

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Gregory G. Raleigh Attorney Docket No. 39843-0190IP1
U.S. Patent No.: 10,028,144
Issue Date: July 17, 2018
Appl. Serial No.: 14/948,065
Filing Date: November 20, 2015
Title: SECURITY TECHNIQUES FOR DEVICE ASSISTED SERVICES

DECLARATION OF DR. PATRICK MCDANIEL

I declare that all statements made herein on my own knowledge are true and that all statements made on information and belief are believed to be true, and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable under Section 1001 of Title 18 of the United States Code.

Date: May 5, 2025

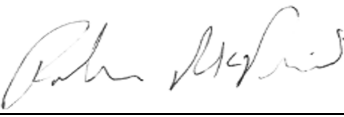
By: 
Patrick McDaniel, Ph.D.

Table of Contents

I.	ASSIGNMENT.....	9
II.	QUALIFICATIONS AND BACKGROUND INFORMATION.....	9
III.	OVERVIEW OF CONCLUSIONS FORMED	20
IV.	LEVEL OF ORDINARY SKILL IN THE ART.....	21
V.	LEGAL STANDARDS	22
	A. Terminology.....	22
	B. Legal Standards.....	23
	C. Anticipation.....	23
	D. Obviousness	23
VI.	MATERIALS CONSIDERED.....	28
VII.	TECHNOLOGY OVERVIEW.....	29
	A. Partitioning of Computing Environments.....	29
	1. Hardware Partitions.....	30
	2. Software Partitions	31
	3. Memory Partitions.....	32
	B. Policy Settings and Enforcement.....	34
	1. Network Traffic Policies	35
	2. Application-Specific Policies.....	37
	C. Redundancies in Computing Systems	38
VIII.	Overview of the '144 patent.....	39
IX.	Prosecution History of the '144 patent.....	42
X.	OVERVIEW AND COMBINATIONS OF PRIOR ART REFERENCES ..	43
	A. U.S. Pat. App. Pub. No. 2008/0052395 A1 (“Wright”)	43
	B. U.S. Pat. App. Pub. No. 2009/0300450 A1 (“Tzannes”)	48
	C. Wright-Tzannes Combination	51
	D. Smith, <i>Virtual Machines: Versatile Platforms for Systems and Processes</i> (“Smith”).....	58
	E. Wright-Smith Combination	60
XI.	MANNER IN WHICH THE PRIOR ART REFERENCES RENDER THE '144 CLAIMS UNPATENTABLE	64
	A. Ground 1A: Wright Renders Obvious Claims 1-11, 13, 15, and 18	64
	B. Ground 1B: Combination of Wright and Tzannes Renders Obvious Claims 12, 16, and 17	172
	C. Ground 1C: Wright in view of Smith renders claims 14 obvious	179

XII. CONCLUSION.....	179
----------------------	-----

CLAIMS

Claim	Identifier	Claim Language
1	[1.1]	A wireless end-user device comprising:
	[1.2]	a wireless wide area network (WWAN) modem to receive and transmit Internet data between the device and at least one WWAN when configured for and connected to the at least one WWAN; and
	[1.3]	one or more processors configured to execute one or more processes in a kernel execution partition, one or more processes in an application execution partition, and one or more processes in a protected execution partition,
	[1.4]	the one or more processes executed in the kernel execution partition including an operating system packet network stack to pass Internet data packet traffic between the WWAN modem and one or more applications executing in the application execution partition,
	[1.5]	the one or more processes executed in the kernel execution partition including ... an application identification agent to classify individual flows of the Internet data packet traffic passing through the stack according to which of the one or more applications is associated with a particular individual flow,
	[1.6]	the one or more processes executed in the kernel execution partition including ... a service measurement agent to measure, for classified individual flows, an amount of the Internet data packet traffic associated with each of the one or more applications, and
	[1.7]	the one or more processes executed in the kernel execution partition including ... a policy control agent to apply application-specific traffic policy controls to classified individual flows,
	[1.8]	the one or more processes executed in the protected execution partition including one or more device agents with limited privileges to access processes executing in the kernel execution partition, including

Claim	Identifier	Claim Language
	[1.9]	one or more device agents ... including a privilege to configure the application-specific traffic policy controls of the policy control agent, and
	[1.10]	one or more device agents ... including ... a privilege to receive at least one of Internet data packet traffic passing between the operating system packet network stack and the WWAN modem, and traffic information including the measured amount of the Internet data packet traffic associated with each of the one or more applications.
2	[2]	The wireless end-user device of claim 1, the one or more processors further configured to execute, in the application execution partition, at least one device-assisted-services agent to communicate with at least one device agent executed in the protected execution partition.
3	[3.1]	The wireless end-user device of claim 1, further comprising one or more memory partitions to store program code for the respective processes of the operating system packet network stack, the application identification agent, the service measurement agent, the policy control agent, and the one or more device agents, the one or more processors configured to retrieve the stored program code from the one or more memory partitions to execute the respective processes.
	[3.2]	The wireless end-user device of claim 1, further comprising one or more memory partitions to store program code for the respective processes of the operating system packet network stack, the application identification agent, the service measurement agent, the policy control agent, and the one or more device agents, the one or more processors configured to retrieve the

Claim	Identifier	Claim Language
		stored program code from the one or more memory partitions to execute the respective processes.
4	[4.1]	The wireless end-user device of claim 3, wherein the one or more memory partitions include a separately protected device-assisted services partition to store the program code for the one or more device agents,
	[4.2]	the one or more device agents further comprising an access control integrity agent having exclusive permission to update program code stored in the separately protected device-assisted services partition.
5	[5]	The wireless end-user device of claim 4, the access control integrity agent further configured to validate the program code for the one or more device agents.
6	[6]	The wireless end-user device of claim 3, wherein the one or more memory partitions comprise a device-assisted services partition to store device service state information, the device service state information only modifiable by the one or more device agents.
7	[7]	The wireless end-user device of claim 6, the one or more device agents further comprising a service control link configured to securely communicate with a network service controller.
8	[8]	The wireless end-user device of claim 7, wherein the service control link is further configured to receive at least one of the application-specific traffic policy controls from the network service controller.
9	[9]	The wireless end-user device of claim 7, wherein the service control link is further configured to transmit at least some of the device service state information to the network service controller.

Claim	Identifier	Claim Language
10	[10]	The wireless end-user device of claim 9, wherein the one or more device agents are further configured to retain the transmitted device service state information until after an instruction is received from the network service controller acknowledging that the device service state information can be deleted.
11	[11]	The wireless end-user device of claim 1, the one or more device agents further configured to, based on the application-specific traffic policy controls, configure application settings for the one or more applications.
12	[12]	The wireless end-user device of claim 1, the one or more device agents further configured to calculate, separate from the traffic information, a service usage measurement for the Internet data packet traffic passing between the operating system packet network stack and the WWAN modem.
13	[13]	The wireless end-user device of claim 1, the one or more device agents further configured to encrypt Internet data packet traffic passing from the operating system packet network stack to the WWAN modem.
14	[14]	The wireless end-user device of claim 1, wherein the protected execution partition comprises a secure virtual execution environment.
15	[15]	The wireless end-user device of claim 1, wherein the privilege to configure the application-specific traffic policy controls of the policy control agent is an exclusive privilege for one or more of the device agents executed in the protected execution partition.
16	[16]	The wireless end-user device of claim 1, the WWAN modem comprising a modem execution partition configured to calculate, separate from the traffic information, a service usage measurement for the Internet data packet traffic passing between the operating system packet network stack and the WWAN modem.

Claim	Identifier	Claim Language
17	[17]	The wireless end-user device of claim 16, the modem execution partition further comprising a modem control link to communicate the calculated service usage measure to at least one of a network service controller and at least one device agent executed in the protected execution partition.
18	[18]	The wireless end-user device of claim 2, further comprising a user interface, the at least one device-assisted-services agent comprising a user-interface agent to display to a user, and allow modification to, at least one policy setting affecting at least one of the application-specific policy controls.

I. ASSIGNMENT

1. I have been retained on behalf of Samsung Electronics Co., Ltd. (“Samsung”) to offer technical opinions related to 10,028,144 (“the ’144 patent”) (SAMSUNG-1001). I understand that Samsung is requesting the Patent Trial and Appeal Board (“PTAB” or “Board”) to institute an *inter partes* review (“IPR”) proceeding of the ’144 patent.

2. I have been asked to provide my independent analysis of the ’144 patent based on the prior art publications cited in this declaration.

3. I am not and never have been, an employee of Samsung. I received no compensation for this declaration beyond my normal hourly compensation based on my time actually spent analyzing the ’144 patent, the prior art publications and materials cited below, and issues related thereto, and I will not receive any added compensation based on the outcome of any IPR or other proceeding involving the ’144 patent.

II. QUALIFICATIONS AND BACKGROUND INFORMATION

4. I am over the age of 18 and am competent to write this declaration. I have personal knowledge, or have developed knowledge of these technologies based upon education, training, or experience, of the matters set forth herein.

5. My qualifications are summarized here and explained in more detail in my curriculum vitae, which is attached as Appendix A to this report. Appendix A also includes a list of my publications and the cases in which I have testified at deposition, hearing, or trial within the past four years.

6. I earned a Ph.D. in Computer Science from the Computer Science and Engineering Department at University of Michigan, Ann Arbor in 2001. I earned a Bachelor of Science degree in Computer Science from Ohio University in 1989, and a Master of Science degree, also in Computer Science, from Ball State University in 1991.

7. Since 2022, I have been Tsun-Ming Shih Professor of Computer Sciences in the School of Computer, Data and Information Sciences at the University of Wisconsin-Madison. Prior to my move to Wisconsin, I was the William L. Weiss Professor of Information and Communications Technology in the School of Electrical Engineering and Computer Science at Pennsylvania State University in University Park, PA. I was also the director of the Institute for Network and Security Research, and founder and co-director of the Systems and Internet Infrastructure Security Laboratory, a research laboratory focused on the study of security in diverse network and computer environments. My research efforts primarily involve computer systems, network management and authentication, systems security, and technical public policy.

8. Prior to 2017, I was an Assistant Professor (2004-2007), Associate Professor (2007-2011), Full Professor (2011-2015), and Distinguished Professor (2016-2017) of Computer Science and Engineering and the William L. Weiss Professor of Information and Communications Technology (2017-2022) at the Pennsylvania State University. Since 2004, I have taught courses in the field of computer systems, systems programming, mobile device security, networks, and network and computer security at both the undergraduate and graduate level. I created and maintained several of these courses at Penn State. I began teaching cybersecurity courses at the University of Wisconsin-Madison but will teach a network security course in Fall of 2023.

9. From 2003-2009, I was also an Adjunct Professor at the Stern School of Business at New York University in New York, NY. At the Stern School of Business, I taught courses in computer and network security and online privacy.

10. I am a Fellow of the Association for Computing Machinery (ACM, the leading professional association for computer science), a Fellow Institute for Electrical and Electronics Engineering (IEEE, the leading professional association for engineering), and Fellow of the American Association for the Advancement of Science (AAAS, the leading professional association for science).

11. I am the director of the NSF Frontier Center for Trustworthy Machine Learning (CTML), a research organization exploring security and privacy of

machine learning (and AI in general). Funded by the US National Science Foundation, the CTML is led by the University of Wisconsin-Madison and has members from Stanford, the University of California, Berkeley, The University of California-San Diego, and the University of Virginia.

12. I was the Program Manager (PM) and lead scientist for the Cyber Security Collaborative Research Alliance (CRA). The CRA is led by Penn State University and includes faculty and researchers from the Army Research Laboratory, Carnegie Mellon University, Indiana University, the University of California-Davis, and the University of California-Riverside. This initiative is a major research project aimed at developing a new science of cyber-security for military networks, computers, and installations.

13. I have served as an advisor to several Ph.D. and master's degree candidates, several of whom have gone on to become professors at various institutions such as North Carolina State University, Purdue University, the University of Toronto, the University of Oregon, and the Georgia Institute of Technology and the University of Florida. I am currently an advisor to seven Ph.D. students.

14. Before joining Pennsylvania State University as a professor, I was a software developer and project manager for companies in the networking industry including Applied Innovation, Inc., and Primary Access Corporation. I was also a

senior researcher at AT&T Research-Labs (previously known as Bell Labs). As part of my duties in these industrial positions, I informed, reviewed, and formed corporate policies and practices relating to the security of mobile devices, networks, and software systems.

