retransmit the push information push_info.” See *id.*, ¶34, FIG. 34.

349. Consistent with these well-known teachings of storing push messages in push server’s memory/storage, Munson, which also enables push messages, discloses a “Series Handler Unit 224” that “receives contents from application service providers” and uses a “series buffer ... to keep the contents until all contents in a series arrive.” SAMSUNG-1017, 3:66-67, 4:1-11. Munson’s Figure 3 illustrates a structure for queuing messages in a content push service 220. SAMSUNG-1017, FIG. 3.

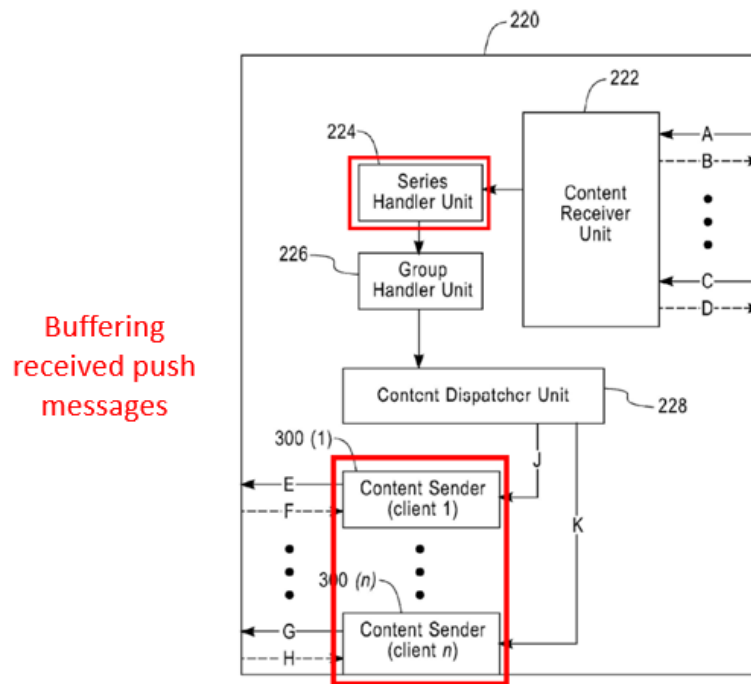
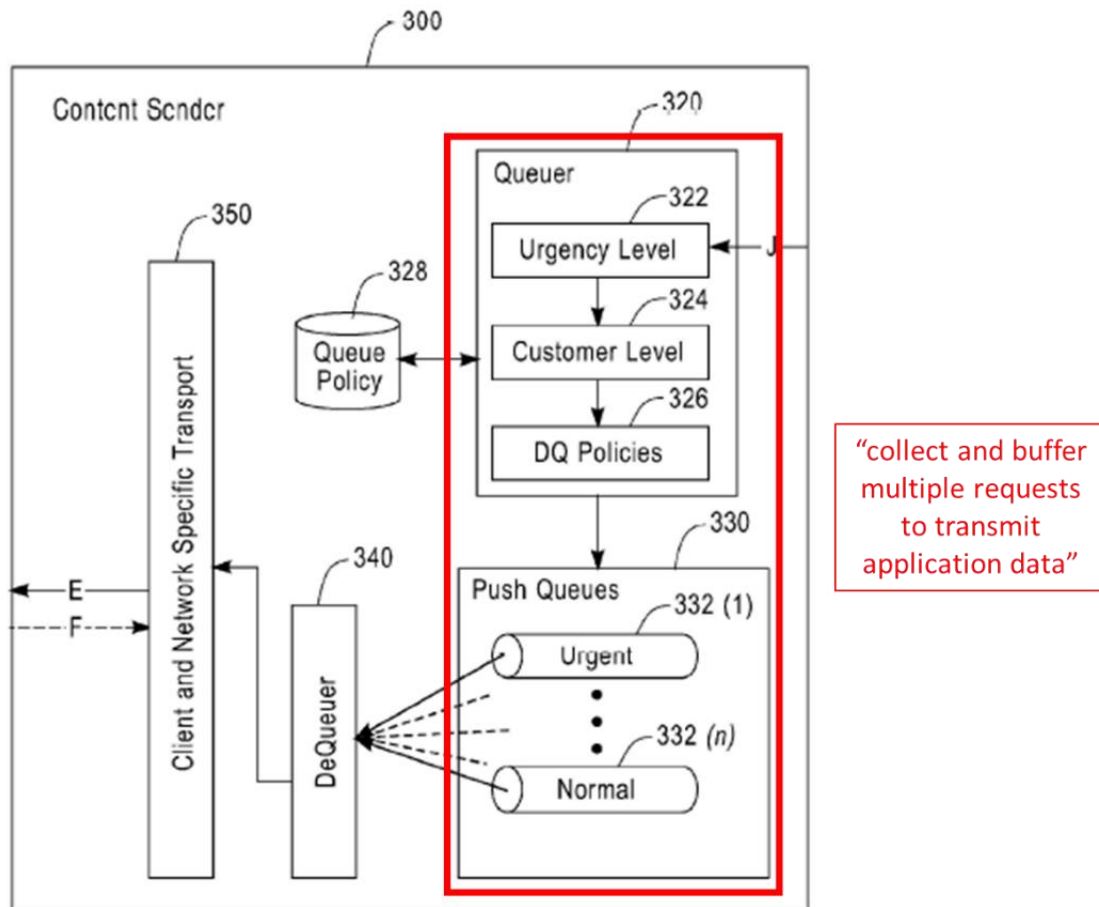


FIG. 3

SAMSUNG-1017, FIG. 3.

350. Munson additionally discloses a plurality of “Content Sender[s] 300” that each contain a “Queuer 320” and “Push Queues 330.” SAMSUNG-1017, 4:12-42, FIGS. 3, 5. Munson discloses that when pushing messages, content senders 300 check “urgency level of contents ... customer level ... [and] dequeue policy.” *Id.* Moreover, each “client device” (e.g., the Houghton-Munson mobile terminal) is assigned a content sender 300. SAMSUNG-1007, 3:40-65. Munson’s Figure 5 illustrates a structure for queuing messages in a content sender 300. SAMSUNG-1007, FIG. 5.



SAMSUNG-1017, FIG. 5.

351. In view of the above-described teachings in Munson and Houghton, and for the reasons explained in §XII.A above, a POSITA would have found obvious to implement Munson's above-described message buffering functionality and associated components as part of Houghton's push server to enable storage and later forwarding of messages to one or more mobile terminals/push client, e.g., upon occurrence of certain trigger events. Thus, in the Houghton-Munson push server, the received network elements messages would be stored in the buffer and associated components, as described in Munson and as implemented in Houghton's push server.

352. Additionally, as described below ([1d3]-[1d4] which I have incorporate here), Houghton's push server includes a message buffer system with logic that facilitate delivery of messages upon occurrence of one or more message delivery triggers.

353. For the above reasons, Houghton-Munson renders obvious [1d1]-[1d2]/[15d1].

- (g) *[1d3]: the logic to determine when one of a plurality of message delivery triggers for the given one of the wireless end-user devices has occurred, wherein for at least some of the received network element messages, the receipt of such a message by the message buffer system is not a message delivery trigger, and for at least one of the message delivery triggers, the trigger is an occurrence of an asynchronous event with time-critical messaging needs, and / [15d2]: determining when one of a plurality of message delivery triggers for the given one of the wireless end-user devices has occurred, wherein for at least some of the received network element messages, the receipt of such a message is not a message delivery trigger, and for at least one of the message delivery triggers, the trigger is an occurrence of an asynchronous event with time-critical messaging needs; and*

354. As explained below, Houghton-Munson renders obvious the message buffer system including logic that determines when one of a plurality of message delivery triggers for the given one of the wireless end-user devices has occurred, where the receipt of a message is not a message delivery trigger and one of the message delivery triggers is an occurrence of an asynchronous event with time-

critical messaging needs (e.g., a user request for a received message or the occurrence of a transaction).

355. In more detail, Houghton contemplates occurrence of trigger events to trigger sending push message to the client devices. For example, Houghton discloses that the ***“push message from the push server 401 may be triggered by any trigger event, local or remote, defined at the server. The trigger may, among others, include an alarm, notification, or measurement result received to the push server 401 from another device or system.”*** SAMSUNG-1007, 21; *see id.*, 21-22, 25-28 (describing various trigger events).

356. Houghton therefore discloses that the delivery of push messages can be triggered upon occurrence of certain trigger events and that the trigger event can be “local or remote” and may include “an alarm, notification, or measurement result received” by the push server 401 from another device or system. *See id.* Thus, a POSITA would have understood of found obvious Houghton’s system would have implemented message delivery triggers for triggering delivery of messages to the wireless end-user devices.

357. Moreover, the triggers noted above—alarm, notification, or measurement result—are message delivery triggers and do not constitute a trigger that is based on receipt of the message by the push server.

358. Additionally, Houghton discloses or renders obvious that the message delivery triggers include an occurrence of “an asynchronous event with time-critical messaging needs,” consistent with how this claim phrase is described in the ’192 patent’s specification. Specifically, and as explained with reference to this limitation in §VIII.A.1 above, the ’192 specification explains that an asynchronous event with time-critical messaging needs encompasses a “user request” for the delivery of the message or the occurrence of a transaction. SAMSUNG-1001, 38:50-64.

359. Consistent with this description, Houghton renders obvious a message delivery trigger that is an occurrence of an asynchronous event with time-critical messaging needs. Specifically, and as noted in the above-cited disclosures, the trigger event can be an alarm or notification received from another device.

360. Houghton further contemplates that message delivery can be adjusted based on whether the message to be delivered is time-critical or “non-time-critical.” *See, e.g.*, SAMSUNG-1007, 23-24 (“[t]he push server 501 on becoming aware of such a situation may inform (arrow D) the application server 502 of the availability of a more desirable network connection (C plus A) to the client and on behalf of the application server 502 or independently decide to push additional commands and data to the wireless terminal using this connection. Further, the push client in the broadband access point has the additional capability to route such information to the wireless terminal. ***The result benefits customers using the wireless terminal in that***

data is managed and updated automatically for their mobile applications in the most cost efficient manner for non-time-critical capabilities such as lazy application code update and lazy application data update. While such alternative routing may also be achieved through normal IP routing and path cost without the presence in the broadband access point 509 of push-aware client software, *the push server 501 and application server 502 would in this known-in-the-art implementation not be aware of when the low cost alternative route was available and thus able to at such time trigger the push of lazy application code and data installation and updates.”*). A POSITA would have therefore found obvious to modify message delivery based on time criticality of the message and therefore, adjust message delivery and the associated trigger in this context.

361. For example, Munson discloses that its push system provides “*asynchronous content push* (i.e., pushing a content) to clients on diverse wireless networks” “according to a schedule of time *or event*” such as coordinating pushes “for a system maintenance purpose during off-peak hours.” SAMSUNG-1017, ¶¶40, 44.

362. Moreover, in Houghton, the push client can send a message based on a user request to the push server, requesting delivery of a push message. Specifically, “IP push command messages” may be triggered when “push client makes” push server aware of certain “client-side events,” which in turn trigger the push server to

deliver push messages intended for particular mobile application(s) 406 residing on the mobile terminal. SAMSUNG-1006, 14, 21. For example, the “client-side events” including, among others, “**creation of photographs, video or other media,**” “**video game actions or events,**” and “**messaging actions**” are triggers that “**push client makes the server aware of**” so that the server pushes a message to the mobile terminal and a particular application residing therein. *Id.* Houghton further discloses that the trigger events include “**remote application user** or application server **event, remote user action.**” *Id.*, claim 22.

363. Given the above disclosures, a POSITA would have understood or found obvious that these client-side events are user requested events or transactions that occur, e.g., as a result of transactions and events occurring on a mobile terminal, including user actions taken with respect to creating media, gaming events, or messaging.

364. A POSITA would have thus understood or found obvious that such user actions or transactions trigger delivery of messages and therefore, constitute message delivery triggers that constitute an asynchronous event with time-critical messaging needs, as that phrase would be understood in the context of the '192 Patent.

365. Because push server waits for these triggers to occur before it deliver the message to push clients, a POSITA would have understood or found obvious that

push server includes logic that determines when one (or more) of these message delivery triggers for the particular terminal/end-user device has occurred and if so, delivering the messages.

366. For the above reasons, Houghton-Munson renders obvious [1d3]/[15d2].

- (h) *[1d4]: upon determining that one of the message delivery triggers has occurred, the logic further to supply one or more messages comprising the buffered content to the transport services stack for delivery on the secure message link maintained between the transport services stack and a device link agent on the given one of the wireless end-user devices. / [15d3]: upon determining that one of the message delivery triggers has occurred, supplying one or more messages comprising the buffered content for delivery on the secure message link maintained between the message link server and a device link agent on the given one of the wireless end-user devices.*

367. Houghton-Munson render obvious that, upon determining that one of the message delivery triggers has occurred (see above analysis for [1d3]/[15d2]), the logic further to supply one or more messages comprising the buffered content (see above analysis for [1d1]-[1d2] relating to storage of received push messages) to the transport services stack for delivery on the secure message link maintained between the transport services stack and a device link agent on the given one of the wireless end-user devices.

368. As described in [1a]-[1b], Houghton's push server would have included a transport services stack that facilitates a secure message link between the push server and the push client. Houghton further discloses, in view of Munson and as described with reference to [1d1]-[1d2], to store the received push messages in push server's memory.

369. In the combination, the stored push messages (and the associated data/content) would be delivered to the push client upon the occurrence of one or more message delivery triggers, as described with reference to [1d3] (which I have incorporated by reference here).

370. Such message delivery would happen using the secure message link already established/maintained between the push server and the push client of the mobile terminal. SAMSUNG-1007, claim 1 ("the push server sends, in response to predetermined trigger events, at any time push application installation, updates, commands or data messages to be directed to a push application or dynamically loaded module or user interface feature from among a plurality of such applications, modules and features by means of said client-initiated or client-opened and client-maintained permanent connection").

371. For the above reasons, Houghton-Munson renders obvious [1d4]/[15d3].

2. Claim 5

372. Houghton-Munson render obvious the message link server (as described with reference to claims 1 and 15 above) and further renders obvious [5a]-[5c], as described below.

(i) *[5a]: wherein the transport services stack is further to receive, over each of the respective secure message links, upload messages forwarded by the respective device link agents from at least a subset of the device software components,*

[5b]: each of the upload messages identifying a corresponding one of the network elements to which the device respective software component has requested delivery,

373. Houghton-Munson renders obvious that push server's transport services stack (as described with reference to [1a] above) receives, over each of the respective secure message links, upload messages forwarded by the respective device link agents (e.g., one or more push clients) from at least a subset of the device software components (e.g., one or more applications).

374. In more detail, Houghton discloses that push client 405 “at any time accept[s] ***data D from each of a plurality of mobile applications 406 and direct[s] such to the push server 401***, possibly for redirection through push server API to other servers, mobile terminals, push clients and computing devices.” SAMSUNG-

1007, 27; *see id.*, 28 (“***the client 405 may also send a message to the server 401 (step 633)***), such as a command execution acknowledgement or result”).

375. Moreover, given that Houghton’s push server includes a transport services stack (as described with reference to [1a] above), including API or IP-based connections between the push server and the application server, and includes similar connections between mobile terminals and the push server, a POSITA would have understood or found obvious that push server’s transport services stack would receive, over the secure message link (as described with reference to claim 1), messages including data (i.e., upload messages) forwarded by the push client(s) (i.e., respective device link agents) from at least a subset of the applications 805 (i.e., device software components).

376. Additionally, as described above, each such message would be directed to an application server, and thus, a POSITA would have understood or found obvious that each such message would identify the particular application server (i.e., network element) to which the initiating software application (i.e., device respective software component) requested delivery.

- (j) [5c]: *the network server system using the interface to a network to deliver content from the upload messages to the respective identified network elements.*

377. As described with reference to [5a]-[5b] and claim 1 (which I have incorporated by reference here), Houghton's push server (i.e., the network server system) includes interfaces that facilitate communication with network elements (i.e., interface to a network), including mobile terminals and application servers, to deliver messages and associated data (i.e., content from the upload messages) to one or more application servers (i.e., the respective identified network element, of which there can be more than one).

378. Moreover, as described with reference to [5a]-[5b] (which I have incorporated by reference here), push server uses an API (application programming interface) or IP connection to facilitate communication between the push server and the application server, and a POSITA would have understood that this interface/network connection would have been used for communicating the upload messages to the application server(s) from the push server.⁵

379. For the above reasons, TS-23.140 renders obvious [5a]-[5c].

⁵ Note that claim 1 does not recite a "network server system" and thus there is no antecedent basis for "the network server system" in claim 5. For purposes of claim interpretation, I have interpreted "network server system" as referring to the message link server of claim 1 and its interface to the network, as recited in claim 1.

3. Claim 6

- (k) [6]: *The message link server of claim 1, wherein at least one of the one or more messages for delivery by the transport services stack comprises multiple identifier/data pairs.*

380. Houghton-Munson renders obvious that at least one of the one or more messages for delivery by the transport services stack comprises multiple identifier/data pairs. For example, as described with reference to limitation [1b] and [1c2] (which I have incorporated by reference here), push messages would be received, by the transport services stack implemented as part of Houghton's push server and from different network elements (e.g., application server or other push client resident on a respective mobile terminal), and each such message intended for an application executing on one of the mobile terminals would include data and corresponding identification of the intended software application.

381. Moreover, a POSITA would have recognized or found obvious that messages received from the push server would have been directed to multiple applications from among the "plurality" of applications ("**multiple identifier/data pairs**") because Houghton discloses various "actions" that occur across applications when receiving a push message. SAMSUNG-1007, 21. This is corroborated by Houghton, which discloses that, in some cases, the "packaging of mobile applications involves combining multiple applications delivered in a single bundle"

(e.g., a “bundle” or “suite” of applications would receive messages packaged together – a message containing “**multiple identifier/data pairs**”). SAMSUNG-1007, 12.

382. This therefore demonstrates that the receipt of multiple identifier/data pairs in the one or more messages for delivery by the transport services stack of the push server.

383. For the above reasons, Houghton-Munson renders obvious claim 6.

4. Claim 7

(l) [7]: *The message link server of claim 1, the device messaging agent on at least one of the wireless end-user devices further to initiate the respective secure Internet data message link to the transport services stack.*

384. In Houghton, “the mobile terminal is provided with means, such as software, which **initiates and maintains contact with the server using Internet technologies**” and this “**mobile-initiated permanent IP connection** allows the server to pass or push messages from outside the operator’s firewall to the mobile terminal at any time.” SAMSUNG-1007, 11; see *id.*, claim 1, 25 (describing push client initiating and maintaining an IP connection link with push server).

385. Additionally, as described in [1a] (which I have incorporated by reference here), SSL is used for secure messaging between push client and push server. Thus, a POSITA would have found obvious for the SSL-based

communication link between push client and push server to be initiated by the client device. Indeed, before the Critical Date, it was well-known for SSL-based communications between a mobile terminal and a server to be initiated by the client device. *See, e.g.*, SAMSUNG-1012, 29:31-30:24 (describing that SSL/TLS requires a handshake phase and a data transfer phase, and in the handshake phase, the ***client side initiates a handshake*** by sending the server a ClientHello message).

386. Given that push client logs into the push server, facilitates communication between these entities using the established IP connection, and uses SSL-protocols that are initiated at the client side/mobile terminal, a POSITA would have found obvious that push client (i.e., the device messaging agent on at least one of the wireless end-user devices) initiates the respective secure Internet Data message link (message link on which data is sent over the Internet) to push server's transport services stack (as described with reference to [1a] above).

387. For the above reasons, Houghton-Munson renders obvious claim 7.

5. Claim 9

(m) [9]: *The message link server of claim 1, wherein one of the message delivery triggers is the expiration of a periodic timer.*

388. Houghton-Munson renders obvious that one of the message delivery triggers is the expiration of a periodic timer.

389. Munson discloses that “content can be pushed according to a schedule of time or event” (“***message delivery trigger***”), and provides an example where a “group push” is performed “during off-peak hours to take advantage of lower rates.” SAMSUNG-1017, [0044].

390. Houghton also contemplates multiple message delivery triggers (SAMSUNG-1007, 21) and additionally discloses a periodic message (sent at “the expiration of a timer”) that ensure that devices in the communication path (i.e., push client at mobile terminal and push server) “do not time expire the connection.” *Id.*, 26, 19 (“Use of a periodic message from the client to the server or server to the client to the effect that client socket, server socket, intermediate Network Address Translation (NAT) devices, intermediate firewalls, and intermediate gateway devices do not time expire the connection.”), 11 (“In an embodiment of the invention, the mobile terminal maintains contact with the server by periodically testing the connection with a suitable keepalive message, a “ping”, which resets timers in intervening operator stateful firewalls and/or NAT (network address translation) firewalls. The mobile terminal may also automatically re-establish the connection when the connection gets broken and adjust its connection settings based on past experience with failure to connect and such broken connections in a given mobile network.”).