15. As shown in Appendix A, I have published extensively in the field of mobile security, network and security management, computer systems, routing and queueing, authentication, systems security, applied cryptography and network security. In addition to writing several articles for industry journals and conferences, I have authored portions of numerous books related to computer systems, applied cryptography and network security. I have served on the editorial boards of several peer-reviewed journals including ACM Transactions on Internet Technology, for which I was the Editor-in-Chief. I was also an Associate Editor for ACM Transactions on Information and System Security and IEEE Transactions of Software Engineering, two highly regarded journals in the field. My curriculum vitae, attached as Appendix A, includes a list of publications on which I am an author. It contains details regarding my experience, education, publications, and other qualifications to provide an expert opinion in connection with this proceeding.

16. My curriculum vitae, included as an appendix to 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.

17. In writing this Declaration, I have considered the following: my own knowledge and experience, including my work experience in the fields of networks, network management and network policy; 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:

18. I have reviewed the '144 patent (SAMSUNG-1001), and relevant excerpts of the prosecution history (SAMSUNG-1002) of the '144 patent. Among various textbooks, documents, and publications, I have also reviewed the following prior art references:

SAMSUNG-1001	U.S. Patent No. 10,028,144 to Gregory G. Raleigh (“the '144 Patent”)
SAMSUNG-1002	Excerpts from the Prosecution History of the '144 Patent (“Prosecution History”)
SAMSUNG-1005	U.S. Patent Application No 2008/0052395 A1 (“Wright”)
SAMSUNG-1006	U.S. Patent Application No. 2009/0300450 A1 (“Tzannes”)

SAMSUNG-1007	(Excerpts) Smith, et al., “Virtual Machines: Versatile Platforms for Systems and Processes,” Elsevier, Inc (2005)
SAMSUNG-1009	U.S. Patent Application No. 2005/0144505 A1 (“Takeuchi”)
SAMSUNG-1010	U.S. Patent Application No. US20090254465 A1 (“Oester”)
SAMSUNG-1011	U.S. Patent Application No. 2006/0015749 A1 (“Mittal”)
SAMSUNG-1012	Declaration of June Munford
SAMSUNG-1013	Mallick, M., Mobile and Wireless Design Essentials (2003)
SAMSUNG-1014	Anderson, R., Security Engineering: A Guide to Building Dependable Distributed Systems. (2001)
SAMSUNG-1015	Jaeger, T., Operating System Security (Synthesis Lectures on Information Security, Privacy, and Trust) (2008)

SAMSUNG-1016	Parziale, L., TCP/IP Tutorial and Technical Overview (IBM Redbooks) (2006)
SAMSUNG-1017	McCanne, S. et al., The BSD Packet Filter: A New Architecture for User-level Packet Capture (1992)
SAMSUNG-1018	Horak, R., <i>Webster's New World Telecom Dictionary</i> , (2008)
SAMSUNG-1019	Ganz, A., et al., Multimedia Wireless Networks: Technologies, Standards, and QoS (2003)
SAMSUNG-1020	Kwok, Y., et al., Wireless Internet and Mobile Computing: Interoperability and Performance (2007)
SAMSUNG-1021	Brownlee et al., Traffic Flow Measurement: Architecture (1997)
SAMSUNG-1022	Saltzer, J. H., The Protection of Information in Computer Systems, Proceedings of the IEEE (1975)
SAMSUNG-1023	Tan, et al., An Empirical Study on the Capacity and Performance of 3G Networks (2008)

SAMSUNG-1024	Wang, et al., Shield: Vulnerability-Driven Network Filters for Preventing Known Vulnerability Exploits, ACM SIGCOMM Computer Communication Review (2004)
SAMSUNG-1025	Silberschatz et al., Operating System Concepts (7th ed. 2005)
SAMSUNG-1026	Cheswick et al., Firewalls and Internet Security (2nd ed. 2003)
SAMSUNG-1027	Koren et al., Fault Tolerant Systems (2007)
SAMSUNG-1028	Loscri et al., Voice and Video Telephony Services in Smartphone, EURASIP Journal on Wireless Communications and Networking (2006)
SAMSUNG-1029	Oxford University Press, <i>A Dictionary of Computing</i> , (6th ed. 2008)
SAMSUNG-1030	Downing et al., <i>Dictionary of Computer and Internet Terms</i> (9th ed. 2006)
SAMSUNG-1031	Kunie et al., “Low power architecture and design techniques for mobile handset LSI Medity M2,” 748-753,

	2008 Asia and South Pacific Design Automation Conference (2008).
SAMSUNG-1032	M. Naruse et al., “A 65nm Single-Chip Application and Dual-Mode Baseband Processor with Partial Clock Activation and IP-MMU,” 260-612, 2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers (2008).
SAMSUNG-1033	Gertz et al., Handbook of Database Security: Applications and Trends, Springer (2007).
SAMSUNG-1034	Oney, Programming The Microsoft Windows Driver Model (2nd ed. 2003).
SAMSUNG-1035	Exhibit A, Headwater Research LLC v. Samsung Electronics Co., Ltd. et al, 2-24-cv-00627, E.D. Tex., filed August 2, 2024.
SAMSUNG-1040	IEEE 100, <i>The Authoritative Dictionary of IEEE Standards Terms</i> (7th ed. 2000)
SAMSUNG-1041	U.S. Patent No. 7,962,798 to Locasto et al. (“Locasto”)
SAMSUNG-1042	Tanenbaum et al., Modern Operating Systems (2001)

SAMSUNG-1043	U.S. Patent Application No. 2007/0143827 A1 (“Nicodemus”)
SAMSUNG-1044	U.S. Patent Application No. 2007/0006282 A1 (“Durham”)
SAMSUNG-1045	U.S. Patent No. 7,633,909 B1 (“Jones”)
SAMSUNG-1046	U.S. Pat. No. 9,032,192B2 (“Frank”)
SAMSUNG-1047	The Secure Sockets Layer (“SSL”) Protocol, V. 3.0, available at <i>https://web.archive.org/web/19970614041044/http://home.netscape.com/eng/ssl3/ssl-toc.html</i> and <i>https://web.archive.org/web/19970617034012/http://home.netscape.com/eng/ssl3/3-SPEC.HTM#1</i>

19. Counsel (Fish & Richardson) has informed me that I should consider these materials through the lens of one of ordinary skill in the art related to the ’144 patent at the time of the earliest possible priority date of the ’144 patent, and I have done so during my review of these materials. I have been informed by Counsel to use the effective filing date of January 28, 2009.

20. I have no financial interest in the outcome of this proceeding. I am being compensated for my work as an expert on an hourly basis. My compensation is not dependent on the outcome of these proceedings or the content of my opinions.

21. My opinions, as explained below, are based on my education, experience, and expertise in the fields relating to the '144 patent. Unless otherwise stated, my testimony below refers to the knowledge of one of ordinary skill in the art as of the earliest possible priority date. Any figures that appear within this document have been prepared with the assistance of Counsel and reflect my understanding of the '144 patent and the prior art discussed below.

III. OVERVIEW OF CONCLUSIONS FORMED

22. This declaration explains the conclusions that I have formed based on my analysis of all of the considered materials and my personal experience. To summarize those conclusions, based upon my knowledge and experience and my review of the prior art references listed above, I believe that:

- Ground 1A renders obvious claims 1-11, 13, 15, and 18.
- Ground 1B renders obvious claims 12, 16, and 17.
- Ground 1C renders obvious claim 14.

23. In support of these conclusions, I provide an overview of the references in Section X and more detailed comments regarding the obviousness of claims 1-18 (“**Challenged Claims**”) of the ’144 patent in Section XI.

IV. LEVEL OF ORDINARY SKILL IN THE ART

24. In my opinion, one of ordinary skill in the art relating to, and at the time of, the Priority Date would have been someone with (1) at least a bachelor’s degree in computer science, computer engineering, electrical engineering, or a related field, and (2) at least two years of industry experience in wireless device security and wireless device communications security. Additional education could substitute for industry experience, and vice versa. *Id.* Further, all positions in this declaration are from the vantage point of a POSITA as of January 28, 2009.

25. Based on my experiences, I have a good understanding of the capabilities of a POSITA. Indeed, I have taught, participated in organizations, and worked closely with many such persons over the course of my career. Based on my knowledge, skill, and experience, I have an understanding of the capabilities of a POSITA. For example, from my industry experience, I am familiar with what a POSITA would have known and found predictable in the art. From teaching and supervising my post-graduate students, I also have an understanding of the

knowledge that a person with this academic experience possesses. Furthermore, I possess those capabilities myself.

V. LEGAL STANDARDS

A. Terminology

26. 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. 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 at the time of the invention was made (not today). The '144 patent was filed November 20, 2015, and claims priority to many provisional applications, the earliest of which was filed January 28, 2009, which I refer to as the "Critical Date" in this proceeding. Because I do not know at what date the invention as claimed was made, if ever, I have used the Critical Date of the '144 patent as the point in time for claim interpretation purposes. My opinion does not change if the invention date is earlier.

B. Legal Standards

27. I have been informed by Counsel and understand that documents and materials that qualify as prior art can render a patent claim unpatentable as being anticipated or obvious.

28. I am informed by Counsel and understand that all prior art references are to be looked at from the viewpoint of a person of ordinary skill in the art at the time of the invention, and that this viewpoint prevents one from using his or her own insight or hindsight in deciding whether a claim is anticipated or rendered obvious.

C. Anticipation

29. I understand that patents or printed publications that qualify as prior art can be used to invalidate a patent claim as anticipated or as obvious.

30. I understand that, once the claims of a patent have been properly construed, the second step in determining anticipation of a patent claim requires a comparison of the properly construed claim language to the prior art on a limitation-by-limitation basis.

31. I understand that a prior art reference “anticipates” an asserted claim, and thus renders the claim invalid, if all limitations of the claim are disclosed in that prior art reference, either explicitly or inherently (i.e., necessarily present).

D. Obviousness

32. I understand that even if a patent is not anticipated, it is still invalid if the differences between the claimed subject matter and the prior art are such that the

subject matter as a whole would have been obvious at the time the invention was made to a POSITA.

33. I have been informed by Counsel and understand that a claim is unpatentable for obviousness and that obviousness may be based upon a combination of references. I am informed by Counsel and understand that the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. However, I am informed by Counsel and understand that a patent claim composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art.

34. I am informed by Counsel and understand that when a patented invention is a combination of known elements, a court determines whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue by considering the teachings of prior art references, the effects of demands known to people working in the field or present in the marketplace, and the background knowledge possessed by a person having ordinary skill in the art.

35. I am informed by Counsel and understand that a patent claim composed of several limitations is not proved obvious merely by demonstrating that each of its limitations was independently known in the prior art. I am informed by Counsel and understand that identifying a reason those elements would be combined can be

important because inventions in many instances rely upon building blocks long since uncovered, and claimed discoveries almost of necessity will be combinations of what, in some sense, is already known. I am informed by Counsel and understand that it is improper to use hindsight in an obviousness analysis, and that a patent's claims should not be used as a "roadmap."

36. I am informed by Counsel and understand that an obviousness inquiry requires consideration of the following factors: (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.

37. I have been informed by Counsel and understand that an obviousness evaluation can be based on a single reference or a combination of multiple prior art references. I understand that the prior art references themselves may provide a suggestion, motivation, or reason to combine, but that the nexus linking two or more prior art references is sometimes simple common sense. I have been informed by Counsel and understand that obviousness analysis recognizes that market demand, rather than scientific literature, often drives innovation, and that a motivation to combine references may be supplied by the direction of the marketplace.

38. I have been informed by Counsel and understand that if a technique has been used to improve one device, and a person of ordinary skill at the time of invention would have recognized that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill.

39. I have been informed by Counsel and understand that practical and common sense considerations should guide a proper obviousness analysis, because familiar items may have obvious uses beyond their primary purposes. I have been informed by Counsel and understand that a person of ordinary skill looking to overcome a problem will often be able to fit together the teachings of multiple prior art references. I have been informed by Counsel and understand that obviousness analysis therefore takes into account the inferences and creative steps that a person of ordinary skill would have employed at the time of invention.

40. I have been informed by Counsel and understand that a proper obviousness analysis focuses on what was known or obvious to a person of ordinary skill at the time of invention, not just the patentee. Accordingly, I understand that any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed.

41. I have been informed by Counsel and understand that a claim can be obvious in light of a single reference, without the need to combine references, if the elements of the claim that are not found explicitly or inherently in the reference can be supplied by the common sense of one of skill in the art.

42. I have been informed by Counsel and understand that there must be a relationship between any such secondary considerations and the invention, and that contemporaneous and independent invention by others is a secondary consideration supporting an obviousness determination.

43. In sum, my understanding is that prior art teachings are properly combined where one of ordinary skill having the understanding and knowledge reflected in the prior art and motivated by the general problem facing the inventor, would have been led to make the combination of elements recited in the claims. Under this analysis, the prior art references themselves, or any need or problem known in the field of endeavor at the time of the invention, can provide a reason for combining the elements of multiple prior art references in the claimed manner.

44. I have been informed by Counsel and understand that in an *inter partes* review, “the petitioner shall have the burden of proving a proposition of unpatentability,” including a proposition of obviousness, “by a preponderance of the evidence.” 35 U.S.C. §316(e).

VI. MATERIALS CONSIDERED

45. My analysis and conclusions set forth in this declaration are based on my educational background and experiences in the field (see Section II). Based on my knowledge and experience, I believe that I am considered to be an expert in the field. Also, based on my knowledge and experience, I understand and know of the capabilities of persons of ordinary skill in the field around 2009, and I taught, participated in organizations, and worked closely with many such persons in the field during that time frame.

46. As part of my independent analysis for this declaration, I have considered the following: the background knowledge/technologies that were commonly known to persons of ordinary skill in this art during the time before the earliest claimed priority date for the '144 patent; my own knowledge and experiences gained from my work experience in the field of the '144 patent and related disciplines; and my experience in working with others involved in this field and related disciplines.

47. In addition, I have analyzed the publications and materials listed above.

48. My analysis and conclusions set forth in this declaration are based on the perspective of a POSITA.

VII. TECHNOLOGY OVERVIEW

49. In this section, I provide a broad overview of the relevant technology and state of the technical community before the priority date of the '144 patent. This description is not intended to be exhaustive, but to provide sufficient background to support and contextualize my analysis below.

A. Partitioning of Computing Environments

50. Modern computing environments increasingly use partitioning strategies to provide “better resource utilization, ... effective isolation and security enforcement” and ensure operational stability. SAMSUNG-1015, 154. Partitioning techniques, such as Virtual Machines, allows “software that is used to be run on the same system can run in separate, isolated [partitions]”, therein reducing interference and providing resilience against faults or malicious activities. SAMSUNG-1015, 5. Examples of such partitioned environments include virtualization platforms (such as VMware, which “brought [Virtual Machine] systems to the masses.” SAMSUNG-1015, 164.), container technologies (such as Solar Containers, which “provides an isolated environment for running applications” SAMSUNG-1015, 105.), and I/O memory protection technologies (such as I/O Memory Management Units (MMU) in Intel and AMD processors, which “translates these virtual I/O addresses to physical addresses, authorizing access to the resultant physical address.” SAMSUNG-1015, 82. These technologies allow multiple applications or systems to

coexist within the same physical computing infrastructure while maintaining distinct operational boundaries and varying degrees of privileges and resource access.

1. Hardware Partitions

51. Hardware partitioning refers to the process of physically or logically dividing hardware resources—such as CPUs, GPUs, storage devices, or network interfaces—into isolated domains. Examples of hardware partitioning techniques include Virtual Machine Monitors (which “multiplexes physical system resources among the operating systems in the [Virtual Machines]” (SAMSUNG-1015, 153), physical separation of processors in multiprocessor systems, and specialized hardware security modules (HSMs) or accelerators designed to perform secure computations independently of the main system (such as the Scomp security protection module, which “provides a tamperproof service to mediate all memory and I/O access.” SAMSUNG-1015, 80.

52. One significant technique in hardware partitioning involves the use of hardware virtualization extensions provided by modern CPUs (e.g., Intel VT). These technologies ensure that “the untrusted [Virtual Machines] cannot circumvent the authority of the [Virtual Machine Manager], and its layered system architecture, which improves security via modularity and minimizing interdependencies.” SAMSUNG-1015, 157. Such secure hardware partitions provide the foundational security needed for sensitive computations, cryptographic operations, or trusted

identity management, ensuring these tasks remain inaccessible to less privileged software or external threats.

2. *Software Partitions*

53. Software partitions implement isolation at the operating system (OS) or application level, enabling multiple applications to coexist while enforcing strict boundaries around their privileges and resource access. This is particularly important for enforcing security policies where applications must be assigned and managed at varying levels of privilege depending on their function or sensitivity.

54. Typical examples of varying privilege levels include administrative-level privileges (e.g., root or administrator rights), normal user privileges, and restricted privileges assigned to sandboxed applications. For instance, “when a user wants to update their password, they use an unprivileged process (e.g., a shell) to invoke privileged code (e.g., the password program) to be run with the privileged code’s label (e.g., UNIX root which provides full system access).” SAMSUNG-1015, 19. In this way, administrative privileges may permit modification of critical system configurations, installation of software, and management of hardware resources, whereas user-level privileges are limited to what is defined in an Access Control List (ACL), which contains entries that “specif[y] a user identity of processes and the operations that processes with that identity can perform on this subject.” SAMSUNG-1015, 28.

55. Operating systems implement these privilege distinctions using constructs such as user and kernel space separation, protection rings (ring levels), and process control mechanisms. For example, most modern OS kernels (Linux, Windows, Unix variants) are “isolated from other processes by protection rings.” SAMSUNG-1015, 26. Specifically, “[p]rotection rings form a hierarchical layering from the most privileged ring, ring 0 where the most-privilege code in the [kernel] runs, to the least privileged ring.” SAMSUNG-1015, 26. Practically speaking, protection rings employ a clear separation between kernel space (ring 0)—where the most privileged code runs with unrestricted hardware access—and user space (ring 3), where user-level applications run with limited access, enforced by the CPU hardware (as the Intel x86 architecture “only supports 4 rings.” SAMSUNG-1015, 86). Intermediate privilege levels (ring 1 and ring 2) are also available. This hierarchical structure ensures that privileged operations, such as direct hardware access, memory management, or interrupt handling, are exclusively performed by trusted kernel code, while regular user applications must request these privileged services via controlled interfaces, system calls, or APIs.

3. *Memory Partitions*

56. Memory partitioning plays a critical role in ensuring that individual processes or partitions operate independently and securely within the computing environment. Such partitions provide isolated memory spaces allocated to different

processes or applications, preventing unauthorized access or interference. Modern systems implement this isolation primarily through virtual memory management (which, for example, allows Virtual Machines to “manage access to a prescribed amount of physical memory per [Virtual Machine] and ensure proper mapping of device resources.” SAMSUNG-1015, 159.), address space separation (which “defines the memory addresses [a process] can access” (SAMSUNG-1015, 41)), and hardware-based memory protection mechanisms (such as I/O MMUs described earlier).

57. Virtual address spaces, managed by operating system kernels, ensure that “each process has its own address space, that defines the memory addresses it can access.” SAMSUNG-1015, 41. In this way, processes perceive its own, isolated contiguous memory space, even though physical memory might be fragmented or shared among multiple applications. The OS and hardware collaborate to translate virtual addresses (used by applications) to physical addresses transparently using hardware components such as the Memory Management Unit (MMU). The MMU “translates these virtual I/O addresses to physical addresses.” SAMSUNG-1015, 82. Such translations are supported by page tables, which contain “mappings from virtual memory pages to their physical memory counterparts.” SAMSUNG-1015, 77. By enforcing strict boundaries through these mappings, the OS prevents applications from directly accessing memory allocated to other processes,

preventing malicious programs from being able to “compromise the secrecy and integrity of system and user application data.” SAMSUNG-1015, 77.

58. Hardware-based memory protection further supports this isolation. Mechanisms like page-level permissions (e.g., read-only, executable, or writable), segmentation, and hardware-enforced bounds checking ensure that even in the case of software bugs or attempts to deliberately breach isolation, a process is limited to accessing only its assigned memory. For example, hardware components called security protection modules (SPMs) in the Honeywell Secure Communications Processor (Scomp) system ensures that “access to words in memory is mediated by the SPM,” meaning, “since each instruction may make a memory reference for its instruction and its operands, plus access to page table entries, multiple access control checks may occur at each instruction.” SAMSUNG-1015, 80. While SComp had dedicated hardware for providing memory protection, modern systems also enforce memory protection via “check[ing] on the page table or TLB entry for read or write access.” SAMSUNG-1015, 80.

B. Policy Settings and Enforcement

59. Effective management of partitioned environments requires clearly defined policies along with mechanisms to enforce these policies rigorously. Policy enforcement ensures that resources are allocated appropriately, security constraints are maintained, and overall system behavior remains predictable and secure.

Common enforcement mechanisms include host firewalls (which provide “control of network communication” (SAMSUNG-1015, 44)), access control lists (ACLs) for resource permissions (as described above), bandwidth throttling mechanisms, and system call filters (i.e., UNIX “consists of hooks to check access for file or inode permissions on some system calls” (SAMSUNG-1015, 45)). These mechanisms allow operating systems or dedicated enforcement components to restrict application behavior, manage resource allocation, and maintain system security according to predefined rules.

1. Network Traffic Policies

60. Network traffic policies are essential for managing and securing the flow of data across system boundaries. These policies specify rules for acceptable traffic patterns, bandwidth allocations, allowable network endpoints, and other conditions under which network communication is permitted or blocked. Perhaps the most recognizable piece of software that enforces network traffic policies are firewalls. Firewalls “come in basically three flavors, depending on whether they filter at the IP packet level, at the TCP session level, or at the application level.” SAMSUNG-1014, 375.

61. At the packet-level, a firewall “merely filters packet addresses and port numbers.” SAMSUNG-1014, 375. Conversely, at the TCP session level, firewalls “reassemble and examine all packets in each TCP circuit.” SAMSUNG-

SAMSUNG-1014, 375. Finally, at the application-level, a firewall can “act[] as a proxy for one or more services, such as mail, telnet, and the Web. It’s at this level that you can enforce rules such as stripping out macros from incoming Word documents, and removing active content from Web pages.” SAMSUNG-SAMSUNG-1014, 375. Each layer a firewall operates at provides a tradeoff between context and speed; packet-level firewalls are simple and fast, yet unable to prevent more complicated attacks, while application-level firewalls operate at the other extreme, wherein they can express sophisticated rules (such as, “block emails with attachments that are greater than 100mb”), but can place a heavy demand on computational resources.

62. Measurement and enforcement of network traffic policies within operating systems, typically occurs through kernel modules or integrated firewall mechanisms (e.g., Berkeley Packet Filter (BPF)-based filtering, which is “a kernel agent called a packet filter, which discards unwanted packets as early as possible.” SAMSUNG-1017, 1.). These mechanisms monitor and evaluate network packets against predefined policy rules. For instance, a packet-level firewall configured with rules might restrict outbound connections to specific IP addresses, limit bandwidth allocation to certain applications or processes, or block packets exceeding specified size thresholds to ensure network performance and security.

2. *Application-Specific Policies*

63. As discussed above, a firewall can be an “application relay, which acts as a proxy for one or more services, such as mail, telnet, and the Web.” SAMSUNG-1014, 375. As the name implies, application-specific policies provide a more granular approach, tailoring enforcement directly to individual applications rather than general network behavior (and, more importantly, to their specific behaviors, as opposed to coarse information like a port number). The strength of application-specific policies is their ability to understand complex program semantics, where rules associated with e-mail “will understand RFC 822 headers, MIME-formatted attachments, and may well be able to identify virus-infected software.” SAMSUNG-1026, 185.

64. Effective enforcement of application-specific policies depends on accurately collecting detailed application-level parameters. For example, policies may be configured to check for “incorrect message formats, ... viruses, strange attachments, or even corporate secrets.” SAMSUNG-1026, 204. That is, policies for filtering application-level behavior are flexible enough to capture complex application semantics, such as data access patterns, file sizes, source IP addresses (which could correspond to a location within an organization), among other parameters. Indeed, popular filters such as the Berkely packet filter (BPF), tcpdump, Cisco routers, among others all have “languages [that] describe individual packets,

but they can describe connections and even entire service suites.” SAMSUNG-1026, 214.