391. Given Houghton's disclosures of delivery triggers and Munson's use of periodic timers for message delivery, a POSITA would have been motivated and would have had a reasonable expectation of success in implementing Munson's teaching of using periodic timers (and expiration of the same) as a message delivery trigger, within Houghton's system that already contemplates using periodic timers. A POSITA would have been motivated to use periodic timers as a message delivery trigger, as contemplated by Munson, to enable more efficient use of network resources and avoid repeated message requests that can impact performance of network communications.

392. For the above reasons, Houghton-Munson renders obvious claim 9.

6. Claims 11-12

- (n) *[11]: The message link server of claim 1, wherein one of the message delivery triggers is the receipt of a transmission on the respective secure message link from the device link agent of the given one of the wireless end-user devices, or a response generated to a transmission received from that device link agent. / [12]: The message link server of claim 11, wherein the transmission is a heartbeat message generated by the given device link agent, or a request received from the given device link agent.*

393. As described below, Houghton provides multiple examples of message delivery triggers that include "the receipt of a transmission on the respective secure message link from the device link agent of the given one of the wireless end-user

devices,” where the transmission is a request received from the given device link agent.”

394. For example, Houghton describes that “client side events” can result in “pushing application commands.” SAMSUNG-1007, 14, 21, 16 (“Figure 4 illustrates an exemplary implementation of the system *for pushing application commands to a mobile terminal based on* server side and *client side events*”). Specifically, Houghton also describes that the “client” can notify the “server” of a “terminal event” that triggers a “return message.” SAMSUNG-1007, 20-21.

395. Additionally, Houghton contemplates that a “connectionless protocol, such as UDP/IP,” may be used “when the *client 405 notifies the server 401* of a[] terminal event, user interface event or application event” and the *push server “in return push[es] to the client 405 a service triggered by the event....* The sending of such a return message is frequently time critical and the fastest available combination of techniques will be used.” SAMSUNG-1007, 21.

396. In the context of established IP connections, the “push client 405 may monitor the local resources for a state change (a trigger event) in step 641. If a state change of other event is detected in step 642, the push client 405 may send a message to the server 401 (step 643).” SAMSUNG-1007, 26.

397. Moreover, as described above with reference to claim 9, an example message trigger includes periodic polling to ensure that devices in the

communication path (push client at mobile terminal and push server) “do not time expire the connection,” which a POSITA would have recognized or found obvious as being akin to a heartbeat message generated by the given device link agent. SAMSUNG-1007, 19, 21, 26; SAMSUNG-1001, 70:6-71:16.

398. Moreover, as described above with reference to claim 9, an example message trigger includes periodic polling to ensure that devices in the communication path (push client at mobile terminal and push server) “do not time expire the connection,” which a POSITA would have recognized or found obvious to be a heartbeat message generated by the given device link agent. SAMSUNG-1007, 19, 21, 26; SAMSUNG-1001, 70:6-71:16.

399. For the above reasons, Houghton-Munson renders obvious claims 11-12.

7. Claim 13

- (o) *[13]: The message link server of claim 1, wherein one of the message delivery triggers is the receipt of a particular network element message from one of the network elements.*

400. Houghton-Munson render obvious that one of the message delivery triggers is the receipt of a particular network element message from one of the network elements (e.g., a message from a particular application server). *See, e.g.*, SAMSUNG-1007, 21-22 (“Figure 7 illustrates an example of [a] ‘COMMAND

PUSH’ procedure *wherein the push server 701 is triggered by a trigger event as for example from an application server 702 to push an application command message to the push client 704* and thereby initiate a mobile terminal client trigger event in a mobile application 705 from a plurality of such applications, program modules and user interface features on the terminal 705.”).

401. For the above reasons, Houghton-Munson renders obvious claim 13.

XIII. GROUND 6: CLAIMS 2 AND 3 ARE RENDERED OBVIOUS BY HOUGHTON, MUNSON, AND SHEN

A. Combination of Houghton, Munson, and Shen

402. As described in §XII.A, a POSITA would have found obvious to combine the teachings of Houghton and Munson. A POSITA would have further found obvious to combine the teachings of these references with those of Shen.

403. Like Houghton and Munson, Shen discloses a push messaging system in which a push server (e.g., an MMS Relay/Server) receives and stores messages before transmitting those messages to MMS User Agents. SAMSUNG-1004, pp. 16-18, 54-56; SAMSUNG-1014, [29]-[34], FIGS. 1-2, 5. Shen, however, explains that the store and forward functionality of this server exposes a “potential security problem” given that such systems are “not an end-to-end solution ...” SAMSUNG-1014, [0004].

404. To overcome this “security problem,” Shen’s invention discloses that “an authentication and key management module 502 is added in the MMS gateway to distribute symmetric keys to users and a certificate management module 504 is added in the user device,” as shown below:

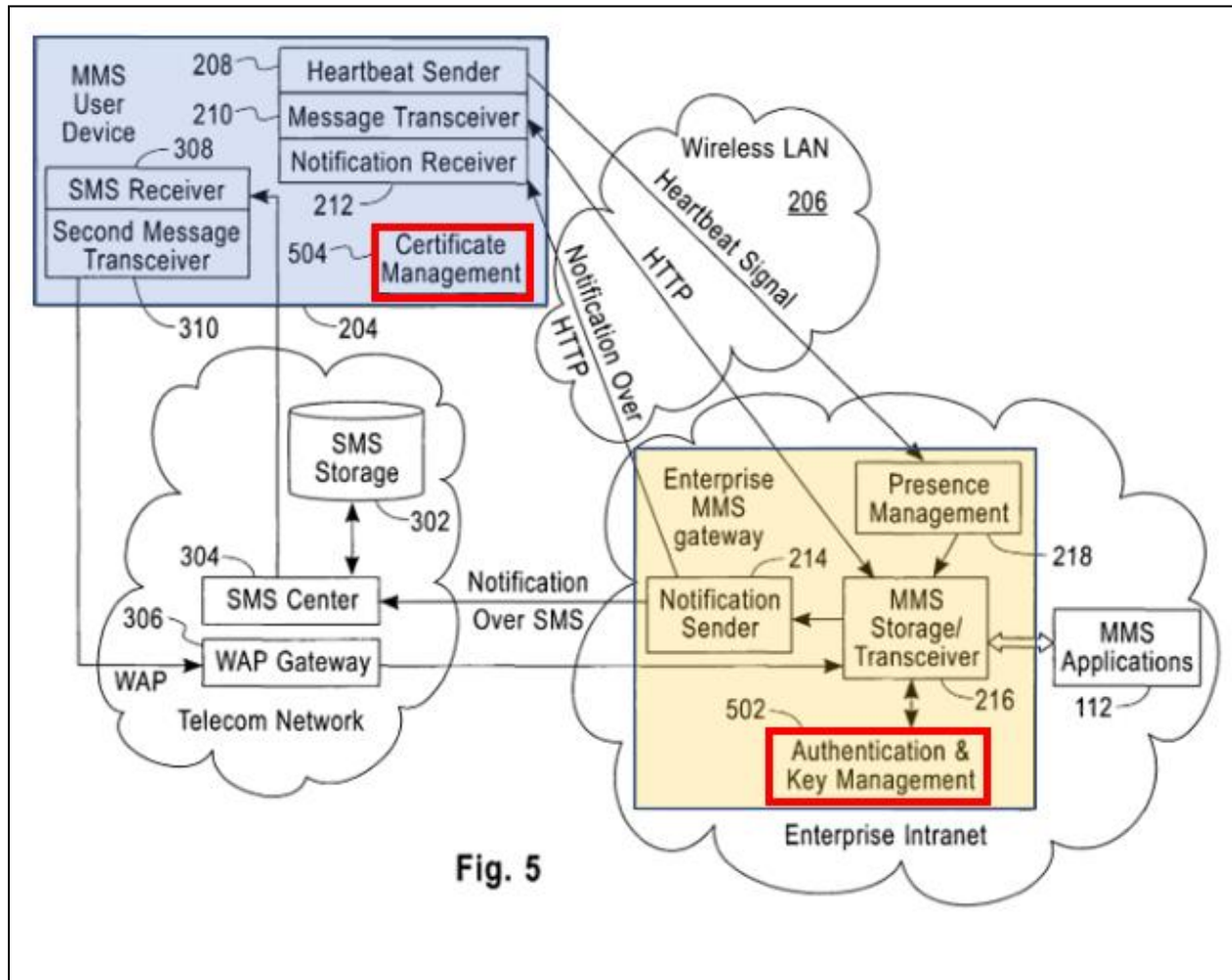


Fig. 5

405. Using these modules, Shen discloses generation of a symmetric key that is then used by the MMS Relay/Server to encrypt data communications transmitted to the MMS User Device/Agent. SAMSUNG-1014, [56]-[60].

406. Combining the above-described teaching of Shen with those of Houghton and Munson would have been obvious to a POSITA at least because such a combination would have merely involved (1) combining prior art elements according to known methods to yield predictable results and (2) the use of known technique to improve similar devices (methods, or products) in the same way.

407. Indeed, each of Houghton, Munson, and Shen provide similar push messaging architectures whereby push clients on the mobile terminal/user device (referred to as MMS User Agent in Shen) communicate with a push server (referred to as MMS Relay/Server in Shen).

408. And given the “security problems” in data communications occurring in such environments, a POSITA would have been motivated to implement known techniques, as disclosed in Shen, to solve these problems—namely implementing an “authentication and key management module 502” at Houghton’s push server that would generate and “distribute symmetric keys to users” and would use this symmetric key to encrypt data transmitted to “the user device.” SAMSUNG-1014, [0054]-[0060].

409. A POSITA would have been motivated to so combine the teachings of these references to achieve a push messaging system “having improved security,” resulting from “provi[sion] [of] an end-to-end security solution for” the underlying messaging applications, which is particularly beneficial in the context of “enterprise

applications” that are also contemplated by Houghton’s system. SAMSUNG-1014, [0017], [0021] (describing additional advantages stemming from Shen’s invention); SAMSUNG-1007, 22 (describing applicability of push messaging in the context of enterprise applications). Moreover, it was well known before the Critical Date for push serves to encrypt content of messages sent to push clients/client devices. Chou is one example of a push message system that uses a “WAP gateway” (e.g., a network message server) that encrypts content received from a “content server” for transmission to a “mobile client” that then “decodes the response.” SAMSUNG-1009, ¶¶[0054]-[0060]. Rakic and Shen provide additional examples of a push server encrypting message traffic. SAMSUNG-1015, ¶¶[0065]-[0068]; SAMSUNG-1014, ¶¶[0054]-[0060].

410. So configuring the Houghton-Munson system to implement Shen’s above-described modules and associated encryption key generation functionality to facilitate encryption of messages (as described above) would have amounted to the use of a known technique to improve similar devices in the same way, and combining prior art elements according to known methods to yield predictable results.

411. Given that Houghton and Munson’s system described message delivery in a similar push messaging environment (as in Shen) involving a similar communication between push clients/MMS User Agents and the push server/MMS Relay/Server, a POSITA would have found it straightforward—with a reasonable

expectation of success—to modify and/or program Houghton’s push server and push client to implement Shen’s above-described modules and associated functionality for generating an encryption key and encrypting communications between the push client and the push server using that key.

412. Additionally, the resulting system’s elements would each perform functions they were known to perform prior to the combination—i.e., Houghton-Munson’s system would continue to facilitate communication of push messages between push servers and push clients, and Shen’s teachings (as implemented in combination) would facilitate encryption by the push server for messages that are then sent to a push client.

413. As described below, the combination of Houghton, Munson, and Shen (referred to herein as Houghton-Munson-Shen) renders obvious claims 2 and 3.

B. Claim Analysis

1. Claim 2

(a) *[2]: The message link server of claim 1, further comprising an encrypt function to encrypt the one or more messages supplied to the transport services stack for delivery on the secure message link maintained between the message link server and the device link agent on the given one of the wireless end-user devices*

414. Houghton-Munson-Shen renders obvious an encrypt function to encrypt the one or more messages supplied to the transport services stack for delivery

on the secure message link maintained between the message link server and the device link agent on the given one of the wireless end-user devices.

415. As described with reference to [1a] and [1d4] (which I have incorporated by reference here), Houghton discloses that messages would be supplied to the push server's transport services stack for delivery on the secure message link between the push server and the push client (resident on a particular user device/mobile terminal).

416. Additionally, as described in §XIII.A above (which I have incorporated by reference here), the combination would have implemented Shen's teachings of authentication and key management module at Houghton's push server, which would have facilitated generation of a shared key between the mobile terminal/push client and the push server, where such shared key can be used to "encrypt[] MMS messages with the shared secret key." SAMSUNG-1014, [0059].

417. In this manner, Houghton-Munson-Shen would have implemented an encryption function that encrypts the messages sent by the push server to the push client using the shared keys (where such message would be delivered by the transport services stack on the secure link to the push client, as described above). *Id.*

418. For the above reasons, Houghton-Munson-Shen renders obvious claim 2.

2. Claim 3

- (b) *[3]: The message link server of claim 2, wherein the encrypted one or more messages are transported to the device link agent on the given one of the wireless end-user devices using one or more of encryption on the transport services stack, IP (Internet Protocol) layer encryption, and tunneling.*

419. Houghton-Munson-Shen renders obvious the encrypted one or more messages are transported to the device link agent on the given one of the wireless end-user devices using one or more of encryption on the transport services stack, IP (Internet Protocol) layer encryption, and tunneling.

420. As described with respect to claim 2 (which I have incorporated by reference here), Houghton-Munson-Shen renders obvious encrypting messages and transporting the encrypted messages to a particular push client on a mobile terminal. Additionally, as explained with reference to [1a] (which I have incorporated by reference here), the combination renders obvious encryption on the transport services stack in view of the above-described teachings of SSL, HTTPS, and IPsec on the transport/communication link between the push server and the push client (i.e., encryption on the transport services stack and IP layer encryption). As Ozaki confirms, IPsec is encryption at the IP layer. SAMSUNG-1005, ¶¶19-21, 8-10.

421. For the above reasons, Houghton-Munson-Shen renders obvious claim 3.

XIV. GROUND 7: CLAIM 4 IS RENDERED OBVIOUS BY HOUGHTON, MUNSON, AND ELLISON

A. Combination of Houghton, Munson, and Ellison

422. A POSITA would have found obvious to combine the teachings of Houghton and Munson with those of Ellison such that Ellison's secure platform would have been incorporated into Houghton's mobile terminals. SAMSUNG-1007, FIG. 2-4, 8; SAMSUNG-1019, 8:25-32, 8:66-67, 9:1-6, 9:28-62, FIG. 2.

423. As further explained below, a POSITA would have combined the teachings of these references to improve the security of Houghton's mobile terminal. Such a combination would ensure that malicious applications could not easily gain access to stored message content after the encryption used during transmission had been removed by the intended user device.

424. A POSITA would have recognized that Ellison's secure platform would have improved the security of Houghton's push message system by improving device security (i.e., entry points to the network). SAMSUNG-1019, 8:25-32, 8:66-67, 9:1-6, 9:28-62, FIG. 2.

425. The increased confidence in device and/or network security provided by Ellison would have enabled a POSITA to Houghton's push service to additional application servers and clients. SAMSUNG-1019, 8:25-32, 8:66-67, 9:1-6, 9:28-62, FIG. 2.

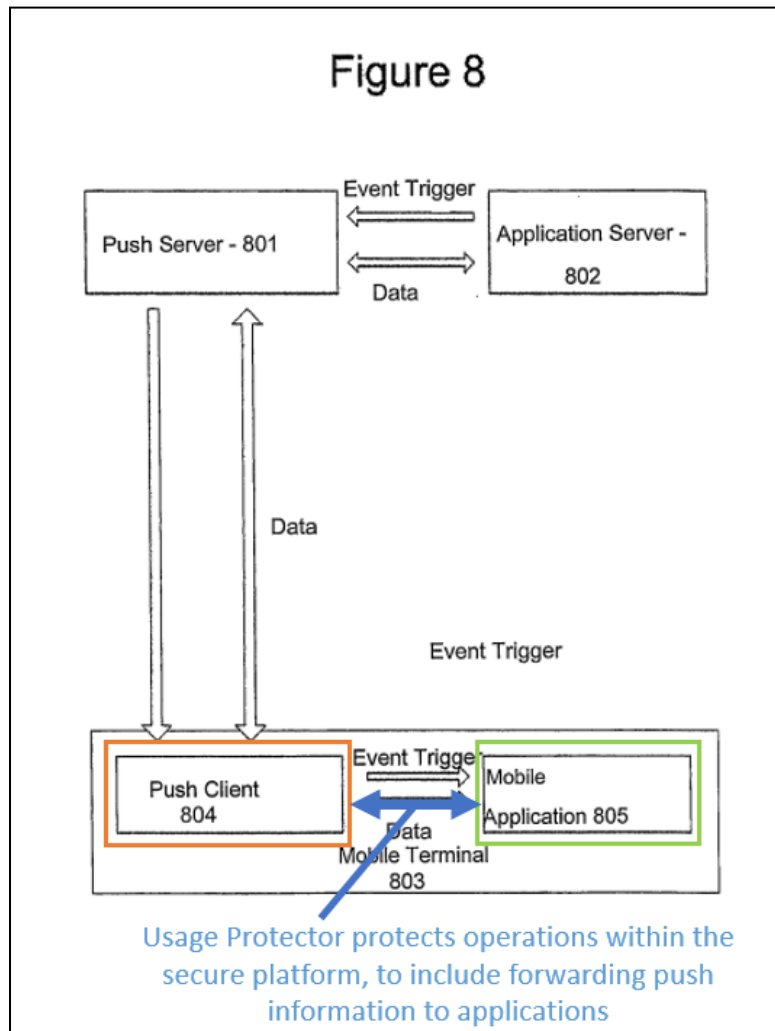
426. Moreover, device level security improvements are an efficient, beneficial option for improving network security, which would have been a natural solution for Houghton's system, which is concerned with secure messaging in the push messaging context. SAMSUNG-1019, 8:25-32, 8:66-67, 9:1-6, 9:28-62, FIG. 2.

427. As explained below in more detail, combining the teachings of Houghton, Munson, and Ellison would have been obvious at least because such a combination would have merely involved (1) combining prior art elements according to known methods to yield predictable results.

428. Incorporating Ellison's security techniques into Houghton's mobile terminals would also have been predictable and foreseeable with a reasonable expectation of success because Ellison describes that its techniques can be implemented in "computer system[s]" that contain a "processor." SAMSUNG-1019, 5:11-16.

429. A POSITA would have recognized or found obvious that the examples of "mobile terminals" provided in Houghton include a "processor" capable of implementing Ellison's techniques as it was known in the art that mobile terminals contained processors. SAMSUNG-1007, 16.

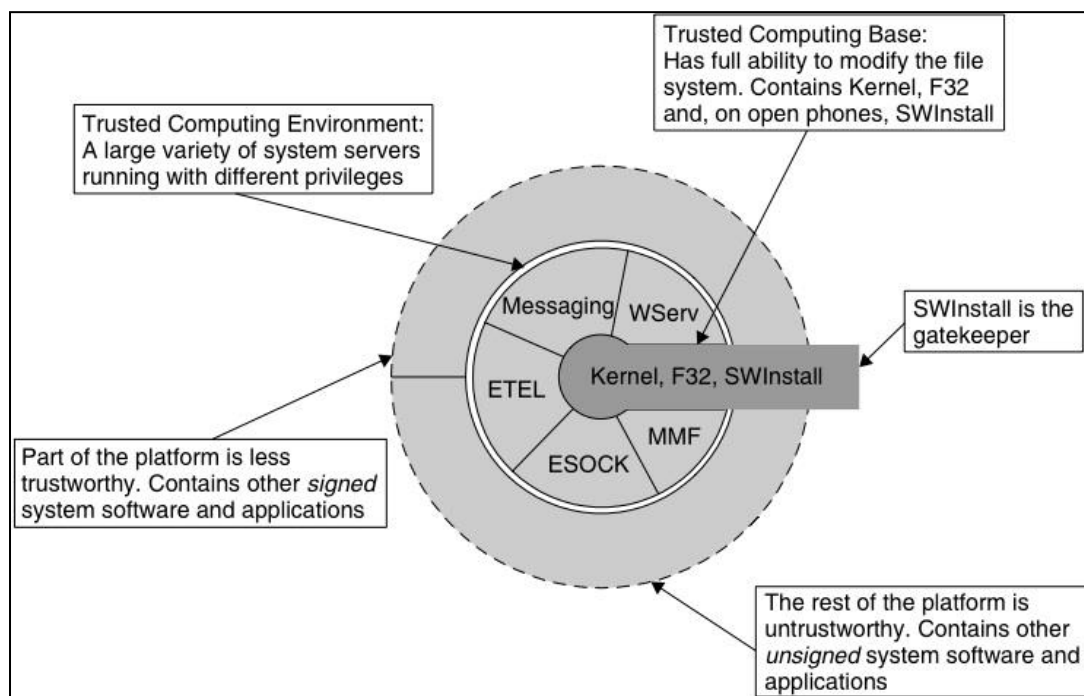
430. In an example of the combined system, illustrated below in Houghton's Figure 7, Ellison's secure platform would have been incorporated into Houghton's mobile terminals. SAMSUNG-1019, 8:25-32, 8:66-67, 9:1-6, 9:28-62, FIG. 2.