65. One key advantage of application-specific policies over broader network traffic policies is their enhanced precision. Specifically, “application-level filters deal with the details of the particular service they are checking.” SAMSUNG-1026, 185. They enable finely tuned controls that directly reflect the operational requirements and security posture of specific applications. For example, application-specific policies can be “quite critical [in some environments],” as “mail can be checked for dirty words, indications that proprietary or restricted data is passing the gateway, ... web queries can be checked for conformance with company policies, and dangerous mail attachments can be stripped off.” SAMSUNG-1026, 186. Naturally, application-specific policies “are usually more complex than packet filters”, which “doesn’t need to understand much about the traffic it is limiting.” and therefore can incur higher performance overheads than their simpler, packet-based counterparts SAMSUNG-1026, 185.

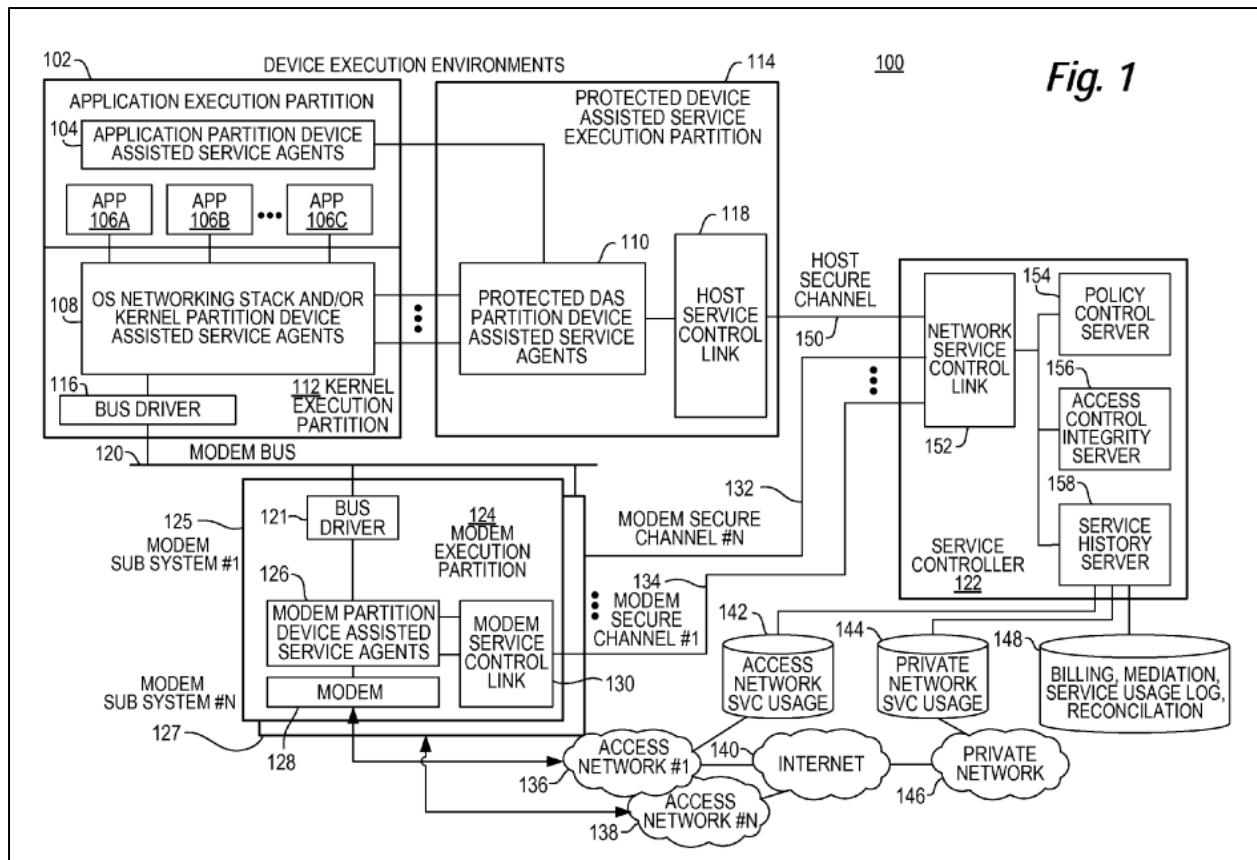
C. Redundancies in Computing Systems

66. In safety-critical and fault-tolerant system design, redundancy is a core practice. By having multiple agents or components in parallel, systems can “either detect or override the effects of the failed component.” SAMSUNG-1027, 3. For example, “by having two processors, we can detect the failure of a single processor;

by having three, we can use the majority output to override the wrong input of a single faulty processor.” SAMSUNG-1027, 3. In this way, systems can continue operation correctly even if one component fails. Importantly, aside from the continued system operation, redundant measurement components also provide independent verification of the same quantity. The outputs of such redundant measurement components are “compared (by the Compare unit) to the output of the voter to identify a faulty primary (if any).” SAMSUNG-1027, 25-26. If any of the components produce values that disagree with most of the others, it can be swapped out for “a spare module instead.” SAMSUNG-1027, 26.

VIII. OVERVIEW OF THE '144 PATENT

67. The '144 Patent discloses a mobile phone having partitioned environments that receives policies and/or rules from a remote server. *See, e.g.*, SAMSUNG-1001, FIG. 1. More specifically, the '144 Patent is directed to “a wireless end-user device that has a wireless wide-area network (WWAN) modem and multiple execution environments.” SAMSUNG-1001, Abstract. The '144 Patent, as shown in FIG. 1, shows four execution partitions. SAMSUNG-1001, 5:16-41.



SAMSUNG-1001, FIG. 1.

68. FIGS. 2-11 show other embodiments of the environment in FIG. 1, where each figure “illustrates another secure execution environment for device assisted services.” SAMSUNG-1001, 1:38 – 2:3. The ’144 Patent teaches that “each of these execution partitions are optimized for different software functions, each providing programs with the basic physical memory, data memory, CPU or APU or modem processor execution resources, high level and/or low level OS, memory management, file storage, I/O device resources (e.g., user interface (UI), peripherals, etc.), network communications stack, other device resources, and/or other resources

that are required or used for operation of the programs.” SAMSUNG-1001, 5:30-41. In more detail, “FIG. 1 illustrates an application execution partition 102 and a kernel execution partition 112, which are shown as separate partitions within the device execution environments” and have a host secure channel 150 to communicate with the service controller 122 via a network service control link 152. SAMSUNG-1001, 6:61-7:31.

69. The ’144 Patent teaches that “if the network stack service usage reporting is hacked, corrupted, and/or compromised, there is a secure additional or back-up service measure located on the device and/or in the network.” SAMSUNG-1001, 12:29-33. One example of a back-up service is the modem agent 226 shown in FIG. 2. SAMSUNG-1001, 12:29-33. The ’144 Patent teaches that “the service measure provided by the modem agent 226, modem encryption 225, and/or modem bus 120 ... can be executed in a protected partition (e.g., modem execution partition 124 as shown in FIG. 2...).” SAMSUNG-1001, 12:37-41; *see also id.*, 16:12-15 (“In some embodiments, the access control integrity agent 422 provides additional security, for example, in the event that the protected DAS partition 814 is breached or compromised.”). As shown in the context of FIGS. 4-5, an application identify agent 420, an access control integrity agent 422, a policy control agent 424, a policy implementation agent 426, and a service measure/service monitoring agent 428 can be in the protected DAS execution partition 414. *See* SAMSUNG-1001, 14:45-

15:29. In some implementations, a service measurement and/or policy control drivers 210 can be executed in the protected DAS partition 214, while in other embodiments, the service measurement / policy control drivers 210 can be executed in the kernel execution partition 712. SAMSUNG-1001, 12:18-24, 15:59 – 16:15. As demonstrated herein, the claims of the '144 Patent were known or obvious.

IX. PROSECUTION HISTORY OF THE '144 PATENT

70. The application for the '144 Patent was filed on November 20, 2015.

71. This application claims priority to multiple previously-filed non-provisional applications, the earliest of which (U.S. Application No. 12/380,780) was filed on March 2, 2009. SAMSUNG-1001, face. This application also claims priority to multiple provisional applications, the earliest of which was filed on January 28, 2009. SAMSUNG-1001, face.

72. During prosecution, the as-filed claims did not encounter any rejections (based on prior art or otherwise) and were allowed in the first action issued from the Office.

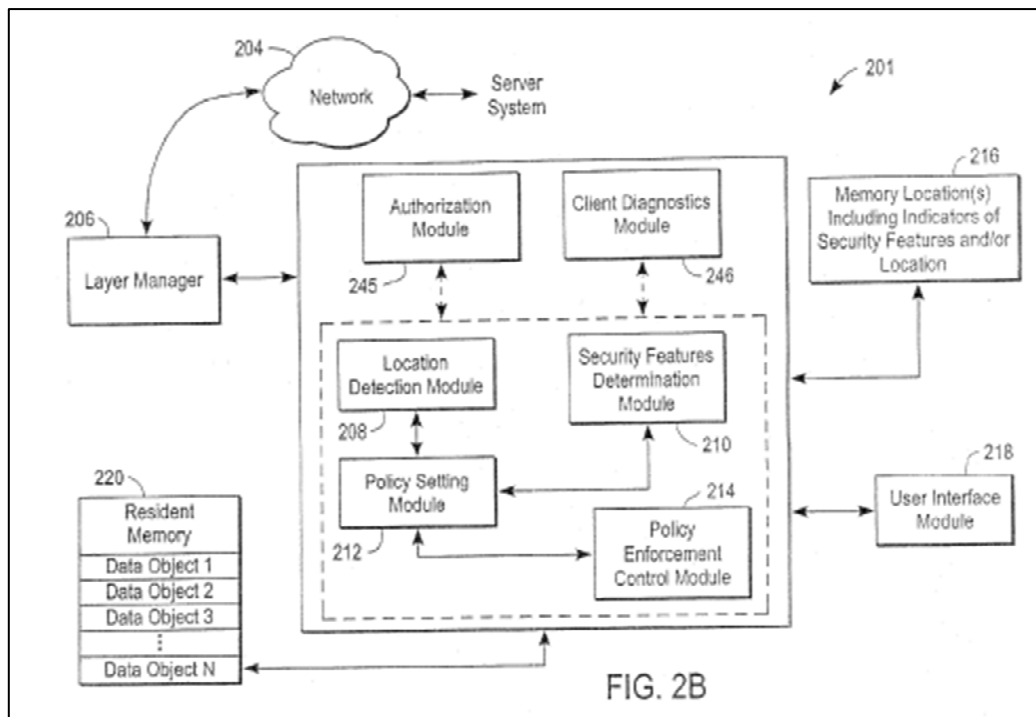
73. In the Notice of Allowance (NOA), the Examiner stated that the claims “are determined to be directed to patent-eligible subject matter and the Examiner does not consider any of the pending claims to be abstract....” SAMSUNG-1002, 25; id., 24-26.

X. OVERVIEW AND COMBINATIONS OF PRIOR ART REFERENCES

A. U.S. Pat. App. Pub. No. 2008/0052395 A1 (“Wright”)

74. Wright provides a system for administering “protection of data accessible by” a mobile device based on criteria,” such as device location in a networked environment and/or “security features associated with” the device. ¶13.

75. In FIG. 2B (below), Wright depicts a “mobile computer system” (also referred to as client system or client-side system 201) that communicates with a server system. SAMSUNG-1005, ¶53 (explaining that client system 201 can be implemented as software executing on mobile computing device).

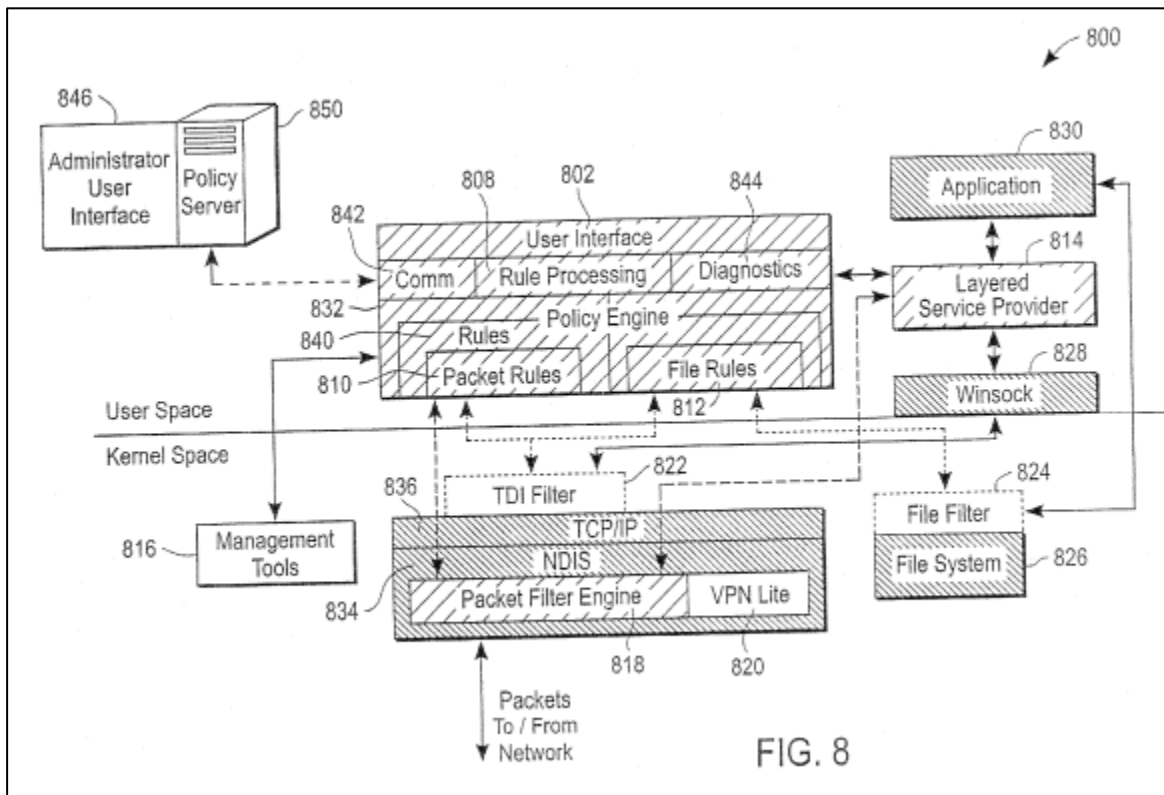


SAMSUNG-1005, FIG. 2B

76. In the above depicted embodiment, Wright's client system includes multiple components/modules (depicted above), which are responsible for the client-side management and enforcement of policies as they relate to inbound or outbound network traffic (as received and processed by layer manager 206). SAMSUNG-1005, ¶¶65-67, 54. The client side system 201 "monitors the user's changes in location and/or security features" (via the location detection and security features determination modules 208 and 210) and "applies the appropriate policies automatically as the user moves about or different security features are activated or deactivated." SAMSUNG-1005, ¶53, 67-68. Policy setting module 212 "receives, installs and updates security information including the security policies and/or software information" received from the server system. SAMSUNG-1005, ¶¶68, 73. Policy enforcement module 214 includes "instructions for enforcing the security policy currently set by the policy setting module 212." *Id.*, ¶71. The client system 201 also includes "a client diagnostics module (e.g. FIG. 2B, 246) [that] periodically and automatically initiates tests," and reports "[r]esults including errors from these tests ... over a network 204 (e.g. Internet) to the" server system. *Id.*, ¶61, Abstract.

77. Like the FIG. 2 embodiment, Wright's FIG. 8 provides an embodiment of a server system interacting with a client mobile device for protection of data accessible by the mobile device. SAMSUNG-1005, ¶¶235, 65. Much like the FIG.

2 embodiment, FIG. 8 (shown below) depicts a “policy server 850 having an administrator user interface 846” that “is communicatively coupled over a network with the client side system through a communication port 842.” SAMSUNG-1005, ¶¶236-237; *see id.* ¶235 (“...a server system embodiment for administering the protection of data accessible by a mobile device.”).



SAMSUNG-1005, FIG. 8

78. As depicted above, the client side system is partitioned into a user space (also referred to as application space) and a kernel space. SAMSUNG-1005, FIG. 8. The application/user space includes a policy engine 832, a user interface module 804, applications 830, and a layer services provider 814. *See id.*

79. Policy engine 832, which performs functionality similar to that of policy setting module of FIG. 2B, has communication interfaces with “management tools 816 of the operating system,” “file filter 824 operating in the kernel space that controls access to the file system 826,” “user interface module 802,” “layered service provider (LSP) 814,” and “packet filter engine 818 operating within a driver 834” in the kernel space, which also includes an application filter (implemented as transport driver interface in FIG. 8). SAMSUNG-1005, ¶¶237-238 (explaining diagnostics module 844 and rules processing 808 are part of policy engine 832), 283.

80. A communication interface between one or more applications 830 and LSP 814 facilitates communications therebetween. SAMSUNG-1005, ¶239. In some implementations, LSP 814 “determine[s] which application (e.g. browser e-mail application such as Outlook Exchange®) is accessing the network and what networking services the application will be using.” SAMSUNG-1005, ¶240.

81. This application information is used by kernel components, e.g., to perform enforcement of policies relating to network packet traffic. SAMSUNG-1005, ¶283 (explaining that packet filter engine 818, TDI filter 822 provides implementation examples of “enforcement mechanisms” performed by “policy enforcement control module 214” in FIG. 2B). For example, policy engine 832 provides rules (related to particular security policies) to the packet filter engine 818 and/or the application filter 822, which perform application-specific filtering of

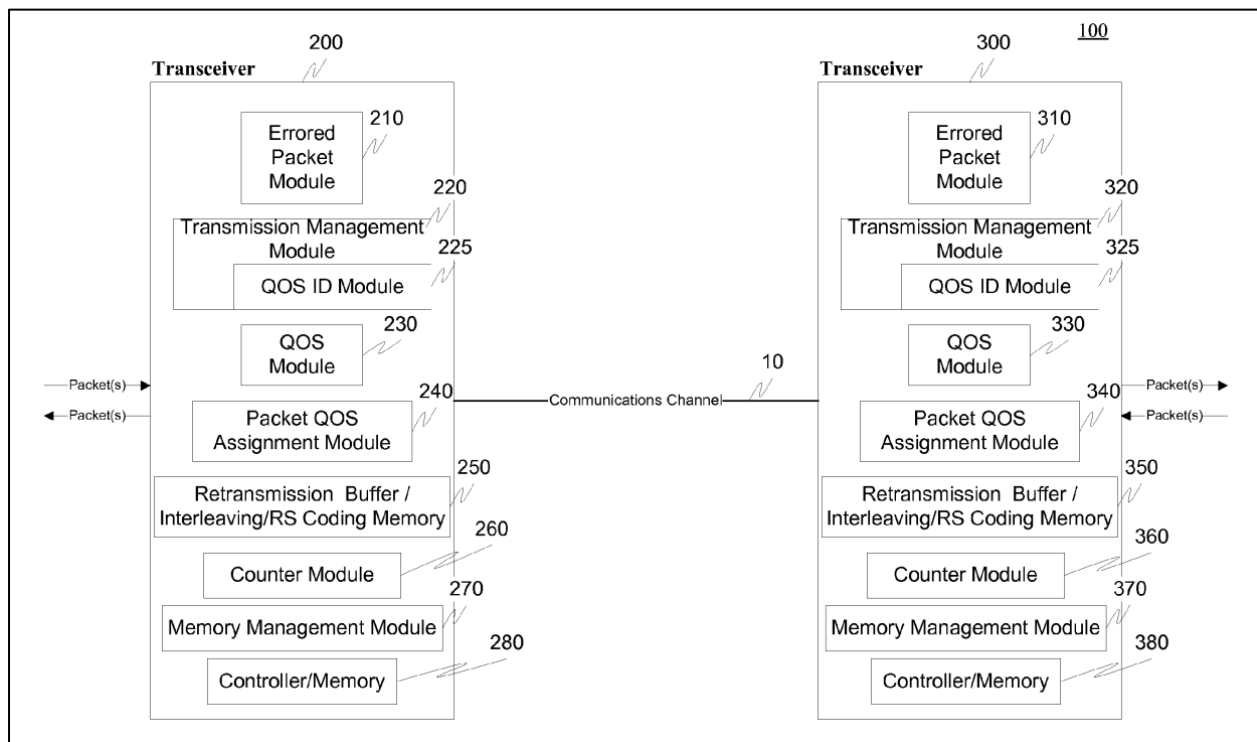
packet traffic based on those rules. SAMSUNG-1005, ¶269 (“rules are passed to the packet filter engine as well as the TDI filter 822 for application specific filtering”); *see id.*, ¶239 (providing application-specific rules based on current security policy, including “which applications can or cannot access the network, as well as the target IP addresses and ports that they are or not allowed to use”).

82. Software agents (e.g., TDI filter 822, packet filter engine 818) in the kernel can employ “an enforcement mechanism [that] is referred to as stateful filtering.” SAMSUNG-1005, ¶264. As part of such packet filtering, “adaptive port blocking” may be performed, which can be used to apply rules to “a set of ports, port types, and actions.” SAMSUNG-1005, ¶264. In this mode, “each port [can be] in one of three modes: open, closed, or stateful” such that (1) “[w]hen the port is open, all traffic (both incoming and outgoing) on that port is permitted to flow through the packet filter,” (2) “[w]hen the port is closed, all traffic on that port is blocked (both incoming and outgoing),” and (3) “[w]hen the port is stateful, all outgoing traffic on that port is permitted to flow through the packet filter, and incoming responses to that outgoing traffic are allowed back through, but unsolicited incoming traffic is blocked.” SAMSUNG-1005, ¶¶264, 267.

B. U.S. Pat. App. Pub. No. 2009/0300450 A1 (“Tzannes”)

83. Tzannes’ invention relates to transmission/retransmission of packet traffic in a communication environment. SAMSUNG-1006, ¶2; *see id.*, ¶¶185-189 (explaining that Tzannes’ system works “for any type of communication system in any environment” and is implemented on “computing devices” including transceivers/modems), ¶189 (explaining that transceiver and modem are used interchangeably), ¶197 (broadly describing what a “packet” encompasses. Tzannes’ system enables identification of different packets being transmitted, based on which the system facilitates “forwarding of low-latency packets without delay” and storing “error-sensitive packets ...for possible transmission.” SAMSUNG-1006, Abstract.

84. An example of Tzannes’ communication system 100 is provided below:



SAMSUNG-1006, FIG. 1

85. As shown, communication channel 10 connects transceivers 200 and 300. SAMSUNG-1006, ¶¶191, 189 (communication channel can be a wired or wireless link). Each transceiver (200/300) includes “in addition to well-known componentry,” errored packet module 210, transmission management module 220, QOS [Quality of Service] ID module 225, QOS module 230, packet QOS assignment module 240, retransmission buffer/interleaving/deinterleaving/RS coding/RS Decoding memory 250, counter module 260, memory management module 270, and controller/memory 280.” SAMSUNG-1006, ¶¶191-192.