SAMSUNG-1007, FIG. 8 (modified to incorporate Ellison's teachings).

431. Moreover, in the context of mobile devices and their associated operating systems, it was well-known before the Critical Date to implement more secure systems, with rings or tiers of protection around different device components.

For example, in the early 2000s, the Symbian mobile operating system (B. Li, E. Reshetova, T. Aura, “Symbian OS platform security model”, LOGIN: Magazine, August 2010—SAMSUNG-1035; Craig Heath, Symbian OS Platform Security: Software Development Using the Symbian OS Security Architecture (Wiley, 2006)) separated processes into three “Trust Tiers”: The Trusted Computing Base (TCB), The Trusted Computing Environment (TCE), and Applications, as illustrated below:



432. As shown above, the first two layers/tiers shown were primarily dedicated to components of the Operating System with relatively higher operating privileged, followed by the Applications tier with applications having relatively lower privileges. The first two tiers included the Trusted Computing Base and the Trusted Computing Environment. As shown, the Trusted Computing Environment includes, among other services, a Messaging service. The Application Tier could be

further subdivided into Signed and Unsigned applications. Signed applications could interact with the TCE and have this layer execute higher privileged requests, whereas Unsigned applications could not access that same functionality.

433. Thus, it was well-known for operating systems to include the messaging services and clients/agents (e.g., push clients) responsible for provision of those services, within the trusted computing environment, and for applications (including those that would be intended recipients of those messages) to be tiered/organized outside this environment.

434. Given that Houghton contemplates mobile terminals with a push client that communicates with applications on the device, the well-known need to protect computing environments (including components providing essential services such as messaging) from other applications/components of the environment, and Ellison's own disclosures confirming the same, a POSITA would have been motivated to similarly enable such improved security within the push messaging environment in Houghton by implementing Ellison's teachings of secure environments, such that the push client is part of the device's secure execution environment and enables communications with applications that reside outside this trusted/secure execution environment. Such a combination would have been straightforward to implement with a reasonable expectation of success given the well-known use of such techniques in

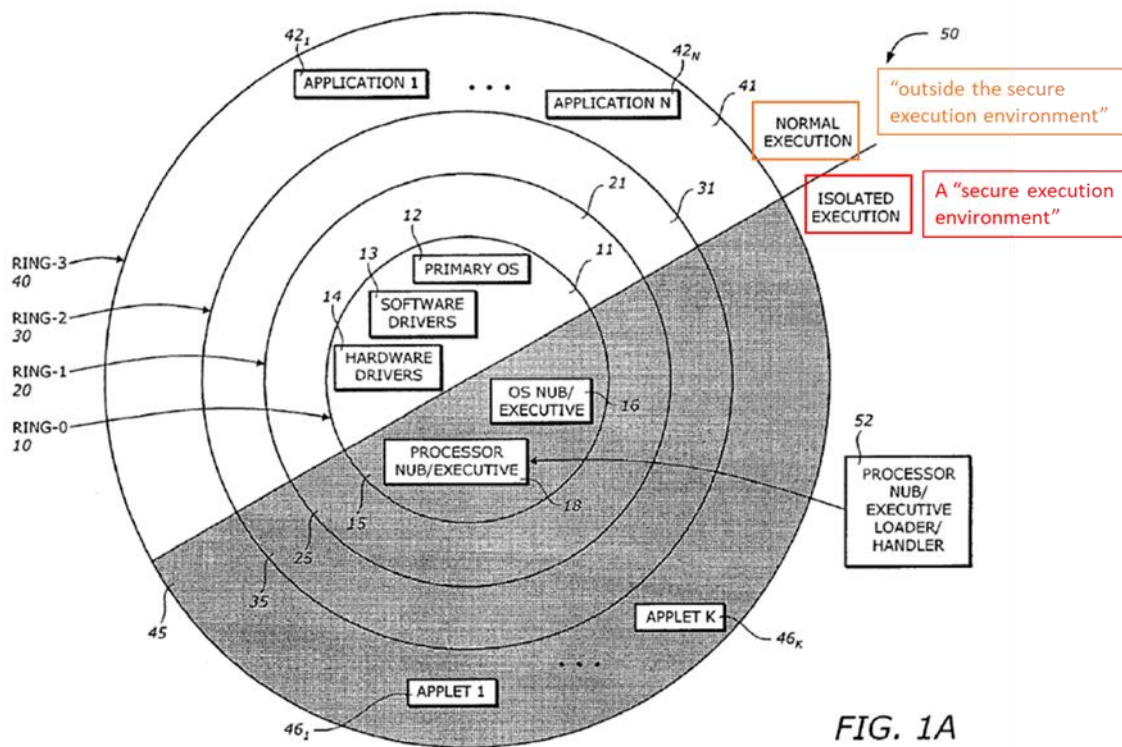
the art and the recognition and use of security/encryption already contemplated in Houghton.

B. Claim Analysis

1. Claim 4

- (a) *[4]: The message link server of claim 1, wherein the device link agent executes in a secure execution environment on at least one of the devices, and at least one of the software components executes outside of the secure execution environment on that device.*

435. Ellison discloses both an “isolated execution mode” where “access ... is restricted” and a “normal execution mode” that “operates in a non-secure environment, or a normal environment without the security features provided by the isolated execution mode.” SAMSUNG-1019, 4-65-67, 5:1, 6:1-26, 8:25-32, FIGS. 1A, 1C. Additionally, the teachings of Ellison allow for “normal” execution, for instance, for applications “***outside of the secure execution environment on that device.***” SAMSUNG-1019, 4-65-67, 5:1, 6:1-26, FIGS. 1A, 1C.

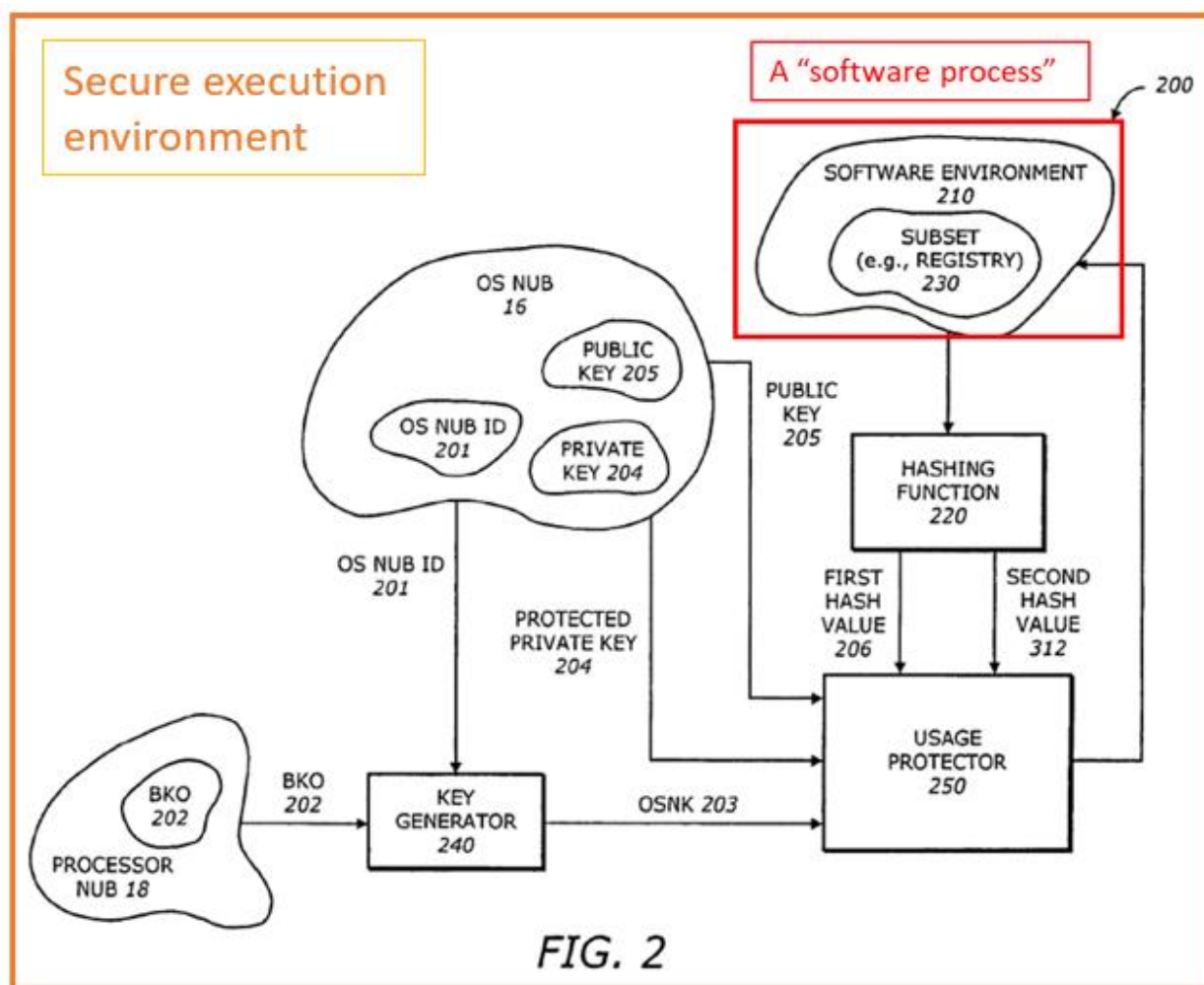


SAMSUNG-1019, FIG. 1A.

436. Ellison further discloses an “isolated area 70” which is “[a] memory area that is defined by the processor 110 when operating in [an] isolated execution mode.” SAMSUNG-1019, 6:1-26, 8:25-32, FIGS. 1A-1C, 2. Ellison refers to this technique, illustrated below in Figure 2, as a “secure platform.” SAMSUNG-1019, 6:1-26, 8:25-32, 8:66-67, 9:1-6, 9:28-62, FIGS. 1A-1C, 2.

437. Ellison further teaches that the secure platform includes a “key generator 240” that “generates a key operating system nub key (OSNK) 203” which is then supplied to “trusted agents.” *Id.*, 8:66-9:6, FIG. 2. One example of a trusted agent is a “usage protector 250” that “uses the OSNK 203 to protect the usage of [a]

subset 230.” *Id.*, 9:28-40, FIG. 2. Ellison’s usage protector 250 uses a hashing function with the subset 203’s ONSK 203 to determine if the subset 203 has been altered following changes (e.g., reads or writes to the subset 203), thereby providing “protection against unauthorized reads, and detection of intrusion, tampering or unauthorized modification” *Id.*, 9:47-62.



SAMSUNG-1019, FIG. 2.

438. In implementing the Houghton-Munson-Ellison combination, Ellison’s above-described teachings of a usage protector and secure platform, which operate

within the secure, isolated area (secure execution environment) would be incorporated into Houghton's push client executing on the mobile terminals, which in turn would provide secure access to applications residing outside the secure execution environment.

439. In doing so, the combined system would enable more secure transmission of push messages to applications on the mobile terminal, and prevent potential bad actors from sending malware to the mobile terminal in the form of such push messages or from modifying or otherwise gaining unauthorized access to the messages. *Id.*; SAMSUNG-1019, 8:25-32, 8:66-67, 9:1-6, 9:28-62, FIG. 2.

440. Moreover, given that Houghton contemplates mobile terminals with a push client that communicates with applications on the device, the well-known need to protect computing environments (including components providing essential services such as messaging, e.g., push client) from other applications/components of the environment, and Ellison's own disclosures confirming the same and providing such tiered, security architecture within the device, a POSITA would have been motivated to similar enable such improved security within the push messaging environment in Houghton, by implementing the push client (and associated components/functionality contemplated in Houghton, as described above) as part of the trusted computing environment/secure execution environment and providing the

applications that communicate with this push client outside this trusted/secure environment.

441. In this manner, Houghton-Munson-Ellison render obvious that the push client (device link agent) executes in a secure execution environment and one or more of the applications to which push messages are routed (at least one of the software components) executes outside of the secure execution environment.

xv. GROUND 8: CLAIM 8 IS RENDERED OBVIOUS BY HOUGHTON, MUNSON, AND RAKIC

A. Claim Analysis

1. Claim 8

(a) *[8]: The message link server of claim 1, further comprising a secure server to provide secure authorization signatures to the given one of the wireless end-user devices, the secure authorization signatures indicating the authorized software components that are allowed to receive data from se-cure message link messages via the message link server.*

442. For the reasons described below, Houghton-Munson in view of Rakic renders obvious that message link server (e.g., push server) comprises a secure server (e.g., a secure push message server) to provide secure authorization signatures to the given one of the wireless end-user devices (e.g., mobile terminals), the secure authorization signatures indicating the authorized software components (e.g., applications receiving application data that have previously undergone an

application registration process) that are to receive data from secure message link messages via the message link server.

443. In more detail, and as described with reference to [1b] (which I have incorporated by reference here), a POSITA would have found obvious that the mobile applications would register with the push client and/or the push server so that such applications may use the push messaging framework provided by the push client/push server.

444. As a result, when an application is registered and subsequently is the intended recipient of a push message received from a push server, the message is routed to the application based on information in the push message “specifying which mobile application 406 from a plurality of such applications” is the application identifier of the application as included in the message. SAMSUNG-1007, 21.

445. In this manner, a POSITA would have found obvious that the push client includes data indicating the authorized/registered applications (i.e., software components) that are allowed/authorized to receive data from secure message link message via the push server (i.e., message link server).

446. While a POSITA would have found obvious for the destination application identifier to be present in the received push message (as described with reference to [1b] (which I have incorporated by reference here)), Houghton and

Munson do not expressly describe validation or authentication of the message that verifies the identity of the sender of the message.

447. Without such verification, a POSITA would have understood that the device and/or the destination application may be vulnerable to attacks by unknown third party senders. Houghton and Munson also do not expressly disclose verification that the message is indeed intended for the correct application, which too can inadvertently result in confidential or sensitive data or information leaking to an incorrect device or an application.

448. In the context of push environments and push communications of the type disclosed in Houghton and Munson, Rakic provides one well-known solution to address the above-described computer security concerns. As described in §VII.C and XI.A (which I have incorporated by reference here), Rakic discloses a database device 108 that stores information related to client devices (e.g., codes obtained from a client device such as IMEI or a component in the client device (e.g., removable memory), and a secure push message server 504 uses stored information in the database device 108 to generate a key, an electronic signature, and a secure push message for a push message intended for a particular client device. SAMSUNG-1015, [0047]-[0049], [65].

449. The secure push message server generates the key based on the above-described code(s) (e.g., IMEI) that are obtained from client device 102 or a

component thereof, and subsequently, generate an electronic signature based on the key and the message. *Id.*, [0066]. The secure push message server 504 generates and sends the secure push message to client device 102, e.g., by appending the electronic signature to the message and/or other information (e.g., a timestamp). *Id.*, [0067]. At the client device, a secure push message client can validate the message and the sender by generating a key and then generating a signature using this generated key and the data block of the received message. *Id.*, [0109]-[0112]. If the signature generated at the push message client matches the signature portion of the received message, the secure push message is determined to be valid. *Id.*

450. A POSITA would have been motivated to implement the above-described techniques in Rakic within Houghton's push server to address the above-described security problems. Indeed, a POSITA would have been motivated to do so because, in view of Rakic, the above-described techniques enable (1) the client device to "authenticate the sender" and (2) "verify that an intended recipient of the secure push message is the client device, and that the client device includes a correct component." SAMSUNG-1015, [0041]. In doing so, a POSITA would have recognized the improvements to data and computer security for both the client device and the underlying applications to which the data is being transmitted. *Id.*

451. In view of the above motivations, a POSITA would have implemented Rakic's secure push message server 504 and database device 108, as well as their

above-described functionality, as part of Houghton's push server. So, configuring the Houghton-Munson's system to implement Rakic's teachings would have amounted to the use of a known technique to improve similar devices in the same way, and combining prior art elements according to known methods to yield predictable results.

452. Indeed, given that each of Houghton, Munson, and Rakic contemplate push messaging with a similar push architecture involving a push server (e.g., push server and Secure Push Message Server) that communicates with a push client/user device, a POSITA would have found it straightforward—with a reasonable expectation of success—to implement a secure server, including the signature generation functionality of Rakic's secure push messaging server, as well as its database storing the client device information, within Houghton's push server.

453. Additionally, the resulting system's elements would each perform functions they were known to perform prior to the combination—i.e., TS-23.140's system would continue to facilitate communication of messages between MMS Relay/Server and MMS User Agents, and Rakic's teachings (as implemented in combination) would facilitate provision of secure signatures (along with the messages) to the message receiving device(s).

454. In the resulting combination, Rakic's secure push message server and its associated signature generation functionality would be implemented in the push

server and would be used to generate signatures for messages intended for different mobile terminals and their respective push clients.

455. A POSITA would have understood that the signatures are authorization signatures because they are used for authentication of the sender and/or verify that the intended recipient of the message is the mobile terminal and that the mobile terminal includes a correct component. SAMSUNG-1015, [0041].

456. Additionally, as Rakic explains, the signature generating functionality “verif[ies] that an intended recipient of the secure push message is the client device, and that the client device includes a correct component.” *See id.*

457. A POSITA would have found obvious that one such component could be an application of the device. In other words, Rakic does not limit its teachings of verification of component to hardware components, and in the context of push communications that are intended to be delivered to specific software components/applications, a POSITA would have found obvious that such components would include software components. As such, in view of Rakic as applied in the context of Houghton and Munson, a POSITA would have found obvious that the above-described secure authorization signatures would indicate the authorized application(s) (i.e., software components) that are allowed/authorized to receive messages including application data via the push server.

XVI. ADDITIONAL REMARKS

I currently hold the opinions set expressed in this declaration. But my analysis may continue, and I may acquire additional information and/or attain supplemental insights that may result in added observations.