86. Tzannes’ techniques facilitate appropriate transmission/retransmission of packets according to their specified QOS level, which indicates the importance of certain service metrics of the packets. SAMSUNG-1006, ¶¶198-204; see id., ¶199 (identifying latency and packet error rate (PER) as two service metrics). In the context of FIG. 1 above, when transmitting transceiver 200 receives a packet from higher layers, the transceiver’s QOS module 230 can identify packets based on their QOS level, and together with QOS assignment module 240, a QOS identifier is associated with the packet. SAMSUNG-1006, ¶213. If the QOS identifier indicates a low PER packet, transmission management module 220 forwards the packet for storage in retransmission buffer and subsequently transmits the packet. See SAMSUNG-1006, ¶214. Alternatively, if the QOS identifier indicates a low latency

packet, the packet can be transmitted to the receiving transceiver/modem using the transmission management module 220 (i.e., without storage in the retransmission buffer). *See id.*

87. “For successful retransmission, the receiving modem should be able to inform the transmitting modem which packet, or packets, need to be retransmitted. One exemplary way of performing this is by transmitting packets with an appended bit field that contains a counter indicating the place of each packet in a stream of packets. This counter value is also known as a Sequence ID (SID). For example, a bit field containing a 16-bit counter could be appended to each packet and the counter module 260 would be incremented by one after each packet was transmitted. In cooperation with the packet assignment module 240, a packet counter field could be appended to the packet in a number of places, for example, at the beginning or end of the packet, or at the beginning or end of the packet header.” SAMSUNG-1006, ¶216. In some instances, packets received from a higher layer can already have the packet count information included in the packet. *Id.*, ¶217. Additionally, a CRC field can be appended to the packet counter field that computes a cyclic redundancy check over the packet counter field bits, and which can be used by the receiver to determine if the packet counter field is received correctly (without bit errors). *Id.*

88. At the receiving transceiver/modem 300, (1) “[i]f the received packet is identified as a low-latency packet by the QOS ID module 325, the packet is passed to a higher-layer” or (2) “[i]f a received packet is identified by the QOS ID module 325 as a low-PER packet, the packet is forwarded, with the cooperation of the transmission management module 320, to the retransmission buffer 350 for a minimum amount of time before passing to a higher-layer.” SAMSUNG-1006, ¶220. The retransmission buffer 350 and the errored packet module 310, can be used to detect missing or errored packets. SAMSUNG-1006, ¶226-227. For example, “if the errored packet module 310, in cooperation with the counter module 360, detects the receipt of a packet with a counter number equal to 5, wherein the errored packet module 310 is expecting to receive a packet with a counter equal to 3, the errored packet module 310 can determine that two packets, namely packets numbered 3 and 4, were lost due to errors.” SAMSUNG-1006, ¶227. In such instances, the transmitting transceiver 200 can retransmit (based, e.g., on a request or other messaging from the receiving transceiver 300) the missing or otherwise errored packets to the receiving transceiver 300. *See* SAMSUNG-1006, ¶¶230-248.

C. Wright-Tzannes Combination

89. Wright teaches transmission of packet traffic over a network between a mobile device and one or more components in the network. *See generally*

SAMSUNG-1005; §X.A *supra*. In the same field of network communication systems, Tzannes similarly teaches packet communications between computing devices over wireless or wired links. *See generally* SAMSUNG-1006; §X.B *supra*.

90. Tzannes additionally provides techniques facilitating retransmission of lost or dropped packets based on QOS policies applicable to packets and using components and functionality implemented at the sending and receiving transceivers/modems. *See supra* §X.B. As explained in §X.B, Tzannes teaches packet transmission flows based on QOS levels applied to particular packets and further provides a packet counter value in the packet to facilitate packet retransmission. As to the latter, Tzannes teaches a packet counter module implemented at a transceiver/modem, which calculates a counter value that is appended to the packet. SAMSUNG-1006, ¶216 (“One exemplary way of performing this is by transmitting packets with an appended bit field that contains a counter indicating the place of each packet in a stream of packets. This counter value is also known as a Sequence ID (SID). For example, a bit field containing a 16-bit counter could be appended to each packet and the counter module 260 would be incremented by one after each packet was transmitted.”).

91. Tzannes also contemplates that the packet count values may be provided by higher layers when providing the packets to the modem/transceiver for subsequent delivery. SAMSUNG-1006, ¶217. Given that the packet count values

could be provided by higher layers before the packet is provided to the modem for transmission, a POSITA would have recognized that Tzannes packet counter functionality can be implemented both at the modem and at higher layers within a computing device. For retransmission then, the receiving modem informs the sending modem/transceiver which packet or packets need to be retransmitted based on the packet counter values included in the received packets. *See id.*; ¶¶216-217.

92. A POSITA would have found obvious to combine the above-described teachings of Tzannes' QOS-based packet transmission/retransmission functionality with those of Wright. In the combination, Wright's system (as described in §X.A), would implement the functionality and components recited in Tzannes(including at the transceiver/modem and at higher layers), and would facilitate retransmission of lost or dropped packets to a receiver. A POSITA would have been motivated, and would have expected reasonable success, in so implementing Tzannes functionality for at least the below reasons.

93. **First**, Wright recognizes the importance of “alleviat[ing] [network] congestion” (SAMSUNG-1005, ¶255), and in the relevant timeframe, it was well-known that network congestion often resulted in lost or erroneous packet transmissions, which are remedied by retransmission of the lost or erroneous packets. A POSITA would have thus been motivated to combine the above-described teachings in Tzannes with those of Wright to provide a more flexible and

resilient packet transmission system that accounts for network congestion and ensures any lost or undelivered packets are delivered, particularly when QOS rules/service levels call for such delivery. *See, e.g.*, SAMSUNG-1006, ¶228 (“Once a packet(s) is found to be in error, there are several exemplary ways in which a receiving modem can communicate information to the transmitting modem indicating that a retransmission of one or more packets is required.”); *id.*, ¶283 (“if the communications channel was not performing well, e.g., an increase in bit errors, it may be advantageous to increase the retransmission capability while decreasing the FEC/interleaving capability or vice-versa, which could have an impact on how the memory sharing should be configured.”); *see also id.*, ¶¶226-227, 261-263. The facilities of Tzannes would enhance the context-sensitive (e.g., application oriented) policies of Wright with the ability to more effectively control network communication. For example, the per-packet QoS policies and recovery mechanisms would enable the security policies of Wright to maintain security while avoiding performance challenges. *See* SAMSUNG-1006, ¶218 (facilitating retransmission via provided packet counter value for packets with a QOS service level demanding low packet error rate).

94. ***Second***, and relatedly, a POSITA would find it beneficial to implement a traffic management technique that prioritizes certain types of network packets at a mobile device. Tzannes discloses one such technique, when it teaches a QOS

technique using a “modem to determine the QOS level of each packet” and performing packet transmission/retransmissions based on such QOS level. SAMSUNG-1006, ¶200. A POSITA would have been motivated to implement such teachings in combination to minimize packet loss for certain applications on the mobile device, e.g., by prioritizing bandwidth for real-time applications while ensuring low latency for applications like gaming and VOIP applications. For example, because Tzannes’ QOS technique attempts to implements metrics to address “QOS metrics [such as] delay (or latency) and Packet Error Rate (PER),” a POSITA would have found such traffic shaping advantageous. *See* SAMSUNG-1006, ¶5.

95. Indeed, a POSITA would recognize that, without such QOS techniques, packets may be dropped, delayed, and never transferred, which can result in poor performance for certain types of applications. In contrast, Tzannes teachings, in combination, would advantageously enable prioritizing certain types of packets, e.g., those marked with a QOS level indicating low packet error rate, and facilitating retransmission of the same, e.g., using packet counter values embedded within the packet. Such a facility would help reduce the communication inefficiencies common with the implementation (and interactions between) security policies of Wright. Specifically, a POSITA understood that there is an inherent trade-off between the performance the security offered by security policies over a network. For example,

“[t]o incorporate security, an appropriate access control mechanism must be in place to enforce the authorization specifications reflecting the above security and privacy policies. ... While this solves a security problem, it creates a performance problem. Often, enforcing security incurs overhead, and as a result may degrade network performance of a system.” SAMSUNG-1033, 553.

96. *Third*, in the relevant timeframe, packet transmission and retransmission subject to Quality of Service (QoS) rules was known to provide a framework by which network traffic was controlled for different scenarios, particularly where network capacity is limited. *See, e.g.*, SAMSUNG-1020, 128 (“For example, email applications may be delay insensitive but error sensitive. On the other hand, video streaming may be delay sensitive but could tolerate a higher packet error rate. Hence, it is a challenging task to mix the data and multimedia applications with different QoS requirements efficiently in a cellular system (in particular the air interface QoS differentiation).”); SAMSUNG-1028, Abstract (“Multimedia telephony is a delay-sensitive application. Packet losses, relatively less critical than delay, are allowed up to a certain threshold. They represent the QoS constraints that have to be respected to guarantee the operation of the telephony service and user satisfaction.”). Cellular networks are sensitive to delay. As such, a POSITA would understand that techniques for improving QoS would provide a

suitable mechanism for providing stability of the network and user satisfaction, by ensuring delays do not exceed an acceptable threshold.

97. Consistent with this, Tzannes teaches packet transmission/retransmission techniques based on QOS service rules applicable to each packet. *See* §X.B, *supra*. Wright similarly contemplates “alleviating” network congestion and provides policies governing transmission of packets received or transmitted by the device. *See* §§X.A-X.B, *supra*. A POSITA would have thus been motivated to combine well-known QOS policies and associated packet transmission/retransmission functionality with Wright to achieve predictable results of enforcing QOS policies, as taught by Tzannes, as they relate to transmitted packet traffic, as contemplated by Wright.

98. ***Fourth***, Wright teaches or renders obvious a WWAN modem for transmission of packet traffic over a network. Given that Tzannes contemplates its functionality being implemented at a modem/transceiver and that its techniques can be implemented for any type of computing device or communication system, a POSITA would have had expected reasonable success in implementing components and associated functionality in Tzannes’ modem/transceiver as part of Wright’s WWAN modem. In this regard, it would have been straightforward and well-within a POSITA’s skill level to implement Tzannes’ modem components, including the packet counter module, as part of Wright’s WWAN modem, to facilitate, among

other functionality, the computation and insertion of the packet count values in a packet counter field of a packet.

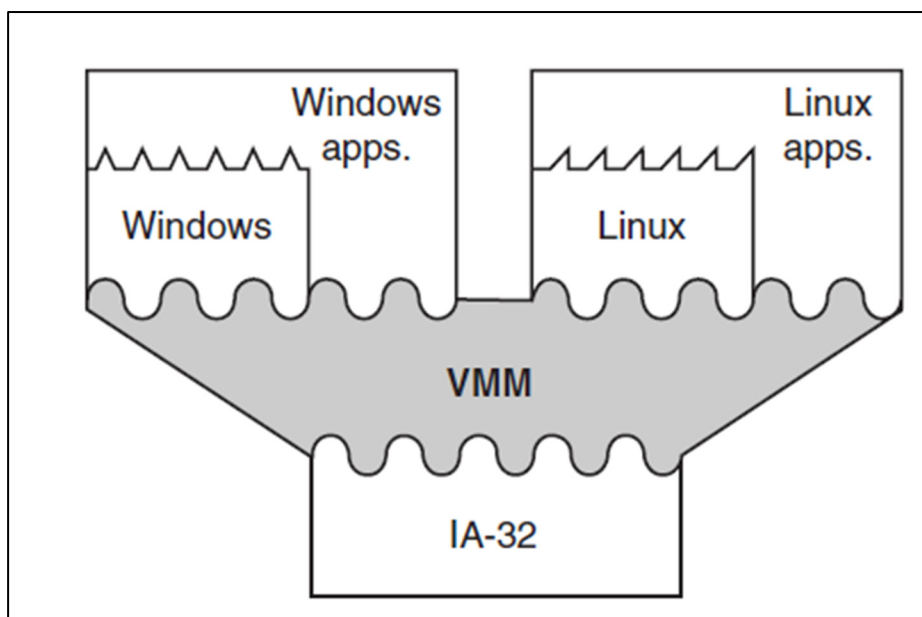
99. Moreover, Tzannes also contemplates QOS policy and associated rules, as well as inclusion of packet count values, being specified by higher layers of the computing device (i.e., layers higher than that of the modem). SAMSUNG-1006, ¶¶198, 212. Wright similarly contemplates higher layer processing of policies, given its disclosure of policies being configured by policy engine 832 and determining and providing of network parameters (e.g., at OSI layers 6 and 7) in some implementations by LSP 814—both of which are in the application/user space and at layers higher than the modem. SAMSUNG-1005, ¶¶236-281, FIG. 8. It would thus be obvious and well-within a POSITA’s skillset to configure QOS specification and packet count functionality at higher layers, as contemplated by Tzannes, within components of Wright’s system that similarly operate at higher layers, and which are tasked with similar policy configuration and network analysis/parameter determination.

D. Smith, *Virtual Machines: Versatile Platforms for Systems and Processes* (“Smith”)

100. Smith describes various types of virtual machines (VMs), including system VMs, that “provide a complete system environment in which many processes[] can coexist.” SAMSUNG-1007, 25; SAMSUNG-1012, 72. VMs

“provide a secure way of partitioning major software systems that run concurrently on the same hardware platform” such that “[s]oftware running” in one VM environment “is isolated from software running on other guest systems.” SAMSUNG-1007, 18, 26, 29; SAMSUNG-1012, 72.

101. “In practical terms, a virtual machine executes software (either an individual process or a full system, depending on the type of machine) in the same manner as the machine for which the software was developed. The virtual machine is implemented as a combination of a real machine and virtualizing software,” which may be implemented with “...resources different from the real machine, either in quantity or in type.” SAMSUNG-1007, 15-16. Smith further explains that VMs enable “support [of] different operating systems simultaneously.” SAMSUNG-1007, 26; SAMSUNG-1012, 72-73. In one example, Smith demonstrates a Windows VM running Windows applications and a Linux VM running Linux applications on a single device. *Id.*



SAMSUNG-1007/SAMSUNG-1012, FIG. 1.11

E. Wright-Smith Combination

102. As discussed above, Wright discloses a mobile phone having partitioned environments that receive policies and rules from a remote server. *See supra*, §X.A. To the extent that Wright does not disclose a secure virtual environment, that would have been obvious in view of Smith.

103. Smith teaches implementations of various types of VMs. *See* SAMSUNG-1007, 15-16, 18, 25-26, 29. Smith additionally explains that VMs “provide a secure way of partitioning major software systems” to isolate software therein. SAMSUNG-1007, 26, SAMSUNG-1012, 72; §X.D. Smith also explains that “[r]unning each application in its own virtual machine increases the robustness

of the system,” since erratic behavior on a VM is “less likely to affect the operation” of applications running on a different VM. SAMSUNG-1007, 29; 13; SAMSUNG-1012, 506-507; 56. “Multiple, replicated virtual machines can be implemented on a single hardware platform...,” and Smith teaches that “[t]he different system environments (possibly with different operating systems) also provide isolation and enhanced security.”

104. In the relevant timeframe, a POSITA would have been motivated to increase Wright’s robustness and enhance security, and to do so, a POSITA would have further been motivated to implement Wright’s protected execution partition to run in a VM separate from an environment running the user applications (e.g., which may or may not be implemented in another VM). The resulting Wright-Smith wireless end-user device includes, a VM Monitor (VMM), which is virtualizing software that provides two VMs, one VM hosting Wright’s policy-related engines (policy engine 832, LSP 814, and Winsock 828), and another VM hosting a second OS running conventional applications, such as Wright’s applications 830. SAMSUNG-1007, 24; SAMSUNG-1012, 72-73; see also my analysis in Section VII.A, above.

105. A POSITA would have expected success in implementing one or more VMs with Wright, because such an implementation would have required routine programming knowledge well-within a POSITA’s skill. Indeed, this would have

amounted to using a known technique (VMs) to improve operation of a known device (wireless end-user device) to achieve predictable results (running software in a VM). A POSITA would have found it obvious to implement a portion or all of the protected execution partition in a *secure execution environment* (e.g., a virtual machine (VM)) for at least the following additional reasons.

106. *First*, a POSITA would have found it obvious that a VM (i.e., *secure execution environment*) would provide an isolated environment that would enable components of the protected execution partition to operate independently without unauthorized access to or from other modules in Wright's user space. A POSITA would understand that isolation can be enforced by a VM (e.g., a hypervisor), and a VM would enable segmentation of computer resources and the ability to manage permissions across VMs. Using a VM to provide an isolated environment reduces ability for malicious activities in one agents to impact the *secure execution environment* (e.g., the PEP). A POSITA would understand that isolating environments would facilitate maintaining reliability and confidentiality of data, which is a goal of Wright. SAMSUNG-1005, ¶¶ Abstract, 6-17, 285; *see my analysis above for §VII.A.*

107. *Second*, a POSITA would have found it obvious that a VM (i.e., *secure execution environment*) would provide a suitable environment for encrypted data. Wright already teaches that one or more device agents, e.g., policy engine 832,

implements policies based on a determination of whether a file or data is encrypted. *See, e.g., SAMSUNG-1005, ¶¶296, 299-300.* A POSITA would understand that encrypting data in a VM ensures that sensitive data within a VM is inaccessible to unauthorized users or module. A POSITA would have found it obvious that a VM could reduce the opportunity for unauthorized access was suitable controlling access by minimizing user access rights to a minimum requirement for functionality and regular audits of access privileges. Wright teaches that one or more device agents, such as the policy engine 832, implements policies based on a determination of whether a file or data is encrypted. *See, e.g., SAMSUNG-1005, ¶¶296, 299-300.* Thus, a POSITA would find it obvious to implement Wright's policy engine 832 within one or more virtual machines, i.e., ***secure execution environment***.

108. ***Third***, a POSITA would understand that a VM would enable regular back up of data and software. For example, regular backup of the VM ensures that the ***secure execution environment*** (PEP) of the Wright's wireless end-user device would enable quick restoration of the device in the event of hardware failure, data loss events, or ransomware. A VM would enable snapshots or backups of a module executing in Wright's PEP. A VM snapshot would provide a secure, nearly immutable storage of inputs and outputs to the PEP to prevent unauthorized modification. A POSITA would understand that the snapshot / backup provided by the VM provides a reliable path to recovery of the ***secure execution environment***

(e.g., PEP). A POSITA would understand that a VM machine would be beneficial in the event that Wright's *secure execution environment* (e.g., PEP) was compromised. If there was a compromise of the PEP, the use of a VM enable for an uncompromised VM at an earlier time to be implemented.

109. **Fourth**, a POSITA would understand that a VM would enable segmentation of network traffic. For example, implementing the *secure execution environment* (e.g., PEP) would enable relative isolation of traffic to and from the PEP to be separated from other parts of the mobile device infrastructure, such as Wright's applications 830. A POSITA would understand that segmentation minimizes the risk of migration of malicious software or issues that would otherwise compromise security of sensitive, policy management operations of the policy engine 832 and LSP 818 in the PEP. For example, implementing a virtual firewall between the *secure execution environment* (e.g., PEP) and other applications would segment PEP traffic from other network traffic, thereby increasing security by limiting the *secure execution environment*'s exposure to malicious data.

XI. MANNER IN WHICH THE PRIOR ART REFERENCES RENDER THE '144 CLAIMS UNPATENTABLE

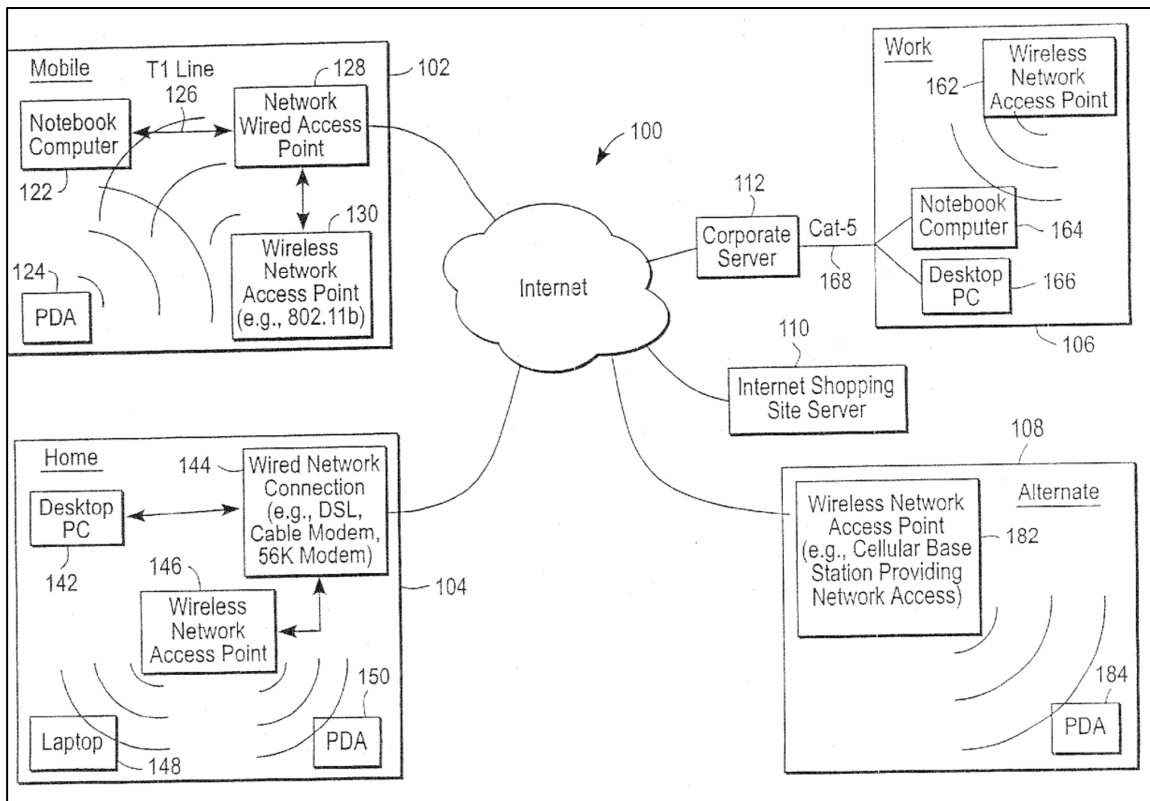
A. Ground 1A: Wright Renders Obvious Claims 1-11, 13, 15, and 18

[1.1] A wireless end-user device comprising

110. Wright discloses a *wireless end-user device*. Wright’s system and techniques “provide[] for the administration of the protection of data accessible by a mobile device,” which “include portable computing devices that access a network or another computer.”¹ SAMSUNG-1005, ¶15. Wright refers to its mobile devices as “wireless device[s]” because they can “access a network ... through a wireless connection.” *Id.*

111. As shown in FIG. 1 below, Wright teaches different types of end-user devices, including, *e.g.*, notebook computer 122, personal digital assistant (PDA) 124, notebook computer 164, PC Desktop 166, and mobile devices 148, 150) disposed at several location categories (102, 104, 106, 108), via a network 100. SAMSUNG-1005, ¶¶49-52, FIG. 1.

¹ Emphasis added throughout unless otherwise noted.



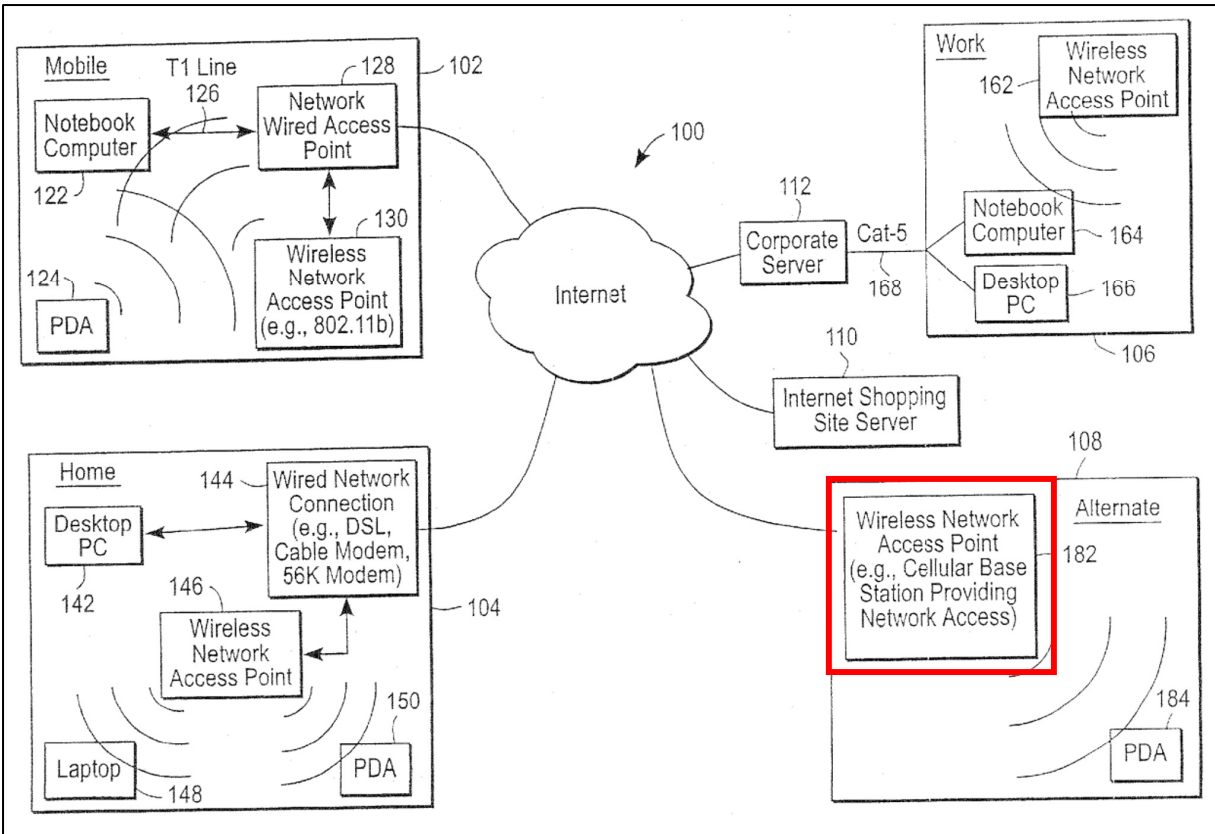
SAMSUNG-1005, FIG. 1

112. Similarly, in the context of FIGS. 2A-2B and FIGS. 7-8, Wright teaches systems implemented using mobile devices, which a POSITA would have understood would include the example end-user devices described in the context of FIG. 1. SAMSUNG-1005, ¶¶53-84; FIGS. 2A-2B, 7-8. Additional details of Wright's wireless end-user device are described below with reference to elements [1.2] to [1.10].