I hereby declare that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true. I further declare that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of the Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Dated: 17 November 2023

By: 

APPENDIX A

Patrick Gerard Traynor

Professor

Associate Chair for Research in CISE

John and Mary Lou Dasburgh Preeminent Chair in Engineering

Department of Computer & Information Science & Engineering (CISE)

University of Florida

E301 CSE Building, PO Box 116120

Gainesville, FL 32611 USA

traynor@cise.ufl.edu

<http://www.cise.ufl.edu/~traynor>

Table of Contents

EDUCATIONAL BACKGROUND	4
EMPLOYMENT HISTORY	4
CURRENT FIELDS OF INTEREST	4
I. TEACHING	6
A. Courses Taught	6
B. Continuing Education	6
C. Curriculum Development	6
D. Individual Student Guidance	7
E. Teaching Honors and Awards	11
II. RESEARCH AND CREATIVE SCHOLARSHIP	12
A. Thesis	12
B. Published Journal Papers (Refereed)	12
C. Published Books and Parts of Books	14
D. Edited Proceedings	14
E. Conference Presentations	14
E.1. Conference Presentations with Proceedings (Refereed)	14
E.2. Conference Presentations with Proceedings (Non-Refereed)	20
E.3. Conference Presentations without Proceedings	20
F. Other	20
F.1. Submitted Journal Papers	20
F.2. Refereed Research Reports	20
F.3. Software	20
F.4. Published Papers (Non-Refereed)	21
F.5. Books in Preparation	21
F.6. Workshops and External Courses	21
G. Research Proposals and Grants (Principal Investigator)	22
H. Research Proposals and Grants (Contributor)	24
I. Research Honors and Awards	25
III. SERVICE	27
A. Professional Activities	27
A.1. Memberships and Activities in Professional Societies	27
A.2. Conference Committee Activities	27
B. On-Campus Committees	28
B.1. University of Florida	28
B.2. Georgia Tech	29
C. Special Assignments	29
D. Ph.D. Examining Committees	29
E. External Member of M.S. Examining Committee	33
F. Consulting and Advisory Appointments	33
G. Civic Activities	33
IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION	34
A. Honors and Awards	34
B. Invited Conference Session Chairmanships	34
C. Professional Registration	34
D. Patents	34
E. Editorial and Reviewer Work for Technical Journals and Publishers	35
F. Expert Witness Services	37
V. OTHER CONTRIBUTIONS	39
A. Seminar Presentations (Invited Papers and Talks at Meetings and Symposia)	39

B. Special Activities	43
---------------------------------	----

EDUCATIONAL BACKGROUND

Degree	Year	University	Field
Ph.D.	2008	Pennsylvania State University State College, PA <i>Dissertation:</i> Characterizing the Impact of Ridigity on the Security of Cellular Telecommunications Networks <i>Advisors:</i> Thomas F. La Porta and Patrick D. McDaniel	Computer Science & Engineering
M.S.	2004	Pennsylvania State University State College, PA	Computer Science & Engineering
B.S.	2002	University of Richmond Richmond, VA <i>Minors:</i> Biology, Business Admin	Computer Science

EMPLOYMENT HISTORY

Title	Organization	Years
Professor	University of Florida	August 2018–Present
Associate Professor	University of Florida	August 2014–July 2018
Associate Professor	Georgia Institute of Technology	March 2014–August 2014
Assistant Professor	Georgia Institute of Technology	2008–March 2014
Research Assistant	Pennsylvania State University	2004–2008
Teaching Assistant	Pennsylvania State University	2004

CURRENT FIELDS OF INTEREST

My research focuses on the security of cellular/telephony networks and mobile systems. The security of these systems generally relies on their closed nature and trust in the honest behavior of users. However, with the recent disintegration of these assumptions and with over than six billion subscribers around the world, cellular and mobile systems represent the next great expansion in global critical infrastructure and, because of their unique characteristics, require new and different approaches to security.

Recognizing this, my research focuses on three specific themes: (1) developing efficient techniques to allow telephony providers and customers to authenticate the origin of incoming calls; (2) measuring and improving the security of emerging mobile financial systems and (3) efficient and strong privacy-preserving techniques for mobile devices. Additionally, I have significant expertise in fraud detection, particularly for payment systems.

I have a strong interest in solutions that can be deployed in both the short and long terms, and am actively engaging both industry and government in this capacity. My research, if successful, will help to not only improve the general security of networked devices, but also to maintain the historical reliability of telephony networks as they become the dominant digital access technology.

I. TEACHING

A. Courses Taught

Semester/Year	Course Number & Title	Number of Students	Comments
Fall 2022	CNT 5410 Computer and Network Security	75	New Topics
Fall 2021	CNT 5410 Computer and Network Security	45	New Topics
Fall 2019	CNT 5410 Computer and Network Security	28	New Topics
Fall 2018	CIS 6930 Cellular and Mobile Network Security	16	New Course
Fall 2017	CNT 5410 Computer and Network Security	27	New Topics
Fall 2016	CNT 5410 Computer and Network Security	60	New Topics
Spring 2016	CNT 5410 Computer and Network Security	13	New Topics
Spring 2015	CNT 5410 Computer and Network Security	12	New Topics
Fall 2014	CNT 5410 Computer and Network Security	30	New Course
Spring 2014	CS 6262 Network Security	55	New Projects
Fall 2013	CS 3251 Computer Networks I	73	Expanded Syllabus
Spring 2013	CS 6262 Network Security	65	All New Projects
	CS 8001 Information Security Seminar	20	New Speakers
Fall 2012	CS 8803 Cellular & Mobile Network Security	17	New Topics
	CS 8001 Information Security Seminar	20	New Speakers
Spring 2011	CS 8001 Information Security Seminar	20	New Speakers
Fall 2011	CS 6262 Network Security	27	Expanded Syllabus
	CS 8001 Information Security Seminar	35	New Speakers
Spring 2011	CS 3251 Computer Networks I	61	Expanded Syllabus
	CS 8001 Information Security Seminar	20	New Speakers
Fall 2010	CS 8803/4803 Cellular & Mobile Network Security	16	New Course
	CS 8001 Information Security Seminar	31	New Speakers
Fall 2009	CS 6262 Network Security	55	Expanded Syllabus
Spring 2009	CS 3251 Computer Networks I	45	Expanded Syllabus
Fall 2008	CS 8003 Destructive Research	10	New Course

Guest lecturer for CS 4235 (Introduction to Information Security) and CS 8803 (e-Democracy) in Fall 2008.

Advised ECE 4811/CS 4802 (Vertically Integrated Project) with Ed Coyle

B. Continuing Education

None.

C. Curriculum Development

CS 8803 Cellular and Mobile Network Security: *Fall 2010.* Developed an entirely new course around security issues facing cellular and mobile networks. Students learned about wireless basics, spectrum issues, core network architectures (GSM, ISDN, IMS, SIP), air interfaces (GSM, 3G), mobility management, authentication, mobile phone operating systems (Android, iPhone), Android security, congestion and denial of service, privacy and eavesdropping. Students also complete a research project and aim towards publishing this work at a major venue. My aim is for this class to become part of the regular offering of security courses and receive a non-8803 number. Semester projects were also judged and encouraged using a “venture capital” model, in which students had to pretend as if they were pitching their ideas for a start-up

company to potential investors.

CS 6262 Network Security: *Fall 2009.* Totally rewrote the syllabus and slide material, giving the class its first major overhaul in a number of years. While many old themes remain, new lecture blocks including Web Security, Cellular Security and Social Engineering were developed from scratch. This new course material was made available to all other faculty members teaching this class, who have since used my slides and syllabus.

CS 3251 Computer Networks I: *Spring 2009.* Modified undergraduate networking course to include a persistent focus on security at all layers of the protocol stack. I have also created new lectures focusing on the physical layer and cellular networks and new exams to include all of the abovementioned changes.

CS 8803 Destructive Research: *Fall 2008.* Developed course based around understanding how so-called secure systems have been defeated by attackers. With such knowledge, students would have the context to develop the next generation of more secure systems. I delivered more than 1/3 of the lectures in this seminar course and paid special focus on vulnerabilities in cellular networks, analog telecommunications and electronic voting. Students were also instructed on techniques for performing research, writing technical papers and making conference and lecture-style presentations. I have offered these slides to future 7001 classes to help impact a wider audience.

D. Individual Student Guidance

1. Research Scientists Supervised

None.

2. Ph.D. Students Graduated

Chaitrali Amrutkar Georgia Institute of Technology

Fall 2009–Fall 2013

Her research discovered vulnerabilities in mobile web browsers and developed techniques to detect malicious mobile web pages. Joined Oracle in Spring 2014.

Jasmine Bowers University of Florida

Fall 2015–Summer 2020

Her research focuses on mobile applications, and the development of tools for building secure systems. Now: Research Scientist, MITRE

Henry “Hank” Carter Georgia Institute of Technology

Fall 2010–Spring 2016

Developing techniques for secure function evaluation for privacy-preserving applications on constrained mobile devices. Now: Assistant Professor, Villanova University

Italo Dacosta Georgia Institute of Technology

Fall 2008–Summer 2012

Co-advised with Mustaque Ahamad. Research on scaling performance of SIP network components. Graduated Summer 2012, currently research scientist at EPFL.

David Dewey Georgia Institute of Technology

Fall 2011–Summer 2015

Investigated compiler techniques to remove software vulnerabilities from code. Now CTO of MailChimp.

- Brad Reaves** University of Florida
Fall 2014–Spring 2017
Develop strong authentication techniques for cellular networks. Now: Assistant Professor at North Carolina State University.
- Nolen Scaife** University of Florida
Fall 2014–Spring 2019
Developed techniques to detect credit card skimming. First: Assistant Professor at the University of Colorado Boulder. Now: Director, Global Cyber Intelligence at Walmart
- Imani Sherman** University of Florida
Fall 2018–Summer 2021
Developing usable interfaces against robocalls. Co-advised with Juan Gilbert. Now: Assistant Professor at the University of California, San Diego
- Luis Vargas** University of Florida
Fall 2016–Summer 2021
Developing techniques for network-based detection and mitigation of malware in a healthcare environment. Now: Data Scientist at the Alethia Group
- Hadi Abdullah** University of Florida
Fall 2016–Summer 2022
Evaluating the security of ML-driven voice interfaces. Now: Research Scientist at Visa Research
- Christian Peeters** University of Florida
Fall 2016–Summer 2022
Develop techniques to detect and defend against call and message interception attacks in cellular networks. Now: Research Scientist at Harbor Labs

2. Ph.D. Students Supervised

- Logan Blue** University of Florida
Fall 2016–Present
Investigating facial feature reconstruction from voice recordings.
- Nathaniel Bennett** University of Florida
Fall 2022–Present
Finding vulnerabilities in cellular core networks via fuzzing.
- Cassidy Gibson** University of Florida
Fall 2019–Present
Investigating weaknesses in web software.
- Ryon Kennedy** University of Florida
Fall 2020–Present
Finding vulnerabilities in cellular core networks via fuzzing.
- Seth Layton** University of Florida
Fall 2020–Present
Detecting deepfakes in audio samples.
- Allison Lu** University of Florida
Fall 2022–Present
Measuring repeatability in computer security.

Daniel Olszewski University of Florida
Fall 2019–Present
Removing unwanted/insecure features from software.

Tyler Tucker University of Florida
Fall 2021–Present
Evaluating the security of Bluetooth/cellular radios.

Kevin Warren University of Florida
Fall 2019–Present
Detecting deepfake audio through linguistic information.

3. Ph.D. Students - Other

Saurabh Chakradeo Georgia Institute of Technology
Fall 2010–Spring 2013
Research exploring malicious mobile applications. Left to join Facebook.

Brendan Dolan-Gavitt Georgia Institute of Technology
Spring 2009
Research project on using kernel type graphs to detect dummy structures.

Eric (Yu) Liu Georgia Institute of Technology
Fall 2008
Research on the spread of malware through cellular infrastructure.

Chaz Lever Georgia Institute of Technology
Fall 2011–Spring 2014
Developing techniques to measure the spread of malware in cellular networks. Left Georgia Tech to create a startup.

Frank Park Georgia Institute of Technology
Fall 2008–Spring 2010
Research on multi-factor authentication using cellular phones. Left program after failing comprehensive exam to join startup.

Ferdinand Schober Georgia Institute of Technology
Fall 2009–Summer 2010
Developed mechanisms for smart networks and smart mobile devices to fight infection and provide remote remediation. Returned to Microsoft.

4. M.S. Students Supervised

Chaitrali Amrutkar Georgia Institute of Technology
Fall 2008–Spring 2009
Research on improving performance of security critical functions in IMS cellular core. Completed her Ph.D with me at GT.

Logan Blue University of Florida
Fall 2015–Spring 2016
Investigated problems of cellular and network security.

David Dewey Georgia Institute of Technology
Fall 2009–Spring 2010
Research on security issues caused by transitive trust assumptions in the Windows COM infrastructure. Completed his Ph.D. with me at GT.

Christopher Grayson Georgia Institute of Technology

Fall 2012–Fall 2013

Developed continuous authentication mechanisms using the multitude of sensors available on a mobile phone. Now at Bishop Fox Consulting (industry).

Young Seuk Kim Georgia Institute of Technology

Fall 2012–Fall 2013

Performed research that compared the security vulnerabilities found in the traditional and mobile web.

Daniel Komaromy Georgia Institute of Technology

Fall 2008–Summer 2009

Research on building a real-time streaming audio system using attribute-based crypto for broadcast encryption.

Nigel Lawrence Georgia Institute of Technology

Fall 2011–Spring 2012

Discovered hijacking attacks in SNMPv3, a widely used and thought to be secure network management protocol. Now at Solute (industry).

Philip Marquardt Georgia Institute of Technology

Fall 2009–Present

Research on developing an iPhone application to prevent individuals from being profiled by Shopper Loyalty Programs. First with MIT Lincoln Labs, now Raytheon

Rishikesh Naik Georgia Institute of Technology

Fall 2008–Spring 2010

Research on converting expensive cryptographic primitives (e.g., Secure Function Evaluation) into efficient applications for mobile phones. Now with Cisco Systems.

Ashish Nautiyal CISE

Fall 2015–Spring 2016

Research on connecting telephone calls to the larger authentication infrastructure.

Nilesh Nipane Georgia Institute of Technology

Fall 2008–Spring 2010

Research on creating provably anonymous networks on a base of secure function evaluation. Now with VMWare.

Walter “Nolen” Sciafe Georgia Institute of Technology

Spring 2012–Spring 2014

Developed the OnionDNS architecture, which prevents domain delisting attacks by leveraging a Tor hidden service. Joined Ph.D. program at UF.

Tyler Tucker University of Florida

Fall 2018–Spring 2021

Evaluating the security of Bluetooth radios.

5. M.S. Special Problems Students

Siddhant Deshmukh University of Florida

Fall 2016–Present

Developed tools for analysis of mobile digital financial services.

Chinmay Gangakhedkar Georgia Institute of Technology

Spring 2009

Research on multi-factor authentication using mobile phones.

Christopher Grayson Georgia Institute of Technology
Spring 2013

Research on continuous authentication using mobile phones.

Aarushi Karnany University of Florida
Fall 2016–Present

Developed tools for analysis of mobile digital financial services.

Rohit Matthews Georgia Institute of Technology
Spring 2011

Developed mobile phone-based tools for measuring performance and reachability throughout the Internet.

Ashwin Narasimhan Georgia Institute of Technology
Spring 2009

Research on developing efficient security mechanisms for the IMS cellular core.

Aamir Poonawalla Georgia Institute of Technology
Spring 2010

Helped develop a call provenance infrastructure, which included both networking and machine learning components.

Erin Reddick Georgia Institute of Technology
Fall 2008–Fall 2009

Research on IPTV security with GTRI.

Lalanthika Vasudevan Georgia Institute of Technology
Spring 2009

Research on developing efficient security mechanisms for the IMS cellular core.

6. Undergraduate Special Problems Students

Ethan Shernan Georgia Institute of Technology
Spring 2014

Developed an infrastructure for detecting billing bypass fraud attacks.

Young Seuk Kim Georgia Institute of Technology
Fall 2011–Spring 2012

Developed a mobile phone application for taking measurements of cellular networks.

Dane Van Dyck Georgia Institute of Technology
Summer 2009

Research on virtualization support for mobile phones.

E. Teaching Honors and Awards

1. Georgia Institute of Technology, CETL “Thanks For Being A Great Teacher” Award, Fall 2013.
2. Georgia Institute of Technology, CETL “Thanks For Being A Great Teacher” Award, Fall 2012.
3. United State Army Signal Corps, “Helmet” Award, 2010.
4. Georgia Institute of Technology, CETL “Thanks For Being A Great Teacher” Award, Spring 2009.
5. Pennsylvania State University CSE Graduate Student Teaching Award, 2005

II. RESEARCH AND CREATIVE SCHOLARSHIP

A. Thesis

1. Patrick Gerard Traynor. *Characterizing the Impact of Rigidity on the Security of Cellular Telecommunications Networks*. PhD thesis, The Pennsylvania State University, May 2008.

B. Published Journal Papers (Refereed)

1. Cassidy Gibson, Vanessa Frost, Katie Platt, Washington Garcia, Luis Vargas, Sara Rampazzi, Vincent Bindschaedler, Patrick Traynor, and Kevin Butler. Analyzing the Monetization Ecosystem of Stalkerware. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2022.
2. Bradley Reaves, Luis Vargas, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Characterizing the Security of the SMS Ecosystem with Public Gateways. *ACM Transactions on Privacy and Security (TOPS)*, 22(1), 2018.
3. Patrick Traynor, Kevin Butler, Jasmine Bowers, and Bradley Reaves. FinTechSec: Addressing the Security Challenges of Digital Financial Services. *IEEE S&P Magazine*, 15(5):85–89, 2017.
4. Nolen Scaife, Henry Carter, Rachel Jones, Lyrissa Lidsky, and Patrick Traynor. OnionDNS: A Seizure-Resistant Top-level Domain. *International Journal of Information Security (IJIS)*, 2017.
5. Bradley Reaves, Jasmine Bowers, Nolen Scaife, Adam Bates, Arnav Bharatiya, Patrick Traynor, and Kevin Butler. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. *ACM Transactions on Privacy and Security (TOPS)*, 2017.
6. Henry Carter and Patrick Traynor. OPFE: Outsourcing Computation for Private Function Evaluation. *International Journal of Information and Computer Security (IJICS)*, 2017.
7. Stephan Heuser, Bradley Reaves, Praveen Kumar Pendyala, Henry Carter, Alexandra Dmitrienko, William Enck, Negar Kiyavash, Ahmad-Reza Sadeghi, and Patrick Traynor. Phonion: Practical Protection of Metadata in Telephony Networks. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2017.
8. Bradley Reaves, Jasmine Bowers, Sigmond A. Gorski III, Olabode Anise, Rahul Bobhate, Raymond Cho, Hiranava Das, Sharique Hussain, Hamza Karachiwala, Nolen Scaife, Byron Wright, Kevin Butler, William Enck, and Patrick Traynor. *droid: Assessment and Evaluation of Android Application Analysis Tools. *ACM Computing Surveys (CSUR)*, 2016.
9. Chaitrali Amrutkar, Young Seuk Kim, and Patrick Traynor. Detecting Mobile Malicious Webpages in Real Time. *IEEE Transactions on Mobile Computing (TMC)*, To Appear 2016.
10. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Outsourcing Secure Two-Party Computation as a Black Box. *Journal of Security and Communication Networks (SCN)*, To Appear 2016.
11. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Secure Outsourced Garbled Circuit Evaluation for Mobile Devices. *Journal of Computer Security (JCS)*, 24(2):137–180, 2016.
12. Adam Bates, Kevin Butler, Micah Sherr, Clay Shields, Patrick Traynor, and Dan Wallach. Accountable Wiretapping -or- I Know They Can Hear You Now. *Journal of Computer Security (JCS)*, 23(2):167–195, 2015.
13. Henry Carter, Chaitrali Amrutkar, Italo Dacosta, and Patrick Traynor. For Your Phone Only: Custom Protocols for Efficient Secure Function Evaluation on Mobile Devices. *Journal of Security and Communication Networks (SCN)*, 7(7):1165–1176, 2014.

14. Chaitrali Amrutkar, Patrick Traynor, and Paul van Oorschot. An Empirical Evaluation of Security Indicators in Mobile Web Browsers. *IEEE Transactions on Mobile Computing (TMC)*, 14(5), 2015.
15. Andrew Harris, Seymour Goodman, and Patrick Traynor. Privacy and Security Concerns Associated with Mobile Money Applications in Africa. *Washington Journal of Law, Technology & Arts*, 8(3), 2013.
16. Italo Dacosta, Saurabh Chakradeo, Mustaque Ahamad, and Patrick Traynor. One-Time Cookies: Preventing Session Hijacking Attacks with Stateless Authentication Tokens. *ACM Transactions on Internet Technology (TOIT)*, 12(1), 2012.
17. Cong Shi, Xiapu Luo, Patrick Traynor, Mostafa Ammar, and Ellen Zegura. ARDEN: Anonymous netwoRking in Delay tolErant Networks. *Journal of Ad Hoc Networks*, 10(6):918–930, 2012.
18. Patrick Traynor. Characterizing the Security Implications of Third-Party EAS Over Cellular Text Messaging Services. *IEEE Transactions on Mobile Computing (TMC)*, 11(6):983–994, 2012.
19. Italo Dacosta, Vijay Balasubramaniyan, Mustaque Ahamad, and Patrick Traynor. Improving Authentication Performance of Distributed SIP Proxies. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 22(11):1804–1812, 2011.
20. Patrick Traynor, Chaitrali Amrutkar, Vikhyath Rao, Trent Jaeger, Patrick McDaniel, and Thomas La Porta. From Mobile Phones to Responsible Devices. *Journal of Security and Communication Networks (SCN)*, 4(6):719 – 726, 2011.
21. Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure Attribute-Based Systems. *Journal of Computer Security (JCS)*, 18(5):799–837, 2010.
22. Patrick Traynor, Kevin Butler, William Enck, Kevin Borders, and Patrick McDaniel. malnets: Large-Scale Malicious Networks via Compromised Wireless Access Points. *Journal of Security and Communication Networks (SCN)*, 2(3):102–113, 2010.
23. Patrick Traynor. Securing Cellular Infrastructure: Challenges and Opportunities. *IEEE Security & Privacy Magazine*, 7(4), 2009.
24. Kevin Butler, Sunam Ryu, Patrick Traynor, and Patrick McDaniel. Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 20(12):1803–1815, 2009.
25. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Mitigating Attacks On Open Functionality in SMS-Capable Cellular Networks. *IEEE/ACM Transactions on Networking (TON)*, 17(1), 2009.
26. Patrick Traynor, Michael Chien, Scott Weaver, Boniface Hicks, and Patrick McDaniel. Non-Invasive Methods for Host Certification. *ACM Transactions on Information and System Security (TISSEC)*, 2008.
27. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. *Journal of Computer Security (JCS)*, 16(6):713–742, 2008.
28. Patrick Traynor, Raju Kumar, Heesook Choi, Sencun Zhu, Guohong Cao, and Thomas La Porta. Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. *IEEE Transactions on Mobile Computing (TMC)*, 6(6), 2007.

C. Published Books and Parts of Books

1. Andrew Harris, Frank S. Park, Seymour Goodman, and Patrick Traynor. *Emerging Privacy and Security Concerns for Digital Wallet Deployment*. Privacy in America: Interdisciplinary Perspectives. Scarecrow Press, July 2011.
2. Kevin Butler, William Enck, Patrick Traynor, Jennifer Plasterr, and Patrick McDaniel. *Privacy Preserving Web-Based Email*. Algorithms, Architectures and Information Systems Security, Statistical Science and Interdisciplinary Research. World Scientific Computing, November 2008.
3. Patrick Traynor, Patrick McDaniel, and Thomas La Porta. *Security for Telecommunications Networks*. Number 978-0-387-72441-6 in Advances in Information Security Series. Springer, August 2008.

D. Edited Proceedings

None.

E. Conference Presentations

E.1. Conference Presentations with Proceedings (Refereed)

1. Christian Peeters, Tyler Tucker, Anushri Jain, Kevin Butler, and Patrick Traynor. LeopardSeal: Detecting Call Interception via Audio Rogue Base Stations. In *Proceedings of the ACM International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2023.
2. Tyler Tucker, Hunter Searle, Kevin Butler, and Patrick Traynor. Blue's Clues: Practical Discovery of Non-Discoverable Bluetooth Devices. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2023.
3. Hadi Abdullah, Aditya Karlekar, Saurabh Prasad, Muhammad Sajidur Rahman, Logan Blue, Luke Bauer, Vincent Bindschaedler, and Patrick Traynor. Attacks as Defenses: Designing Robust Audio CAPTCHAs Using Attacks on Automatic Speech Recognition Systems. In *Symposium on Network and Distributed System Security (NDSS)*, 2023.
4. Daniel Olszewski, Sandeep Sathyanarayana, Weidong Zhu, Kevin Butler, and Patrick Traynor. HallMonitor: A Framework for Identifying Network Policy Violations in Software. In *IEEE Conference on Communications and Network Security (CNS)*, 2022.
5. Hadi Abdullah, Aditya Karlekar, Vincent Bindschaedler, and Patrick Traynor. Demystifying Limited Adversarial Transferability in Automatic Speech Recognition Systems. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. (Acceptance rate: 32%).
6. Logan Blue, Kevin Warren, Hadi Abdullah, Cassidy Gibson, Luis Vargas, Jessica O'Dell, Kevin Butler, and Patrick Traynor. Who Are You (I Really Wanna Know)? Detecting Audio DeepFakes Through Vocal Tract Reconstruction. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2022. (Acceptance rate: 17.2%).
7. Grant Hernandez, Marius Muench, Dominik Maier, Alyssa Milburn, Shinjo Park, Tobias Scharnowski, Tyler Tucker, Patrick Traynor, and Kevin R. B. Butler. FirmWire: Transparent Dynamic Analysis for Cellular Baseband Firmware. In *Symposium on Network and Distributed System Security (NDSS)*, 2022. (Acceptance rate: 16.2%).
8. Christian Peeters, Christopher Patton, Imani N. Sherman, Daniel Olszewski, Thomas Shrimpton, and Patrick Traynor. SMS OTP Security (SOS): Hardening SMS-Based Two Factor Authentication. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2022. (Acceptance rate: 18.2%).

9. Hadi Abdullah, Muhammad Sajidur Rahman, Christian Peeters, Cassidy Gibson, Washington Garcia, Vincent Bindschaedler, Thomas Shrimpton, and Patrick Traynor. Beyond L_p Clipping: Equalization based Psychoacoustic Attacks against ASRs. In *The Asian Conference on Machine Learning (ACML)*, 2021.
10. Imani Sherman and Daniel Delgado and Juan Gilbert and Jaime Ruiz and Patrick Traynor. Characterizing User Comprehension in the STIR/SHAKEN Anti-Robocall Standard. In *Proceedings of the Annual Research Conference on Communications Information and Internet Policy (TPRC 49)*, 2021.
11. Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2021. (Acceptance rate: 12.1%).
12. Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Logan Blue, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. Hear “No Evil”, See “Kenansville”: Efficient and Transferable Black-Box Attacks on Speech Recognition and Voice Identification Systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2021. (Acceptance rate: 12.1%).
13. Imani Sherman, Jasmine Bowers, Liz-Laure Laborde, Juan E. Gilbert, Jaime Ruiz, and Patrick Traynor. Truly Visual Caller ID? An Analysis of Anti-Robocall Applications and their Accessibility to Visually Impaired Users. In *IEEE International Symposium on Technology and Society (IEEE ISTAS)*, 2020.
14. Imani Sherman, Jasmine Bowers, Keith McNamara, Juan Gilbert, Jaime Ruiz, and Patrick Traynor. Are You Going to Answer That? Measuring User Responses to Anti-Robocall Application Indicators. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2020. (Acceptance rate: 17.4%).
15. Joseph Choi, Dave Tian, Grant Hernandez, Christopher Patton, Benjamin Mood, Thomas Shrimpton, Patrick Traynor, and Kevin Butler. A Hybrid Approach to Secure Function Evaluation Using SGX. In *Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACCS’19)*, 2019. (Acceptance Rate: 17.0% for full papers).
16. Vanessa Frost, Dave Tian, Christie Ruales, Patrick Traynor, and Kevin Butler. Examining DES-based Cipher Suite Support within the TLS Ecosystem. In *Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACCS’19)*, 2019. (Acceptance Rate: 22.0% for all papers).
17. Dave Tian, Joseph Choi, Grant Hernandez, Patrick Traynor, and Kevin Butler. A Practical Intel SGX Setting for Linux Containers in the Cloud. In *Proceedings of the ACM Conference on Data and Application Security and Privacy (CODASPY’19)*, 2019. (Acceptance rate: 23.5%).
18. Nolen Scaife, Jasmine Bowers, Christian Peeters, Grant Hernandez, Imani Sherman, Lisa Anthony, and Patrick Traynor. Kiss from a Rogue: Evaluating Detectability of Pay-at-the-Pump Card Skimmers. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2019. (Acceptance rate: 12.0%).
19. Jasmine Bowers, Imani Sherman, Kevin Butler, and Patrick Traynor. Characterizing Security and Privacy Practices in Emerging Digital Credit Applications. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019. (Acceptance rate: 25.6%).
20. Hadi Abdullah, Washington Garcia, Christian Peeters, P. Traynor, K. Butler, and J. Wilson. Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2019. (Acceptance Rate: 17.1%).

21. Lius Vargas, Logan Blue, Vanessa Frost, Christopher Patton, N. Scaife, K. Butler, and P. Traynor. Digital Healthcare-Associated Infection Analysis of a Major Multi-Campus Hospital System. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2019. (Acceptance Rate: 17.1%).
22. Dominik Wermke, Nicolas Huaman, Yasemin Acar, Bradley Reaves, Patrick Traynor, and Sascha Fahl. A Large Scale Investigation of Obfuscation Use in Google Play. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2018. Acceptance Rate: 20.1%.
23. Nolen Scaife, Christian Peeters, and Patrick Traynor. Fear the Reaper: Characterization and Fast Detection of Card Skimmers. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2018. Acceptance Rate: 19.0%.
24. Dave (Jing) Tian, Grant Hernandez, Joseph Choi, Vanessa Frost, Christie Raules, Kevin Butler, Patrick Traynor, Hayawardh Vijayakumar, Lee Harrison, Amir Rahmati, and Mike Grace. AT-tention Spanned: Comprehensive Vulnerability Analysis of AT Commands Within the Android Ecosystem. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2018. Acceptance Rate: 19.0%.
25. Logan Blue, Hadi Abdullah, Luis Vargas, and Patrick Traynor. 2MA: Verifying Voice Commands via Two Microphone Authentication. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2018. (Acceptance Rate: 20.0%).
26. Nolen Scaife, Christian Peeters, Camilo Velez, Hanqing Zhao, Patrick Traynor, and David Arnold. The Cards Aren't Alright: Detecting Counterfeit Gift Cards Using Encoding Jitter. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2018. (Acceptance Rate: 10.4%).
27. Christian Peeters, Hadi Abdullah, Nolen Scaife, Jasmine Bowers, Patrick Traynor, Bradley Reaves, and Kevin Butler. Sonar: Detecting SS7 Redirection Attacks Via Call Audio-Based Distance Bounding. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2018. (Acceptance Rate: 10.4%).
28. Tyler Ward, Joseph Choi, Kevin Butler, John M. Shea, Patrick Traynor, and Tan Wong. Privacy Preserving Localization Using a Distributed Particle Filtering Protocol. In *IEEE MILCOM*, 2017. (Acceptance Rate: 56%).
29. Bradley Reaves and Logan Blue and Hadi Abdullah and Luis Vargas and Patrick Traynor and Thomas Shrimpton. AuthentiCall: Efficient Identity and Content Authentication for Phone Calls. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2017. (Acceptance Rate: 16.3%).
30. Jasmine Bowers and Bradley Reaves and Imani N. Sherman and Patrick Traynor and Kevin Butler. Regulators, Mount Up! Analysis of Privacy Policies for Mobile Money Applications. In *Proceedings of the USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2017. (Acceptance Rate: 26.5%).
31. Bradley Reaves, Logan Blue, and Patrick Traynor. AuthLoop: End-to-End Cryptographic Authentication for Telephony over Voice Channels. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2016. (Acceptance Rate: 15.5%).
32. Dave Tian, Nolen Scaife, Adam Bates, Kevin Butler, and Patrick Traynor. Making USB Great Again with USBFILTER. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2016. (Acceptance Rate: 15.5%).
33. Bradley Reaves, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Detecting SMS Spam in the Age of Legitimate Bulk Messaging. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2016. (Acceptance Rate: 35.0%).

34. Nolen Scaife, Henry Carter, Patrick Traynor, and Kevin Butler. CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2016. (Acceptance Rate: 17.6%).
35. Bradley Reaves, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Sending out an SMS: Characterizing the Security of the SMS Ecosystem with Public Gateways. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2016. (Acceptance Rate: 13.0%).
36. Benjamin Mood, Debayan Gupta, Henry Carter, Kevin Butler, and Patrick Traynor. Frigate: A Validated, Extensible, and Efficient Compiler and Interpreter for Secure Computation. In *Proceedings of the IEEE European Symposium on Security and Privacy*, 2016. (Acceptance Rate: 17.3%).
37. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Outsourcing Secure Two-Party Computation as a Black Box. In *Proceedings of the International Conference on Cryptology and Network Security*, 2015. (Acceptance Rate: 52.9%).
38. Nolen Scaife, Henry Carter, and Patrick Traynor. OnionDNS: A Seizure-Resistant Top-level Domain. In *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*, 2015. (Acceptance Rate: 28.1%).
39. Bradley Reaves, Nolen Scaife, Adam Bates, Patrick Traynor, and Kevin Butler. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2015. (Acceptance Rate: 15.7%).
40. Bradley Reaves, Ethan Shernan, Adam Bates, Henry Carter, and Patrick Traynor. Boxed Out: Blocking Cellular Interconnect Bypass Fraud at the Network Edge. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2015. (Acceptance Rate: 15.7%).
41. David Dewey, Bradley Reaves, and Patrick Traynor. Uncovering Use-After-Free Conditions In Compiled Code. In *Proceedings of the International Conference on Availability, Reliability and Security (ARES)*, 2015. (Acceptance Rate: 22%).
42. Ethan Shernan, Henry Carter, Dave Tian, Patrick Traynor, and Kevin Butler. More Guidelines Than Rules: CSRF Vulnerabilities from Noncompliant OAuth 2.0 Implementations. In *Proceedings of the International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA)*, 2015. (Acceptance Rate: 22.7%).
43. Henry Carter, Charles Lever, and Patrick Traynor. Whitewash: Outsourcing Garbled Circuit Generation for Mobile Devices. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2014. (Acceptance Rate: 19.9%).
44. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Secure Outsourced Garbled Circuit Evaluation for Mobile Devices. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2013. (Acceptance Rate: 16.2%).
45. Chaitrali Amrutkar, Matti Hiltunen, Shobha Venkataraman, Kaustubh Joshi, Patrick Traynor, Trevor Jim, and Oliver Spatscheck. Why is My Smartphone Slow? On The Fly Diagnosis of Poor Performance on the Mobile Internet. In *Proceedings of The 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2013. (Acceptance Rate: 19.6%).
46. Saurabh Chakradeo, Brad Reaves, Patrick Traynor, and William Enck. MAST: Triage for Market-scale Mobile Malware Analysis. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2013. (Acceptance Rate: 15.0%)(Best Paper).
47. Charles Lever, Manos Antonakakis, Brad Reaves, Patrick Traynor, and Wenke Lee. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2013. (Acceptance rate: 18.8%).

48. Chaitrali Amrutkar, Kapil Singh, Arunabh Verma, and Patrick Traynor. VulnerableMe: Measuring Systemic Weaknesses in Mobile Browser Security. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, 2012. (Acceptance rate: 25%) (Best Paper - SAIC Student Paper Competition (GT)) (Finalist - CSAW AT&T Applied Security Research Best Paper Competition 2012).
49. Chaitrali Amrutkar, Patrick Traynor, and Paul van Oorschot. A Measurement Study of SSL Indicators on Mobile Browsers: Extended Life, or End of the Road? In *Proceedings of the Information Security Conference (ISC)*, 2012. (Acceptance rate: 32%) (Best Student Paper).
50. Italo Dacosta, Mustaque Ahamad, and Patrick Traynor. Trust No One Else: Detecting MITM Attacks Against SSL/TLS Without Third-Parties. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, 2012. (Acceptance Rate: 20.2%).
51. Adam Bates, Kevin Butler, Micah Sherr, Clay Shields, Patrick Traynor, and Dan Wallach. Accountable Wiretapping -or- I Know They Can Hear You Now. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2012. (Acceptance Rate: 17.8%).
52. Yacin Nadji, Jon Giffin, and Patrick Traynor. Automated Remote Repair for Mobile Malware. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2011. (Acceptance Rate: 18.5%).
53. Nilesh Nipane, Italo Dacosta, and Patrick Traynor. "Mix-In-Place" Anonymous Networking Using Secure Function Evaluation. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2011. (Acceptance Rate: 18.5%).
54. Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. (sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2011. (Acceptance Rate: 13.9%).
55. Philip Marquardt, David Dagon, and Patrick Traynor. Impeding Individual User Profiling in Shopper Loyalty Programs. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*, 2011. (Acceptance Rate: 35.1%).
56. David Dewey and Patrick Traynor. No Loitering: Exploiting Lingering Vulnerabilities in Default COM Objects. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2011. (Acceptance Rate: 20.1%).
57. Vijay Balasubramaniyan, Aamir Poonawalla, Mustaque Ahamad, Michael Hunter, and Patrick Traynor. PinDrOp: Using Single-Ended Audio Features to Determine Call Provenance. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2010. (Acceptance Rate: 17.2%).
58. Patrick Traynor, Joshua Schiffman, Thomas La Porta, Patrick McDaniel, Abhrajit Ghosh, and Farooq Anjum. Constructing Secure Localization Systems with Adjustable Granularity. In *IEEE Global Communications Conference (GLOBECOM)*, 2010. (Acceptance Rate: 35.6%).
59. Patrick Traynor. Characterizing the Security Implications of Third-Party EAS Over Cellular Text Messaging Services. In *Proceedings of the Second IEEE International Conference on Security and Privacy in Communication Networks (SecureComm)*, 2010. (Acceptance Rate: 25.0%).
60. Kapil Singh, Samrit Sangal, Nehil Jain, Patrick Traynor, and Wenke Lee. Evaluating Bluetooth as a Medium for Botnet Command and Control. In *Proceedings of the International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA)*, 2010. (Acceptance Rate: 30.7%).

61. Italo Dacosta and Patrick Traynor. Proxychain: Developing a Robust and Efficient Authentication Infrastructure for Carrier-Scale VoIP Networks. In *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2010. (Acceptance Rate: 17.0%).
62. Frank S. Park, Chinmay Gangakhedkar, and Patrick Traynor. Leveraging Cellular Infrastructure to Improve Fraud Prevention. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2009. (Acceptance Rate: 19.0%).
63. Patrick Traynor, Michael Lin, Machigar Ongtang, Vikyath Rao, Trent Jaeger, Thomas La Porta, and Patrick McDaniel. On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2009. (Acceptance Rate: 18.4%).
64. Brendan Dolan-Gavitt, Abhinav Srivastava, Patrick Traynor, and Jonathon Giffin. Robust Signatures for Kernel Data Structures. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2009. (Acceptance Rate: 18.4%).
65. Italo Dacosta, Vijay Balasubramaniyan, Mustaque Ahamad, and Patrick Traynor. Improving Authentication Performance of Distributed SIP Proxies. In *Proceedings of the Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm)*, 2009. (Acceptance Rate: 43.3%).
66. Patrick Traynor, Kevin Butler, William Enck, and Patrick McDaniel. Realizing Massive-Scale Conditional Access Systems Through Attribute-Based Cryptosystems. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2008. (Acceptance Rate: 17.7%).
67. Patrick Traynor, Patrick McDaniel, and Thomas La Porta. On Attack Causality in Internet-Connected Cellular Networks. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2007. (Acceptance Rate: 12.3%).
68. Sunam Ryu, Kevin Butler, Patrick Traynor, and Patrick McDaniel. Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems. In *Proceedings of the IEEE International Symposium on Security in Networks and Distributed Systems (SSNDS)*, 2007. (Acceptance Rate: 40%).
69. Luke St. Clair, Lisa Johansen, William Enck, Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Trent Jaeger. Password Exhaustion: Predicting the End of Password Usefulness. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, 2006. (Invited Paper).
70. Kevin Butler, William Enck, Jennifer Plasterr, Patrick Traynor, and P. McDaniel. Privacy-Preserving Web-Based Email. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, December 2006. (Acceptance Rate: 30.4%).
71. Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure Attribute-Based Systems. In *Proceedings of the Thirteenth ACM Conference on Computer and Communications Security (CCS)*, November 2006. (Acceptance Rate: 14.8%).
72. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networks. In *Proceedings of the Twelfth Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2006. (Acceptance Rate: 11.7%).
73. Patrick Traynor, Michael Chien, Scott Weaver, Boniface Hicks, and Patrick McDaniel. Non-Invasive Methods for Host Certification. In *Proceedings of the Second IEEE International Conference on Security and Privacy in Communication Networks (SecureComm)*, August 2006. (Acceptance Rate: 25.4%).

74. Patrick Traynor, JaeShung Shin, Barat Madan, Shashi Phoha, and Thomas La Porta. Efficient Group Mobility for Heterogeneous Sensor Networks. In *Proceedings of the IEEE Vehicular Technology Conference (VTC Fall)*, September 2006. (Acceptance Rate: 58%).
75. Patrick Traynor, Raju Kumar, Hussain Bin Saad, Guohong Cao, and Thomas La Porta. LIGER: Implementing Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. In *Proceedings of the 4th ACM International Conference on Mobile Systems, Applications and Services (MobiSys)*, June 2006. (Acceptance Rate: 15.4%).
76. Patrick Traynor, Guohong Cao, and Thomas La Porta. The Effects of Probabilistic Key Management on Secure Routing in Sensor Networks. In *Proceedings of the 2006 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2006. (Acceptance Rate: 38.8%).
77. Patrick Traynor, Heesook Choi, Guohong Cao, Sencun Zhu, and Thomas La Porta. Establishing Pair-Wise Keys In Heterogeneous Sensor Networks. In *Proceedings of the 25th Annual IEEE Conference on Computer Communications (INFOCOM)*, April 2006. (Acceptance Rate: 18%).
78. William Enck, Patrick Traynor, Patrick McDaniel, and Thomas La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. In *Proceedings of the Twelfth ACM Conference on Computer and Communications Security (CCS)*, November 2005. (Acceptance Rate: 15%).

Removed for external version.

E.2. Conference Presentations with Proceedings (Non-Refereed)

None.

E.3. Conference Presentations without Proceedings

1. Patrick Traynor. Work in Progress Presentations: Fine-Grained Secure Localization for 802.11 Networks. 15th USENIX Security Symposium (SECURITY), August 2006.
2. Patrick Traynor. Work in Progress Presentations: Fundamental Limitations of Sensor Network Security. ACM/USENIX Fourth International Conference on Mobile Systems Applications and Services (MobiSys), June 2006. (Award: Most Entertaining WIP).
3. Patrick Traynor, Heesook Choi, Guohong Cao, and Thomas La Porta. Poster Session: Probabilistic Unbalanced Key Distribution and Its Effects on Distributed Sensor Networks. Workshop on Wireless Security (WiSe), October 2004.

F. Other

F.1. Submitted Journal Papers

None.

F.2. Refereed Research Reports

None.

F.3. Software

1. *GSM Air Interface Simulator*: Developed a full voice, data and SMS capable simulator for the wireless portion of a GSM network. Models communications down to the timeslot for highest possible accuracy. Used in the majority of our work on cellular security.

2. *Malicious Telephony Load Tester*: Built a system on top of the TM1 Telecom Database testing suite to allow for a comparison of malicious traffic of varying composition.

F.4. Published Papers (Non-Refereed)

1. Patrick Traynor. Characterizing the Limitations of Third-Party EAS Over Cellular Text Messaging Services. Technical report, 3G Americas Whitepaper, 2008.
2. Lisa Johansen, Kevin Butler, William Enck, Patrick Traynor, and Patrick McDaniel. Grains of SANs: Building Storage Area Networks from Memory Spots. Technical Report NAS-TR-0060-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, 2007.

F.5. Books in Preparation

None.

F.6. Workshops and External Courses

1. Luis Vargas, Patrick Emami, and Patrick Traynor. On the Detection of Disinformation Campaign Activity with Network Analysis. In *Proceedings of the 2020 ACM SIGSAC Cloud Computing Security Workshop, CCSW '20*, 2020.
2. Siddhant Deshmukh, Henry Carter, Grant Hernandez, Patrick Traynor, and Kevin Butler. Efficient and Secure Template Blinding for Biometric Authentication. In *IEEE Workshop on Security and Privacy in the Cloud (SPC)*, 2016.
3. Debayan Gupta, Benjamin Mood, Joan Feigenbaum, Kevin Butler, and Patrick Traynor. Using Intel Software Guard Extensions for Efficient Two-Party Secure Function Evaluation. In *Workshop on Encrypted Computing and Applied Homomorphic Cryptography (WAHC)*, 2016.
4. Chaitrali Amrutkar and Patrick Traynor. Rethinking Permissions for Mobile Web Apps: Barriers and the Road Ahead. In *Proceedings of the ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)*, 2012.
5. Nigel Lawrence and Patrick Traynor. Under New Management: Practical Attacks on SNMPv3. In *Proceedings of the USENIX Workshop on Offensive Technologies (WOOT)*, 2012.
6. Andrew Harris, Frank S. Park, Seymour Goodman, and Patrick Traynor. Emerging Privacy Concerns for Digital Wallet Deployment. In *Proceedings of the Workshop on Making Privacy in America*, 2009.
7. Patrick Traynor. Privacy and Security Concerns for Personal and Mobile Health Devices. In *Proceedings of the Workshop to Set A Research Agenda for Privacy and Security of Healthcare Technologies*, 2009.
8. Kevin Butler, William Enck, Harri Hursti, Stephen McLaughlin, Patrick Traynor, and Patrick McDaniel. Systemic Issues in the Hart InterCivic and Premier Voting System: Reflections Following Project EVEREST. In *Proceedings of the USENIX/ACCURATE Electronic Voting Technology (EVT) Workshop*, 2008.

G. Research Proposals and Grants (Principal Investigator)

1. Approved and Funded

1. Testing Audio Deep Fake Detectors

Sponsor: Bank of America

Investigator(s): Patrick Traynor (PI), Kevin Butler

Amount: \$274,000 over 2 years

Awarded: August 2021

2. Deploying Defenses for Cellular Networks Using the AWARE Testbed

Sponsor: Department of Homeland Security: CISA:

Investigator(s): Patrick Traynor (PI), Kevin Butler, Guofei Gu, Radu Stoleru, Walter Magnussen, P. R. Kumar

Amount: \$3,100,000 over 4 years

Awarded: October 2019

3. SaTC: CORE: Medium: Securing the Voice Processing Pipeline Against Adversarial Audio

Sponsor: NSF Secure and Trustworthy Cyberspace

Investigator(s): Patrick Traynor (PI), Thomas Shrimpton, Vincent Bindschaedler

Amount: \$1,199,999 over 4 years

Awarded: October 2019

4. Artus Protocol STTR Phase II

Sponsor: Office of Naval Research

Investigator(s): Patrick Traynor (PI), Kevin Butler

Amount: \$800,000 over 4 years

Awarded: August 2019

5. Evaluating the Security of QR Code-Based Payments

Sponsor: Discover Financial

Investigator(s): Patrick Traynor (PI)

Amount: \$50,000 over 1 year

Awarded: September 2018

6. Workshop: Addressing the Technical Security Challenges of Emerging Digital Financial Services

Sponsor: NSF Secure and Trustworthy Cyberspace

Investigator(s): Patrick Traynor (PI), Kevin Butler

Amount: \$50,000 over 1 year

Awarded: September 2017

7. Designing Strong End-to-End Authentication Mechanisms for Modern Telephony Systems

Sponsor: NSF Secure and Trustworthy Cyberspace

Investigator(s): Patrick Traynor (PI)

Amount: \$500,000 over 3 years

Awarded: July 2016

8. Digital Healthcare-Associated Infection: Measurement, Defense and Prevention in a Modern Digital Healthcare Ecosystem

Sponsor: National Science Foundation

Investigator(s): Patrick Traynor (PI), Kevin Butler, Shigang Chen

Amount: \$1,200,000 over 4 years

Awarded: June 2016

9. **Evaluating and Improving Security in Emerging Branchless Banking Systems**
Sponsor: NSF Secure and Trustworthy Cyberspace
Investigator(s): Patrick Traynor (PI)
Amount: \$500,000 over 3 years
Awarded July 2015
10. **Prevention and Detection of Disallowed Connections in Mobile and Pervasive Systems**
Sponsor: CISE-ECE Harris Endowed Seed Fund Program
Investigator(s): Patrick Traynor (PI), Renato Figueiredo (PI)
Amount: \$40,000 over 1 year
Awarded December 2014
11. **Mobile Excursion Study Support**
Sponsor: Hanscom AFB Electronic Systems Command Development Planning Division (ESC/XR)
Investigator(s): Patrick Traynor (PI), Mustaque Ahamad, Jeff Evans, Chuck Bokath
Amount: \$280,000 over 3 months
Awarded July 2012
12. **Characterizing the Security Limitations of Accessing the Mobile Web**
Sponsor: NSF Secure and Trustworthy Cyberspace
Investigator(s): Patrick Traynor (PI) and William Enck (NC State)
Amount: \$334,000 over 3 years
Awarded July 2012
13. **Mitigating Attacks on Mobile Devices and Critical Cellular Infrastructure**
Sponsor: US Department of Defense - Defense University Research Instrumentation Program (DURIP)
Investigator(s): Patrick Traynor (PI), Jon Giffin, Mustaque Ahamad
Amount: \$210,081 over 1 year
Awarded June 2011
14. **Characterizing and Implementing Efficient Primitives for Privacy-Preserving Computation**
Sponsor: DARPA PROgramming Computation on EncryptEd Data (PROCEED) – Broad Agency Announcement
Investigator(s): Patrick Traynor (PI) and Kevin Butler (UOregon)
Amount: \$580,000 over 4 years
Awarded May 2011
15. **Security for Converged IMS Networks**
Sponsor: US Department of Defense
Investigator(s): Patrick Traynor (PI), Mustaque Ahamad and Russ Clark
Amount: \$242,401 over 1 year
Awarded August 2010
16. **CAREER: Protecting User Data on Lost, Stolen and Damaged Mobile Phones**
Sponsor: NSF Trustworthy Computing
Investigator(s): Patrick Traynor (PI)
Amount: \$400,000 over 5 years
Awarded: May 2010
17. **Provably Anonymous Networking Through Secure Function Evaluation**
Sponsor: NSF Trustworthy Computing
Investigator(s): Patrick Traynor (PI)
Amount: \$200,000 over 2 years
Awarded: July 2009

18. **Characterizing and Mitigating Device-Based Attacks in Cellular Telecommunications Networks**
Sponsor: NSF Trustworthy Computing
Investigator(s): Patrick Traynor (PI) and Jonathon Giffin
Amount: \$450,000 over 3 years
Awarded: July 2009

2. Pending

Removed for external version.

H. Research Proposals and Grants (Contributor)

1. Approved and Funded

1. **SaTC: Frontier: Securing the Future of Computing for Marginalized and Vulnerable Populations**
Sponsor: NSF SaTC
Investigator(s): Kevin Butler (PI), Patrick Traynor, Tadayoshi Kohno, Franzi Roesner, Apu Kapadia, Eakta Jain.
Amount: \$7,500,000 for 5 years
Awarded October 2022
2. **ROCKY: Reliable Obfuscated Communications Kit for everYone**
Sponsor: DARPA Resilient Anonymous Communication for Everyone (RACE) – Broad Agency Announcement
Investigator(s): Thomas Shrimpton (PI), Patrick Traynor, Kevin Butler, Vincent Bindschaedler, Nadia Heninger
Amount: \$1,600,000 over 4 years
Awarded May 2019
3. **WiFiUS: Collaborative Research: SELIOT: Securing Lifecycle of Internet-of-Things**
Sponsor: NSF CNS WiFiUS
Investigator(s): Gene Tsudik (PI), Patrick Traynor
Amount: \$300,000 for 2 years
Submitted December 2016
4. **Cloud-based Oblivious Spectrum Mapping and Allocation**
Sponsor: NSF CNS EARS
Investigator(s): John Shea (PI), Tan Wong, Patrick Traynor
Amount: \$532,952 for 2 years
Submitted May 2016
5. **DURIP: Developing Research Capability in Cyber-Physical Systems at the University of Florida**
Sponsor: Small
Investigator(s): Kevin Butler (PI), Patrick Traynor, My Thai
Amount: \$200,000 for 2 years
Submitted: June 2015
6. **Securing the New Converged Telephony Landscape**
Sponsor: NSF TWC: Small
Investigator(s): Mustaque Ahamad (PI) and Patrick Traynor
Amount: \$500,000 for 3 years
Submitted: December 2012

7. **Facilitating Free and Open Access to Information on the Internet**
 Sponsor: NSF Trustworthy Computing
 Investigator(s): Nick Feamster (PI), Wenke Lee, Patrick Traynor, Hans Klein, Roger Dingledine, Michael Freedman and Edward W. Felten
 Amount: \$1,500,000 for 4 years
 Awarded: June 2011
8. **Monitoring Free and Open Access to Information on the Internet**
 Sponsor: Google Focus Program
 Investigator(s): Nick Feamster (PI), Wenke Lee, Mustaque Ahamad, Patrick Traynor, Henry Owen, Ellen Zegura, Zvi Galil
 Amount: \$1,000,000 for 2 years
 Awarded: November 2011
9. **Dynamic-attribute-based Disclosure of Health Information in Emergency Care Scenarios**
 Sponsor: Health Systems Institute (HSI) Seed Grant Program
 Investigator(s): Doug Blough (PI), Mustaque Ahamad, Patrick Traynor and Jim Jose
 Amount: \$50,000 over 1 year
 Awarded: August 2009
10. **Federal Cyber Service Scholarships at Georgia Tech**
 Sponsor: NSF SFS Scholarships
 Investigator(s): Seymour Goodman (PI), Patrick Traynor
 Amount: \$1,250,682 over 5 years
 Awarded: June 2009
11. **Security for IMS-Enabled Converged Applications**
 Sponsor: US Department of Defense
 Investigator(s): Mustaque Ahamad (PI), Patrick Traynor (PI), Michael Hunter, Russ Clark
 Amount: \$146,121 for 1 year
 Awarded: August 2008

2. Pending

Removed for external version.

I. Research Honors and Awards

1. Fellow, Center for Financial Inclusion at Accion, 2017.
2. Sloan Research Fellow, Alfred P. Sloan Foundation, 2014.
3. Best Paper, The ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec); Budapest, Hungary, 2013.
4. Best Student Paper, The Information Security Conference (ISC); Passau, Germany, 2012
5. Lockheed Inspirational Young Faculty Award, 2012
6. Best Demo, "Is Browsing the Internet on Your Mobile Phone Secure?" Chaitrali Amrutkar (Ph.D Advisee), CoC Research Day, 2011
7. Best Poster, "(sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers" Arunabh Verma, Henry Carter (MS, Ph.D Advisees), CoC Research Day, 2011
8. National Science Foundation CAREER Award, 2010

9. Pennsylvania State University Alumni Association Dissertation Award, 2007
10. Pennsylvania State University CSE Graduate Research Assistant Award, 2007
11. AT&T Wireless Fellowship, 2005

III. SERVICE

A. Professional Activities

A.1. Memberships and Activities in Professional Societies

1. Senior Member, Association for Computing Machinery (ACM)
2. Senior Member, Institute of Electrical and Electronics Engineers (IEEE)
3. Member, USENIX Advanced Computing Systems Association (USENIX)

A.2. Conference Committee Activities

1. Program co-Chair, *IEEE Symposium on Security and Privacy (OAKLAND)*: 2023, 2024
2. Program co-Chair, *USENIX Security Symposium (SECURITY)*: 2019
3. Program co-Chair, *Network and Distributed System Security Symposium (NDSS)*: 2017, 2018
4. Program Chair, *USENIX Workshop on Offensive Technologies (WOOT)*: 2016
5. Program Chair, *ACM Conference on Wireless Network Security (WiSec)*: 2014
6. Program Co-Chair, *Annual Computer Security Applications Conference (ACSAC)*: 2012, 2013
7. Program Chair, *USENIX Workshop on Hot Topics in Security (HotSec)*: 2012
8. Chair Invited Talks Committee, *USENIX Security Symposium (SECURITY)*: 2014
9. Workshops Chair, *IEEE Conference on Communications and Network Security (CNS)*: 2016
10. Program Committee, *USENIX Security Symposium (SECURITY)*: 2008, 2009, 2010, 2013, 2015-2018, 2020-2022
11. Program Committee, *IEEE Symposium on Security and Privacy (OAKLAND)*: 2009-2014, 2022.
12. Program Committee, *ACM Conference On Computer and Communications Security (CCS)*: 2009, 2013-2015, 2017
13. Program Committee, *Network and Distributed System Security Symposium (NDSS)*: 2010, 2013-2016, 2020-2021
14. Program Committee, *IEEE European Symposium on Security and Privacy (Euro S&P)*: 2016
15. Program Committee, *Annual Computer Security Applications Conference (ACSAC)*: 2008, 2009, 2010, 2011, 2015
16. Program Committee, *ACM Conference on Wireless Network Security (WiSec)*: 2009, 2010, 2013, 2015-2021
17. Program Committee, *International Conference on Financial Cryptography and Data Security (FC)*: 2010, 2013
18. Program Committee, *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*: 2016.
19. Program Committee, *ICST Conference on Security and Privacy in Communication Networks (SecureComm)*: 2009, 2010
20. Program Committee, *Privacy Enhancing Technologies Symposium (PETS)*: 2015, 2016

21. Program Committee, *International World Wide Web Conference (WWW)*: 2016
22. Program Committee, *USENIX Workshop on Hot Topics in Security (HotSec)*: 2011
23. Program Committee, *ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MOBIHELD)*: 2010
24. Program Committee, *International Workshop on Mobile Security (WMS)*: 2010
25. Program Committee, *European Symposium on Research in Computer Security (ESORICS)*: 2009, 2011
26. Program Committee, *IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS)*: 2009, 2010
27. Program Committee, *Information Security Conference (ISC)*: 2010
28. Program Committee, *IEEE International Conference on Information Privacy, Security, Risk and Trust (PASSAT)*: 2009
29. Program Committee, *Computer Security Architecture Workshop (CSAW)*: 2008
30. Program Committee, *IWCMC Computer and Network Security Symposium*: 2009
31. Program Committee, *IARIA International Conference on Internet Monitoring and Protection (ICIMP)*: 2009
32. Program Committee, *IEEE Workshop on Network Security and Privacy (NSP)*: 2008
33. Program Committee, *IEEE International Workshop on Wireless and Sensor Networks Security (WSNS)*: 2008, 2009
34. Program Committee, *IEEE Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC)*: 2008
35. Program Committee, *ACM Conference on Computer and Communications Security, Industry and Government Track (CCS I&G)*: 2006, 2007
36. Program Committee, *Workshop on Secure Network Protocols (NPsec)*: 2006
37. Program Committee, *International Conference on Information Systems Security (ICISS)*: 2006, 2009, 2010
38. Program Committee, *IEEE LCN Workshop on Network Security (WNS)*: 2006, 2007, 2008

B. On-Campus Committees

B.1. University of Florida

1. Member, Computer and Information Science and Engineering Steering Committee, 2015-2017.
2. Member, Graduate Recruiting Committee, 2015-2017.
3. Chair, Computer and Information Science and Engineering Industrial Advisory Board, 2014-2015.

B.2. Georgia Tech

1. Member, Massive Open Online Master's (MOOMS) Investigation Committee, 2012-2013.
2. Chair, School of Computer Science Ph.D. Review Committee, 2012.
3. Member, School of Computer Science Ph.D Review Committee, 2011.
4. Faculty Advisor, Grey H@T - Georgia Tech Undergraduate Security Club, 2011-2014.
5. Member, School of Computer Science Ph.D. Review Committee, 2011.
6. Member, School Advisory Committee, School of Computer Science, 2011-2013.
7. Member, School of Computer Science Chair Recruiting Committee, 2011.
8. Member, School of Computer Science Faculty Recruiting Committee, 2010, 2011.
9. Chair, College of Computing Ph.D. Welcome Weekend Committee, 2009, 2010, 2011 (co-chair).
10. Member, College of Computing Ph.D. Recruiting Committee, 2009.
11. Member, Georgia Tech Computer and Network Usage Security Policy (CNUSP) Evaluation Group, 2009.

C. Special Assignments

None.

D. Ph.D. Examining Committees

Ph.D. Examining Committees

1. Bradley Reaves, Department of Computer and Information Science and Engineering, University of Florida, Summer 2017.
Advisor: Professor Patrick Traynor.
2. Adam Bates, Department of Computer and Information Science and Engineering, University of Florida, Spring 2016.
Advisor: Professor Kevin Butler.
3. Benjamin Mood, Department of Computer and Information Science and Engineering, University of Florida, Spring 2016.
Advisor: Professor Kevin Butler.
4. Henry Carter, College of Computing, Georgia Tech, Fall 2015.
Advisor: Professor Patrick Traynor.
5. David Dewey, College of Computing, Georgia Tech, Fall 2015.
Advisor: Professor Patrick Traynor.
6. Lateef Yusuf, College of Computing, Georgia Tech, Spring 2014.
Advisor: Professor Umakishore Ramachandran.
7. Chaitrali Amrutkar, College of Computing, Georgia Tech, Summer 2013.
Advisor: Professor Patrick Traynor.
8. Long Lu, College of Computing, Georgia Tech, Summer 2013.
Advisor: Professor Wenke Lee.

9. Manos Antonakakis, College of Computing, Georgia Tech, Summer 2012.
Advisor: Professor Wenke Lee.
10. Junjie Zhang, College of Computing, Georgia Tech, Summer 2012.
Advisor: Professor Wenke Lee.
11. Italo Dacosta, College of Computing, Georgia Tech, Summer 2012.
Advisor: Professor Patrick Traynor.
12. Virendra Kumar, College of Computing, Georgia Tech, Summer 2012.
Advisor: Professor Alexandra Boldyreva.
13. Anirudh Ramachandran, College of Computing, Georgia Tech, Summer 2011.
Advisor: Professor Nick Feamster.
14. Vijay Balasubramaniyan, College of Computing, Georgia Tech, Summer 2011.
Advisor: Professor Mustaque Ahamad.
15. Kapil Singh, College of Computing, Georgia Tech, Summer 2011.
Advisor: Professor Wenke Lee.
16. Abhinav Srivastava, College of Computing, Georgia Tech, Summer 2011.
Advisor: Professor Jon Giffin.
17. Adam O'Neill, College of Computing, Georgia Tech, Summer 2010.
Advisor: Professor Alexandra Boldyreva.
18. David Cash, College of Computing, Georgia Tech, Fall 2009.
Advisor: Professor Alexandra Boldyreva.

External Member of Ph.D. Research Committee

None.

External Member of Ph.D. Examining Committee

1. Shannon Eggers, Department of Materials Sciences and Engineering - Nuclear Engineering Program, University of Florida, Fall 2016.
Advisor: Professor Kelly Jordan.
2. Ed Carlisle, Department of Electrical and Computer Engineering, University of Florida, Summer 2016.
Advisor: Professor Alan George.
3. Claudio Marforio, Department of Computer Science, Swiss Federal Institute of Technology Zurich (ETH Zurich), Fall 2015.
Advisor: Professor Srdjan Capkun.
4. Nils Ole Tippenhauer, Department of Computer Science, Swiss Federal Institute of Technology Zurich (ETH Zurich), Spring 2012.
Advisor: Professor Srdjan Capkun.
5. Bongkyoung Kwon, School of Electrical and Computer Engineering, Georgia Tech, Summer 2009.
Advisor: Professor John Copeland.

Ph.D. Thesis Proposal Committees

1. Bradley Reaves, Department of Computer and Information Science and Engineering, Spring 2016.
Advisor: Professor Patrick Traynor.
2. Maliheh Shirvanian, University of Alabama, Birmingham, Spring 2016.
Advisor: Professor Nitesh Saxena.
3. Benjamin Mood, Department of Computer and Information Science and Engineering, Fall 2015.
Advisor: Professor Kevin Butler.
4. Adam Bates, Department of Computer and Information Science and Engineering, Fall 2015.
Advisor: Professor Kevin Butler.
5. Lateef Yusuf, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Umakishore Ramachandran.
6. Long Lu, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Wenke Lee.
7. Chaitrali Amrutkar, College of Computing, Georgia Tech, Fall 2012.
Advisor: Professor Patrick Traynor.
8. Junjie Zhang, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Wenke Lee.
9. Italo Dacosta, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Patrick Traynor.
10. Manos Antonakakis, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Wenke Lee.
11. Abhinav Srivastava, College of Computing, Georgia Tech, Spring 2011.
Advisor: Professor Jon Giffin.
12. Vijay Balasubramaniyan, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Mustaque Ahamad.
13. Kapil Singh, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Wenke Lee.
14. Anirudh Ramachandran, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Nick Feamster.
15. Adam O'Neill, College of Computing, Georgia Tech, Spring 2010.
Advisor: Professor Alexandra Boldyreva.
16. David Cash, College of Computing, Georgia Tech, Spring 2009.
Advisor: Professor Alexandra Boldyreva.

Ph.D. Qualifying Exam Committees—Georgia Tech

1. Byoungyoung Lee, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Wenke Lee.
2. Yizheng Chen, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Wenke Lee.

3. Xinyu Xing, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Wenke Lee.
4. Brad Reaves, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Patrick Traynor.
5. Chaz Lever, College of Computing, Georgia Tech, Spring 2013.
Advisor: Professor Patrick Traynor.
6. Terry Nelms, College of Computing, Georgia Tech, Spring 2012.
Advisor: Professors Mustaque Ahamad and Roberto Perdesi.
7. Saurabh Chakradeo, College of Computing, Georgia Tech, Spring 2012.
Advisor: Professor Patrick Traynor.
8. Henry Carter, College of Computing, Georgia Tech, Spring 2012.
Advisor: Professor Patrick Traynor.
9. David Dewey, College of Computing, Georgia Tech, Spring 2012.
Advisor: Professor Jon Giffin.
10. Chaitrali Amrutkar, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Patrick Traynor.
11. Yacin Nadji, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Wenke Lee.
12. Yogesh Mundada, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Nick Feamster.
13. Hyojoon Kim, College of Computing, Georgia Tech, Fall 2011.
Advisor: Professor Nick Feamster.
14. Ikpeme Erete, College of Computing, Georgia Tech, Spring 2011.
Advisor: Professor Alex Orso.
15. Chaitrali Amrutkar, College of Computing, Georgia Tech, Spring 2011.
Advisor: Professor Patrick Traynor.
16. Brendan Dolan-Gavitt, College of Computing, Georgia Tech, Spring 2011.
Advisor: Professor Wenke Lee and Professor Jon Giffin.
17. Sam Burnett, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Nick Feamster.
18. Cong Shi, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Mostafa Ammar and Professor Ellen Zegura.
19. Partha Kanuparth, College of Computing, Georgia Tech, Fall 2010.
Advisor: Professor Constantine Dorvolis.
20. Long Lu, College of Computing, Georgia Tech, Spring 2010.
Advisor: Professor Wenke Lee.
21. Virendra Kumar, College of Computing, Georgia Tech, Spring 2009.
Advisor: Professor Alexandra Boldyreva.
22. Frank Park, College of Computing, Georgia Tech, Spring 2009.
Advisor: Professor Patrick Traynor.

23. Italo Dacosta, College of Computing, Georgia Tech, Fall 2008.
Advisor: Professor Mustaque Ahamad and Professor Patrick Traynor.
24. Adam O'Neill, College of Computing, Georgia Tech, Fall 2008.
Advisor: Professor Alexandra Boldyreva.

E. External Member of M.S. Examining Committee

M.S. Thesis Defense Committees None.

F. Consulting and Advisory Appointments

1. Skim Reaper, *Co-Founder and CEO*, 2019-Present.
2. CryptoDrop Anti-Ransomware, *Co-Founder and CEO*, 2017-2018.
3. Pindrop Security, *Research Fellow and Co-Founder*, Spring 2012 - Spring 2014.
4. United States Army (via US Falcon), *Information Assurance Officer Training Program*, Spring 2010.
5. 3G Americas, *Characterizing the Limitations of Third-Party EAS over Cellular Text Messaging Systems*, Fall 2008.

G. Civic Activities

None.

IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION

A. Honors and Awards

1. Fellow, Kavli Foundation, 2017.
2. Fellow, Center for Financial Inclusion at Accion, 2016.
3. Sloan Research Fellow, Alfred P. Sloan Foundation, 2014.

B. Invited Conference Session Chairmanships

1. Session Chair, *Work-in-Progress* at the *USENIX Security Symposium (SECURITY)*, 2016.
2. Session Chair, *Mobile Security* at the *USENIX Security Symposium (SECURITY)*, 2013.
3. Poster Chair, *USENIX Security Symposium (SECURITY)*, 2010, 2011.
4. Session Chair, *Privacy and Anonymity* at the *USENIX Workshop on Hot Topics in Security (HotSec)*, 2011.
5. Session Chair, *Security of Authentication and Protection Mechanisms* at the *IEEE Symposium on Security & Privacy (OAKLAND)*, 2011.
6. Session Chair, *Information Abuse* at the *IEEE Symposium on Security & Privacy (OAKLAND)*, 2010.
7. Session Chair, *RFID Security* at the *ACM Conference on Computer and Communications Security (CCS)*, 2009.
8. Session Chair, *Browser Security Session* at the *USENIX Security Symposium (SECURITY)*, 2009.
9. Session Chair, *Information Security Session* at the *IEEE Symposium on Security and Privacy (OAKLAND)*, 2009.
10. Session Chair, *Work-in-Progress* at the *IEEE Symposium on Security and Privacy (OAKLAND)*, 2009.
11. Session Chair, *Work/Opinions-in-Progress* at the *ISOC Network and Distributed Systems Security (NDSS) Symposium*, 2009.
12. Session Chair, *Privacy Session* at the *USENIX Security Symposium (SECURITY)*, 2008.

C. Professional Registration

None.

D. Patents

1. Patrick G. Traynor, Christian Peeters, Bradley G. Reaves, Hadi Abdullah, Kevin Butler, Jasmine Bowers, Walter N. Scaife, "Detecting SS7 Redirection Attacks With Audio-Based Distance Bounding", United State Patent # 11,265,717, Filed March 2019, Issued March 2022.
2. Patrick G. Traynor, Logan E. Blue, Luis Vargas, "Method and Apparatus for Differentiating Between Human and Electronic Speaker for Voice Interface Security", United State Patent # 11,176,960, Filed June 2019, Issued November 2021.
3. Patrick G. Traynor, Bradley G. Reaves, Logan E. Blue Practical End-to-End Cryptographic Authentication for Telephony Over Voice Channels, United State Patent # 11,329,831, Filed November 2018, Issued May 2022.

4. Walter Nolen Scaife, Patrick G. Traynor and Christian Peeters, "Payment Card Overlay Skimmer Detection", United States Patent # 10,496,914, Filed October 2017, Issued December 2019. (See also # 10,936,928)
5. Patrick G. Traynor, David P. Arnold, Walter Nolen Scaife, Christian Peeters, and Camilo Valez Cuervo, "Detecting counterfeit magnetic stripe cards using encoding jitter", United States Patent # 10,803,261, Filed May 2017, Issued October 2020.
6. Patrick G. Traynor, Bradley Reaves, Logan Blue, Luis Vargas, Hadi Abdullah, and Thomas Shrimpton, "Identity and content authentication for phone calls", United States Patent # 10,764,043, Filed Apr 2017, Issued September 2020.
7. Walter Nolen Scaife, Henry Carter, Patrick G. Traynor and Kevin R. B. Butler. "Malware Detection Through User Data Transformation Monitoring", United States Patent # 10,685,114. Filed September 2015, Issued June 2020.
8. Vijay A. Balasubramaniyan, Mustaque Ahamad, Patrick G. Traynor. "Using Single-Ended Audio Features to Automatically Determine Voice Call Provenance", United States Patent, #9,037,113 June 2010, Issued May 2015. (See also #9,516,497 and #10,523,809)
9. Patrick G. Traynor, Byungsuk Kim and Farooq Anjum. "Secure Localization for 802.11 Networks with Fine Granularity", United States Patent, #8,107,400, Filed July 2008, Issued January 2012.

E. Editorial and Reviewer Work for Technical Journals and Publishers

Associate Editor:

- *ACM Transactions on Information and System Security (TISSEC)* 2015-present

Guest Editor:

Journals

- *IEEE Security and Privacy Magazine (S&P)* 2013

Reviewer for:

Journals

- *ACM Transactions on Information and System Security (TISSEC)* 2008, 2009, 2010, 2011, 2012, 2013
- *IEEE Transactions on Dependable and Secure Computing (TDSC)* 2012, 2013
- *IEEE Security and Privacy Magazine (S&P)* 2010, 2011
- *Communications of the ACM (CACM)* 2010
- *Journal of Anesthesia & Analgesia* 2009
- *IEEE Transactions on Mobile Computing (TMC)* 2008, 2010, 2011, 2012, 2013
- *IEEE Transactions on Internet Technology (TOIT)* 2009, 2010
- *ACM Mobile Computing and Communications Review (MC2R)* 2008
- *IEEE/ACM Transactions on Networking (TON)* 2007, 2008
- *Journal of Pervasive and Mobile Computing (PMC)* 2009, 2010

- *IEEE Transactions on Parallel and Distributed Systems (TPDS)* 2005, 2009, 2010
- *IEEE Transactions on Computers (TOC)* 2010
- *Journal of Security and Communication Networks (SCN)* 2008
- *IEEE Communications Letters (CL)* 2007, 2009
- *IEEE Transactions on Wireless Communications (TWC)* 2007
- *Pervasive and Mobile Computing (PMC)* 2007
- *IEEE Transactions on Software Engineering (TSE)* 2007, 2008
- *Journal of Wireless Networks (WiNet)* 2006, 2007, 2008, 2009
- *Journal of Wireless Communications and Mobile Computing* 2006
- *ACM Computing Surveys (ACMCS)* 2006
- *Information Processing Letters (IPL)* 2006
- *IEEE Transactions on Very Large Scale Integration Systems (TVLSIS)* 2006

Conferences and Workshops

- *ACM International Conference on Mobile Computing and Networking (MobiCom)*, 2011
- *ACM Conference on Computer and Communications Security (CCS)*, 2008, 2011
- *IEEE Symposium on Security and Privacy (OAKLAND)* 2007, 2008
- *Computer Security Foundations (CSF)*, 2011
- *IFIP Conference on Data and Applications Security (DBSec)* 2008
- *Financial Cryptography (FC)* 2007, 2008
- *International Conference on VLSI Design (VLSI)* 2007
- *Annual Computer Security Applications Conference (ACSAC)* 2005, 2006, 2007
- *USENIX Workshop on Hot Topics in Security (HotSec)* 2007
- *International Conference on Information Systems Security (ICISS)* 2007
- *IEEE International Conference on Computer Engineering & Systems (ICCES)* 2007
- *International Workshop on Security (IWSec)* 2006, 2007
- *USENIX Security Symposium (SECURITY)* 2006, 2007
- *IEEE Sarnoff Symposium (SARNOFF)* 2007
- *International Conference on New Technologies, Mobility and Security (NTMS)* 2007
- *IEEE Infocom (INFOCOM)* 2007
- *Network and Distributed System Security Symposium (NDSS)* 2007
- *International Workshop on Storage Security and Survivability (IWSSS)* 2006
- *ACM Conference on Computer and Communications Security (CCS)* 2006

- *IEEE GLOBECOM (GLOBECOM) 2006*
- *International Conference on Mobile and Ubiquitous Systems: Networks and Services (MOBIQUITOUS) 2006*
- *IFIP Conference on Data and Applications Security (DBSec) 2006*
- *Emerging Trends in Information and Communications Security (ETRICS) 2006*
- *International Conference on Applied Cryptography and Network Security (ACNS) 2006*
- *ACM Symposium on Access Control Models and Technology (SACMAT) 2006*
- *IEEE Conference on Communication Systems Software & Middleware (COMSWARE) 2006*
- *International Conference on Cryptology in India (IndoCrypt) 2005*
- *IEEE Symposium on New Frontiers in Dynamic Spectrum Access (DySPAN) 2005*
- *European Symposium on Research in Computer Security (ESORICS) 2005*

F. Expert Witness Services

1. *Epic Games, Inc. & Anor v Google LLC & Ors - Federal Court of Australia Proceeding NSD 190 of 2021:* Expert witness for the Defense (via Corrs Chambers Westgarth). Status: Ongoing. *January 2023 - Present.*
2. *Telefonaktiebolaget LM Ericsson vs Apple, Inc:* Expert witness for the Defendant, Non-Infringement and Invalidity (via WilmerHale LLP). Status: Settled. *February 2022 - December 2022.*
3. *Wepay Global Payments, LLC v. Bank of America N. A.:* Expert Witness for the Defendant (via WilmerHale LLP) Status: Dismissed. *September 2022 - November 2022.*
4. *Apple vs. R.N Nehushtan Trust Ltd.:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). Status: Submitted to PTAB. *August 2022 - Ongoing.*
5. *Apple/Microsoft vs. Zipit Wireless:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). Status: Settled. *May 2021 - November 2022.*
6. *Blackberry Inc v MobileIron, Inc:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). Verdict: Settled. *January 2021 - March 2021.*
7. *Apple Inc v Seven Networks, LLC:* Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). Verdict: Three of four petitions instituted by PTAB. Fourth rejected via discretion, case settled. *August 2019 - November 2020.*
8. *mSIGNIA, Inc. v. InAuth, Inc.:* Expert Witness for the Defendant for Inter Partes Review, Non-Infringement and Invalidity (via Quinn Emanuel Urquhart and Sullivan, LLP). Verdict: Dismissed with prejudice. *October 2017 - December 2018.*
9. *Huawei v. T-Mobile:* Expert Witness for the Defendant for Non-Infringement (via WilmerHale LLP, Alston & Bird LLP) Verdict: Settled *June 2016 - December 2017.*
10. *Telefonaktiebolaget LM Ericsson v Apple:* Expert Witness for the Defendant for Non-Infringement, Invalidity (via WilmerHale LLP). Verdict: Settled *June 2015 - December 2015.*
11. *Mayfonk v Nike:* Expert Witness for the Plaintiff for Infringement (via Paul Hastings). Verdict: Settled. *June 2015 - November 2015.*

12. *Maxim Integrated Products v Bank of the West*: Expert Witness for the Defendant for Non-Infringement (via Paul Hastings LLP). Verdict: Dismissed with prejudice. *January 2014 - August 2014*.
13. *Maxim Integrated Products v Comerica Inc, et al*: Expert Witness for the Defendant for Non-Infringement (via McKenna, Long & Aldridge LLP). Verdict: Settled. *June 2014 - August 2014*.
14. *William Grecia v. Apple Inc. et al*: Expert Consultant for the Defendant for Invalidity (via Kirkland & Ellis LLP). Verdict: Closed in initial pleadings, dismissed with prejudice. *July 2014 - August 2014*.
15. *Intertrust Technologies Corp. v. Apple Inc.*: Expert Consultant for Defendant for Invalidity and Non-Infringement (via Kirkland & Ellis LLP). Verdict: Settled. *October 2013 - February 2014*.
16. *Maxim Integrated Products v KeyCorp Bank*: Expert Witness for the Defendant for Non-Infringement (via Calfee, Halter & Griswold LLP) Verdict: Settled. *April 2013 - June 2013*.
17. *Intellectual Ventures LLC vs. Check Point; et al.*: Expert Consultant for the Plaintiff for Infringement (via Susman Godfrey LLP), Verdict: Infringement on 2 of 4 patents. *October 2012 - February 2015*.

V. OTHER CONTRIBUTIONS

A. Seminar Presentations (Invited Papers and Talks at Meetings and Symposia)

1. Keynote: Well, It Worked on My Computer: Reproducibility, Tech Transfer, and Computer Security Research. National Science Foundation Secure and Trustworthy Cyberspace (SaTC) Vision 2.0 Workshop, March 2023. University of Texas at Dallas.
2. Humans vs The Computer Interfaces: Separating Deepfakes/Bots from People Using Psychoacoustics. UCLA Electrical and Computer Engineering Distinguished Seminar, February 2023. University of California, Los Angeles.
3. Keynote: Exploiting the Gaps Between Human and Machine Understanding of Audio: Frameworks, Attacks, and Defenses. ISCA Symposium on Security and Privacy in Speech Communication (SPSC), November 2021. Virtual.
4. The State of Voice Cloning Technology. Federal Trade Commission (FTC) Workshop on Voice Cloning Technologies, January 2020. Washington, DC.
5. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. North Carolina State University Department of Computer Science Colloquium, January 2020. Raleigh, NC.
6. Moving from research to practice: How to maximize the impact of SaTC projects. National Science Foundation Secure and Trustworthy Cyberspace (SaTC) PI Meeting, October 2019. Alexandria, VA.
7. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. Purdue University Computer Science Excellence Lecture Series, October 2019. West Lafayette, IN.
8. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. Bank of America - Colloquium Series, March 2019. Charlotte, NC.
9. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. CISPA – Helmholtz Center for Information Security, Saarland University, February 2019. Saarbrücken, Germany.
10. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. University of Maryland - Distinguished Colloquium, February 2019. College Park, MD.
11. Responsible Finance for the Digital Client. Foromic Conference, October 2018. Barranquilla, Colombia.
12. Panel: Authentication Challenges for New Interfaces, Devices, and Wireless Networks. ACM Conference on Security and Privacy in Wireless and Mobile Networks, June 2018. Stockholm, Sweden.
13. Sonar: Detecting SS7 Redirection Attacks Via Call Audio-Based Distance Bounding. CyberSecurity@KAIST Workshop - KAIST, June 2018. Daejeon, South Korea.
14. Why Caller-ID Spoofing Is So Easy (and Why End-To-End Solutions Are the Way Forward). IEEE Workshop on Technology and Consumer Protection (ConPro'18), May 2018. San Francisco, CA.
15. Panel: The Future of Cybersecurity. SEC Academic Conference - Auburn University, May 2018. Auburn, AL.
16. Sound Principles: Verifying Voice Commands in an IoT World. IoT Security Workshop - Aalto University, September 2017. Helsinki, Finland.

17. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Eurecom Institute, September 2017. Sophia Antipolis, France.
18. Panel: Infrastructure Stability. ITU-T Focus Group Digital Financial Services, December 2016. Geneva, Switzerland.
19. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. ETH Zurich, December 2016. Zurich, Switzerland.
20. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. University of Richmond, October 2016. Richmond, Virginia.
21. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Indiana University, September 2016. Bloomington, Indiana.
22. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Aalto University Computer Science Department Forum, August 2016. Helsinki, Finland.
23. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. KAIST Information Security Seminar - Korean Advanced Institute of Science and Technology, June 2016. Daejeon, South Korea.
24. Updated Mobile Money Vulnerability Report. International Telecommunications Union Digital Financial Services Working Group Workshop, May 2016. Washington, DC.
25. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. UF Eye Opener Discovery Breakfast - University of Florida, May 2016. Gainesville, FL.
26. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Illinois Science of Security (SoS) Lablet Speaker Series - University of Illinois, Urbana-Champaign, April 2016. Urbana-Champaign, Illinois.
27. New Trends in Cybersecurity: Vulnerabilities in Branchless Banking Systems. United States Departments of State and Justice - Cybersecurity and Cybercrime Workshop for Lusophone Africa, September 2015. Maputo, Mozambique.
28. New Trends in Cybersecurity: Vulnerabilities in Branchless Banking Systems. United States Departments of State and Justice - ECCAS Cybersecurity and Cybercrime Workshop, August 2015. Kinshasa, Democratic Republic of Congo.
29. Chasing Telephony Security: Where the Wild Things... Are? University of Florida - Department Colloquium, January 2014. Gainesville, FL.
30. Chasing Telephony Security: Where the Wild Things... Are? Verizon Wireless RNC/Data Center, October 2013. Alpharetta, GA.
31. Chasing Telephony Security: Where the Wild Things... Are? University of Waterloo - CrySP Speaker Series on Privacy, October 2013. Waterloo, ON, Canada.
32. Analyzing Malicious Traffic in Cellular Networks. GSM Association's (GSMA) Mobile Malware Community Workshop, July 2013. Mountain View, CA.
33. Threats to Mobile Devices. US Federal Trade Commission (FTC) Public Forum - Invited Speaker, June 2013. Washington, D.C.
34. Chasing Telephony Security: Where the Wild Things... Are? University of Wisconsin - Madison, Security Seminar, March 2013. Madison, WI.

35. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. Invited Talk: Centre for Secure Information Technologies (CSIT) Queen's University, March 2013. Belfast, Northern Ireland.
36. Chasing Telephony Security: Where the Wild Things... Are? Stanford Security Seminar, March 2013. Stanford, CA.
37. Chasing Telephony Security: Where the Wild Things... Are? University of California, Berkeley, Security Group, March 2013. Berkeley, CA.
38. Chasing Telephony Security: Where the Wild Things... Are? Carnegie Mellon University CyLab Seminar, February 2013. Pittsburgh, PA.
39. Chasing Telephony Security: Where the Wild Things... Are? University of Oregon Department of Computer Science Colloquium, November 2012. Eugene, OR.
40. Chasing Telephony Security: Where the Wild Things... Are? University of Washington Department of Electrical Engineering, Network Security Lab (NSL): Invited Talk, November 2012. Seattle, WA.
41. Needles and Haystacks: Digging for Ground Truth on Mobile Malware. ZISC Workshop on Secure Mobile and Cloud Computing, ETH Zurich, June 2012. Zurich, Switzerland.
42. Panel: Advice for Early Career Faculty. CRA Career Mentoring Workshop, February 2012. Washington, D.C.
43. Research Challenges in Cellular and Mobile Network Security. US-China Software Workshop (Co-Sponsored by NSF and NSFC), September 2011. Beijing, China.
44. Mobile Security: Understanding Risks to Critical Infrastructure. Invited Talk: US Department of State East African Workshop on Cyberspace Security, July 2011. Nairobi, Kenya.
45. Tomorrow's Issues: Solving the Mobile Security Threat. Invited Talk: Centre for Secure Information Technologies (CSIT) Queen's University, March 2011. Belfast, Northern Ireland.
46. PinDr0p: Using Single-Ended Audio Features to Determine Call Provenance. Invited Talk: MITRE Corporation, March 2011. Burlington, MA.
47. Defeating Session Hijacking Attacks with Disposable Web Credentials. Invited Talk: Facebook, February 2011. Palo Alto, CA.
48. Understanding the Disruptive Potential of Malware in Cellular Networks. Invited Talk: RSA Conference, February 2011. San Francisco, CA.
49. Panel: Voice Security – Now Just a False Sense of Security and Privacy. Invited Panelist: Mobile Security Symposium, February 2011. San Francisco, CA.
50. Understanding the Disruptive Potential of Malware in Cellular Networks. Invited Talk: Concordia University, May 2010. Montreal, QC, Canada.
51. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Qualcomm Research, March 2010. San Diego, CA.
52. Privacy and Security Concerns for Personal and Mobile Health Devices. Invited Talk: Workshop to Set A Research Agenda for Privacy and Security of Healthcare Technologies, October 2009. Indianapolis, IN.
53. Considerations for EAS Over Cellular Text Messaging Services. 3G Americas Webinar, July 2009.

54. University Telephony Research Panel. Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM), July 2009.
55. The Evolving Mobile Landscape: Emerging Security Threats. Mobile Security eConference, SC Magazine, June 2008.
56. Characterizing the Impact of Rigidity on the Security of Cellular Networks. University of Washington, February 2009. Seattle, WA.
57. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Microsoft Research, February 2009. Redmond, WA.
58. Next Year's Problems. Secure Computing (SC) Magazine Webinar, November 2008.
59. Panel: Embedded Systems and their Increasing Impact on Infrastructure Security. Workshop on Embedded Systems Security (WESS), October 2008.
60. Can you DoS me now? Security Issues in Cellular Networks. Georgia Institute of Technology, September 2008. Atlanta, GA.
61. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Georgia Institute of Technology, April 2008. Atlanta, GA.
62. Characterizing the Impact of Rigidity on the Security of Cellular Networks. AT&T Research Labs, April 2008. Florham Park, NJ.
63. Characterizing the Impact of Rigidity on the Security of Cellular Networks. University of Arizona, March 2008. Tucson, AZ.
64. Cellular Networks Security Panel. USENIX Security Symposium, August 2007. Boston, MA.
65. malnets:Large-Scale Malicious Networks via Compromised Access Points. The Pennsylvania State University - ACM Club Invited Speaker, October 2006. State College, PA.
66. malnets:Large-Scale Malicious Networks via Compromised Access Points. The University of Michigan, October 2006. Ann Arbor, MI.
67. Exploiting Open Functionality in SMS-Capable Cellular Networks. The University of Michigan, October 2006. Ann Arbor, MI.
68. Exploiting Open Functionality in SMS-Capable Cellular Networks. High Technology Crime Investigation Association (HTCIA), September 2006. Pittsburgh, PA.
69. Trends in Security: Critical Engineering in the Large. Schlumberger Innovate IT! Workshop, May 2006. Cambridge, MA.
70. Exploiting Open Functionality in SMS-Capable Cellular Networks. InfraGard Pittsburgh Chapter General Meeting, September 2006. Pittsburgh, PA.
71. Exploiting Open Functionality in SMS-Capable Cellular Networks. InfraGard Pittsburgh Chapter General Meeting, September 2006. Pittsburgh, PA.

B. Special Activities

Presentations to Lay Media

1. How technology can fight digital fakery. The Babbage Podcast/The Economist <https://shows.acast.com/theeconomistbabbage/episodes/babbage-how-to-detect-a-deepfake>, January 2023.
2. Deepfake audio has a tell and researchers can spot it. Ars Technica <https://arstechnica.com/information-technology/2022/09/researchers-use-fluid-dynamics-to-spot-deepfake-voices/>, September 2022.
3. This security tool could help stop the problem of ransomware in its tracks. TheJournal.ie <https://www.thejournal.ie/ransomware-researchers-stop-2875032-Jul2016/>, July 2016.
4. Researchers Unleash Ransomware Annihilation. BankInfoSecurity - <http://www.bankinfosecurity.com/researchers-unleash-ransomware-annihilation-a-9255>, July 2016.
5. CryptoDrop Stops Ransomware by Stopping its Encryption. Security Intelligence - https://securityintelligence.com/news/cryptodrop-stops-ransomware-by-stopping-its-encryption/?utm_source=tfeed&utm_medium=twitter, July 2016.
6. Ransomware 'stopped' by new software. BBC - <http://www.bbc.com/news/technology-36772461>, July 2016.
7. Researchers create effective anti-ransomware solution. Help Net Security - <https://www.helpnetsecurity.com/2016/07/12/anti-ransomware-solution/>, July 2016.
8. Florida U boffins think they've defeated all ransomware. http://www.theregister.co.uk/2016/07/12/ransomware_defeated/, July 2016.
9. This Anti-Ransomware Tool Could Save You Hundreds of Pounds. Huffington Post - http://www.huffingtonpost.co.uk/entry/anti-ransomware-tool-save-hundreds-pounds_uk_57838beee4b0935d4b4b30ba, July 2016.
10. Researchers develop method to stop 100% of ransomware before it encrypts all files. Myce - <http://www.myce.com/news/researchers-develop-method-stop-100-ransomware-encrypts-files-79873/>, July 2016.
11. Desarrollan una solución para detener el ransomware. ComputerHoy - <http://computerhoy.com/noticias/software/desarrollan-solucion-detener-ransomware-47972>, July 2016.
12. Why your antivirus software can't stop ransomware. Futurity - <http://www.futurity.org/ransomware-computer-files-1198242-2/>, July 2016.
13. CryptoDrop Gives Users Hope to Prevent Ransomware Infections in the Future. Softpedia - <http://news.softpedia.com/news/cryptodrop-gives-users-hope-to-prevent-ransomware-infections-in-the-future-506187.shtml>, July 2016.

14. Could this be the answer to the ransomware threat?, Consumer Affairs. Consumer Affairs - <https://www.consumeraffairs.com/news/could-this-be-the-answer-to-the-ransomware-threat-071116.html>, July 2016.
15. Extortion extinction: Researchers develop a way to stop ransomware. Phys.org - <http://phys.org/news/2016-07-extortion-extinction-ransomware.html>, July 2016.
16. Researchers Develop A Way To Stop Ransomware By Watching The Filesystem. Slashdot - <https://yro.slashdot.org/story/16/07/08/2242244/researchers-develop-a-way-to-stop-ransomware-by-watching-the-filesystem>, July 2016.
17. Mohul Ghosh. Trak.in - Digital Money Apps In India Are Unsafe and Unsecured - Researchers. <http://trak.in/tags/business/2015/08/17/digital-money-apps-india-unsafe-unsecured/>, August 2015.
18. Richard Handford. Mobile World Live - Survey finds security holes in mobile money apps. <http://www.mobileworldlive.com/money/news-money/survey-finds-security-holes-in-mobile-money-apps/#.Vc27Y-QTmSQ.twitter>, August 2015.
19. JENNIFER VALENTINO-DEVRIES. Wall Street Journal - Researchers Find Security Flaws in Developing-World Money Apps. <http://blogs.wsj.com/digits/2015/08/11/researchers-find-security-flaws-in-developing-world-money-apps/>, August 2015.
20. Jonathon Cheng. Wall Street Journal - Samsung Phone Studied for Possible Security Gap. <http://online.wsj.com/news/articles/SB10001424052702304244904579276191788427198>, December 2013.
21. N. V. The Economist - The Threat in the Pocket. <http://www.economist.com/blogs/babbage/2013/10/difference-engine-0?fsrc=scn/fb/wl/bl/thethreatinthepocket>, October 2013.
22. Antone Gonsalves. ComputerWorld - Let's Dump Anti-Virus and Move On:. <http://blogs.computerworld.com/mobile-security/22969/lets-dump-av-and-move>, October 2013.
23. Mathew J. Schwartz. InformationWeek - Google: Don't Fear Android Malware. <http://www.informationweek.com/security/mobile/google-dont-fear-android-malware/240162399>, October 2013.
24. Kirsten Doyle. ITWeb - Android Threat Exaggerated, or is it? http://www.itweb.co.za/index.php?option=com_content&view=article&id=68055, October 2013.
25. Danielle Walker. SC Magazine - Mobile malware prevalence expands, but privacy-abusing apps should be top of mind. <http://www.scmagazine.com/mobile-malware-prevalence-expands-but-privacy-abusing-apps-should-be-top-of-mind/article/300597/>, June 2013.
26. Jim Burrell. WABE NPR - Mobile Web Browsers Full of Security Risks, Tech Professor Finds. <http://wabe.org/post/mobile-web-browsers-full-security-risks-tech-professor-finds>, December 2012.

27. Mark Huffman. Consumer Affairs - Georgia Tech: mobile browsers fail safety test. <http://www.consumeraffairs.com/news/georgia-tech-mobile-browsers-fail-safety-test-120612.html>, December 2012.
28. Matthew J. Schwartz. Information Week - Blame Screen Size: Mobile Browsers Flunk Security Tests. <http://www.informationweek.com/security/mobile/blame-screen-size-mobile-browsers-flunk/240143999>, December 2012.
29. Jon Gold. Network World - Ga. Tech researchers: Mobile Browsers need better HTTPS indicators. <http://www.networkworld.com/news/2012/120512-mobile-browsers-264846.html>, December 2012.
30. United Press International. Study: Most mobile Web browsers unsafe. http://www.upi.com/Science_News/Technology/2012/12/05/Study-Most-mobile-Web-browsers-unsafe/UPI-73431354743353/#ixzz2EGtQsuLd, December 2012.
31. Suzanne Choney. Mobile browser woes can fool even experts: report. <http://www.nbcnews.com/technology/mobile-browser-woes-can-fool-even-experts-report-1C7451203>, December 2012.
32. Meghan Kelly. VentureBeat - 3 hot security startups to watch. <http://venturebeat.com/2012/02/27/3-security-startups-to-watch-at-the-2012-rsa-conference/>, February 2012.
33. Jacob Goodwin. Government Security News - RSA 2012 – Pindrop Security can distinguish a fraudulent phone call from a real one. <http://www.gsnmagazine.com/node/25721?c=communications>, February 2012.
34. Matt Liebowitz. Phone hack logs keystrokes from nearby computers. MSNBC.com - http://www.msnbc.msn.com/id/44993238/ns/technology_and_science-security/#.TqU5MNSjPh4, October 2011.
35. Jacob Aron. iPhone keylogger can snoop on desktop typing. New Scientist - <http://www.newscientist.com/article/dn21059-iphone-keylogger-can-snoop-on-desktop-typing.html>, October 2011.
36. iPhone Keylogger Can Snoop on Desktop Typing. Slashdot - <http://mobile.slashdot.org/story/11/10/18/2346222/iphone-keylogger-can-snoop-on-desktop-typing>, October 2011.
37. Robert Lemos. Smart Phones Could Hear Your Password. Technology Review - <http://www.technologyreview.com/computing/38913/?p1=A2>, October 2011.
38. Kevin McCaney. Bad vibrations: How smart phones could steal PC passwords. Government Computer News - <http://gcn.com/articles/2011/10/18/smart-phone-sensors-steal-keystrokes.aspx>, October 2011.
39. PhysOrg. Turning iPhone into spiPhone: Smartphones' accelerometer can track strokes on nearby keyboards. PhysOrg.com - <http://www.physorg.com/news/2011-10-iphone-spiphone-smartphones-accelerometer-track.html>, October 2011.
40. Brid-Aine Parnell. Securo-boffins call for 'self-aware' defensive technologies. The Register - http://www.theregister.co.uk/2011/09/14/self_aware_cyber_security_technologies_should_be_a_top_priority/, September 2011.

41. Clay Dillow. 'PinDr0p' Tech Uses Unique Noise Fingerprints to Trace Calls. Popular Science - <http://www.popsci.com/technology/article/2010-10/pindr0p-tech-tags-phone-calls-unique-fingerprints-trace-call-paths-across-networks>, October 2010.
42. Lewis Page. Voice-routing call fingerprint system fights vishing. The Register - http://www.theregister.co.uk/2010/10/06/voice_fingerprints, October 2010.
43. Science Daily. Voice Phishing: System to Trace Telephone Call Paths Across Multiple Networks Developed. <http://www.sciencedaily.com/releases/2010/10/101005121820.htm>, October 2010.
44. Brian Kalish. To Text or Not to Text During Emergencies. NextGov.com - http://www.nextgov.com/nextgov/ng_20100914_5986.php?oref=topnews, September 2010.
45. Ki Mae Heussner. 'Operation Chokehold': Fake Steve Jobs Rallies iPhone Users to Cripple AT&T Network. ABC News - <http://abcnews.go.com/Technology/GadgetGuide/fake-steve-jobs-rallies-iphone-users-cripple-att/story?id=9355447>, December 2009.
46. Bob Brown. Researchers Set Their Sights on iPhones, Mobile Malware. PC World Magazine - http://www.pcworld.com/article/182005/iphone_worms_mobile_malware.html?tk=rss, November 2009.
47. MacGregor Campbell. Botnets show their disruptive potential. New Scientist Magazine - <http://www.newscientist.com/article/mg20427347.000-mobile-botnets-show-their-disruptive-potential.html>, November 2009.
48. Angela Moscaritolo. Remote repair for infected phones in development. SC Magazine - <http://www.scmagazineus.com/remote-repair-for-infected-phones-in-development/article/157504/>, November 2009.
49. Bob Brown. iPhone worms, other smartphone malware in researchers' sights. Network World - <http://www.networkworld.com/news/2009/111109-smartphone-security-georgia-tech.html?hpg1=bn>, November 2009.
50. Urvaksh Karkaria. GT researchers work to secure cellphones. Atlanta Business Chronicle - <http://atlanta.bizjournals.com/atlanta/blog/atlantech/2009/11/cellphone.html>, November 2009.
51. Making Carriers Shoulder Smartphone Security. http://mobile.slashdot.org/story/09/11/11/2318247/Making-Carriers-Shoulder-Smartphone-Security?art_pos=31, November 2009.
52. Ben Meyer. Georgia Tech to Lead Fight Against Cell Phone Hackers. NBC 11 Atlanta - <http://www.11alive.com/news/local/story.aspx?storyid=132505&catid=3>, July 2009.
53. Illena Armstrong. Safeguarding your mobile networks. SC Magazine - <http://www.scmagazineus.com/Safeguarding-your-mobile-networks/article/138289/>, June 2009.
54. Kelli B. Grant. Four Free Cellphone Apps to Help Manage Your Money. SmartMoney Magazine - <http://www.smartmoney.com/Spending/Deals/4-Great-Free-Finance-Apps-for-Cellphones/>, June 2009.
55. Amanda Hoffstrom. Technology's limitations in alerting campus danger. UWire Magazine - <http://www.uwire.com/Article.aspx?id=3738798>, February 2009.

56. Laura Sydell. Compromise Allows Obama To Keep BlackBerry. National Public Radio - <http://www.npr.org/templates/story/story.php?storyId=99790788>, January 2009.
57. Dennis Carter. Questions abound as emergency alert flops Virginia Tech's text-message alert system failed when the sound of gunfire was heard on campus; officials scramble to understand why. eSchool News - http://www.eschoolnews.com/iphone/top-story/index.cfm?i=56122#_56122, November 2008.
58. Jessica Bauer. Study: Text alerts may fail in real emergency. Diamondback Online - <http://media.www.diamondbackonline.com/media/storage/paper873/news/2008/10/14/News/Study.Text.Alerts.May.Fail.In.Real.Emergency-3485509.shtml>, October 2008.
59. Associated Press. Hackers Expected To Start Targeting Cell Phones. <http://cbs5.com/watercooler/Cell.Phones.Hackers.2.840909.html>, 2008.
60. Associated Press. College alert systems unreliable, study says. http://www.ajc.com/search/content/metro/stories/2008/09/25/college_campus_alerts.html, 2008.
61. Lee Shearer. Study: Campus alerts unreliable. Athens Banner Herald http://www.onlineathens.com/stories/092508/uga_336494829.shtml, 2008.
62. Bill Ray. 3G Americas warns against text warning systems. The Register - http://www.theregister.co.uk/2008/09/18/emergency_text/, 2008.
63. 3G Americas. 3G Americas Highlights New Research Report on Use of Cellular Text Messaging for Emergency Alert Services. 3G Americas http://www.3gamericas.org/English/news_room/DisplayPressRelease.cfm?id=3400&s=ENG, 2008.
64. Evan Koblentz. Web Exclusive: From Messaging to Management Duty. Wireless Week - <http://www.wirelessweek.com/Messaging-to-Management-Duty.aspx>, 2008.
65. Christopher Beam. How Do You Intercept a Text Message? Turn your cell phone into a spy gadget. Slate Magazine <http://www.slate.com/id/2161402/>, 2007.
66. Jamming Cellphones with Text Messages. Slashdot <http://it.slashdot.org/it/05/10/05/1839217.shtml?tid=215&tid=172>, 2005.
67. Cell phone networks at risk? CNN http://money.cnn.com/2005/10/05/technology/hacker_cellphones/, 2005.
68. John Schwartz. Text Hackers Could Jam Cellphones, a Paper Says. The New York Times <http://www.nytimes.com/2005/10/05/technology/05phone.html?ex=1286164800&en=d917b9cd43dfaa31&ei=5090&partner=rssuserland&emc=rss>, 2005.