[1.2] a wireless wide area network (WWAN) modem to receive and transmit Internet data between the device and at least one WWAN when configured for and connected to the at least one WWAN; and

113. A POSITA understood or found obvious that Wright's wireless end-user device (*see* [1.1] *supra*) includes a **WWAN modem** given Wright's teachings that this device includes and uses a network interface card or network adapter to access a wireless cellular network (*e.g.*, 3G, GPRS, GSM), which is a type of WWAN.

114. In more detail, Wright's end-user device can communicate with wireless cellular networks, such as GPRS, GSM, and 3G. *See, e.g.*, SAMSUNG-1005, ¶¶289, 293. In one example, Wright's Fig. 1 (reproduced below) shows and describes a mobile device (*e.g.*, PDA 184) communicating with a wireless network access point, which is "associated with a cellular base station providing network access through a" GPRS, GSM, 3G, or "other kind of mobile device wireless communication system." SAMSUNG-1005, ¶52, 73, Fig. 1.



SAMSUNG-1005, FIG. 1 (annotated)

115. Wright further contemplates that network interface cards (NICs) or network adapters are implemented at the device to facilitate the device's wireless communication with such wireless cellular networks. For example, Wright teaches that there may be "different security polic[ies] ... applied for particular classes (e.g., 802.3, 802.11a, 802.11b, GPRS, GSM, 3G) of NICs" and/or that a policy engine resident on the device can determine "from the operating system the class" (e.g., GPRS, GSM, 3G) of the network adapters on the mobile device.

SAMSUNG-1005, ¶¶73, 289, 293.

116. A POSITA would have understood that (1) wireless cellular networks, such as GPRS, GSM, and 3G, are wireless wide area networks or WWANs, and (2) a device's "network adapters" (NAP) and/or "network interface cards" (NIC) that enable access to wireless networks are examples of modems. *See, e.g., SAMSUNG-1013*, 78 ("General Packet Radio System (GPRS) introduced packet-switched capabilities to wireless networks."); *see id.*, 77 ("The Global System for Mobile Communications (GSM) is the most extensively used 2G technology in the world."), 79 ("Enhanced Data Rates for Global Evolution (EDGE) is a technology that allows current GSM networks to offer 3G services within existing frequencies."), 83-84 ("Table 3.5 summarizes the wireless networks that are currently being used for wireless data, showing the network frequency, data transfer rates, and deployment regions."). A POSITA would have thus understood or found obvious that, because Wright's mobile device includes NICs and NAPs that enable access to one or more WWANs, these device components are **WWAN modems**. *See, e.g., SAMSUNG-1013*, 9 ("In addition to the mobile devices, other hardware is often required for wireless solutions. This may include wireless modems, network cards, and wireless access points."); *see also SAMSUNG-1023*, 768 ("In order to determine the performance improvements over the 3G networks, we have carried out TCP bulk data transfer measurements (down-link and uplink) at several different locations in Hong Kong by using a single laptop equipped with a USB modem containing an

HSDPA data card connecting to our server over a lightly loaded HSDPA network.”); SAMSUNG-1045, 11:12-16 (“For instance, network interface circuitry 40 could be, or could include, an Ethernet network interface card (NIC) and port, a cable modem and coaxial cable port, and/or a DSL modem and RJ-11 port. Other examples are also possible.”) .

117. Wright further teaches or suggests that its *WWAN modem* (e.g., mobile device’s NIC/network adapter) *receives and transmit Internet data between the device and at least one WWAN*. For example, Wright teaches its mobile device accessing the Internet via a WWAN (and its associated network access point). As shown in FIG. 1, a PDA communicates with a wireless network access point such “a cellular base station providing network access through a” GPRS, GSM, or 3G network, which in turn enables access to the Internet, such as e.g., to “an Internet shopping site server 110” that the user of the device may be accessing. SAMSUNG-1005, ¶¶51-52, FIG. 1; *see id.*, ¶85 (describing client mobile device “connect[ing] via the Internet” to one or more servers), ¶222, ¶¶231-232 (device being allowed “Internet access”); *see id.*, ¶¶49, 54 (describing communication of data between a device and entities in the network using a network connection).

118. A POSITA would have understood or found obvious that Wright’s WWAN modem, which communicates, via a wireless cellular network, with servers or other devices on the Internet, would *receive and transmit Internet data between*

the device and at least one WWAN. And such communications would happen when the *WWAN modem/device is configured for and connected to the at least one WWAN*, such as when the device/modem connects to the Internet via a wireless network access point “associated with a cellular base station providing network access through” a wireless cellular network. SAMSUNG-1005, ¶¶51-52.

[1.3] one or more processors configured to execute one or more processes in a kernel execution partition, one or more processes in an application execution partition, and one or more processes in a protected execution partition,

119. Wright teaches or suggests that the *wireless end-user device* includes *one or more processors*. SAMSUNG-1005, ¶53. Wright’s disclosed functionality is “implemented as software executing in a mobile computing device having a processor being accessible to memory, the processor being communicatively coupled with one or more network interfaces....” SAMSUNG-1005, ¶53. Indeed, a POSITA would understand that other embodiments of mobile devices and client systems in Wright, e.g., as shown in FIG. 2B and FIG. 8, would similarly include a processor configured to execute software/programming instructions stored in memory. See SAMSUNG-1005, ¶¶53-84.

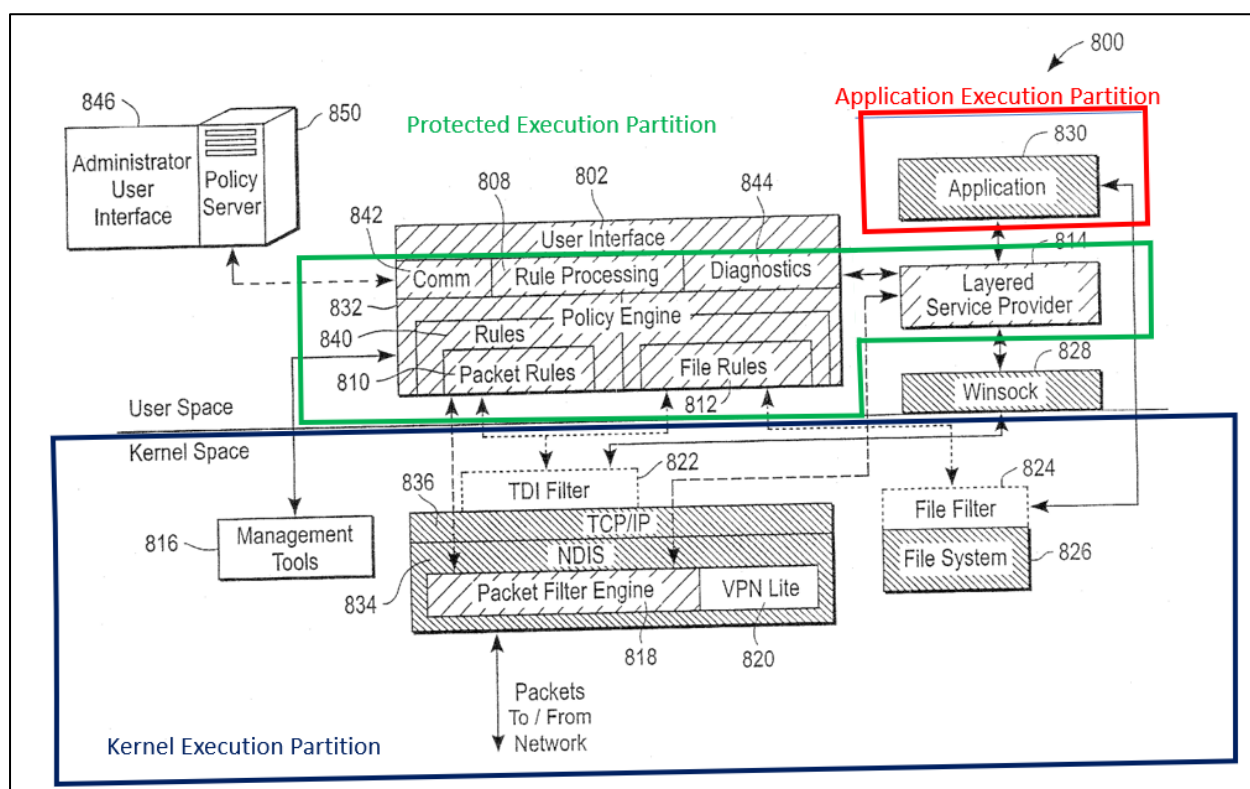
120. Wright further teaches that its *one or more processors* (as described above) are configured to execute *one or more processes* in a *kernel execution partition* (e.g., a kernel space). Wright teaches that its device has a dedicated “kernel

space” that is depicted in FIG. 8 as being partitioned from the user space, thus confirming that its kernel space operates in a separate partition.

121. Wright’s kernel space includes several agents or processes executing therein (*e.g.*, a driver 834, packet filter engine 818, a TDI Filter 822, management tools 816, a file filter 824, file system 826, and VPN Lite 820) that are configured to perform one or more operations (*e.g.*, executing filters, updating parameters, granting/blocking network traffic). *See, e.g.*, SAMSUNG-1005, ¶¶237-240, 265, 281, 296. For example, Wright teaches that “[i]n stateful packet filtering, a packet filter such as the packet filter engine 818 ... filters packets based on a state set by the currently enforced policy with respect to a parameter.” SAMSUNG-1005, ¶265. “[C]omponents such as the policy engine 832, the packet filter engine 818, the layered service provider 814 and the TDI filter 822 may be employed for supporting stateful filtering.” SAMSUNG-1005, ¶268; *see also id.*, ¶281 (“a VPN Lite 820 having a communication interface with the packet filter engine 818 establishes a TLS-encrypted, authenticated connection with the server”); *id.*, ¶296 (“the file filter 824 allows access to 1032 an encrypted copy of the file”); *id.*, ¶243 (“management tools 816 provide information to the policy engine 832 such as the types of adapters connected to the mobile device and specific information about each of them”). Thus, a POSITA would understand Wright’s components

executing in the kernel space are *one or more processes executing in a kernel execution partition*.

122. Moreover, a POSITA would have understood that the system 800 is “a mobile computing device having a processor,” and the processor would be configured to execute instructions for performing one or more of the processes in the kernel space. SAMSUNG-1005, ¶¶53, 63, 228-237.



SAMSUNG-1005, FIG. 8 (annotated)

123. Additionally, as shown in FIG. 8 above, Wright’s device has a “user space” (which is also referred to as “application space”) that includes policy engine 832, layered service provider 814, Winsock 828, and one or more Applications 830.

See SAMSUNG-1005, FIG. 8. Wright teaches that the policy setting module 212 of FIG. 2B includes analogous functionality to that provided by policy engine 832. SAMSUNG-1005, ¶283. Thus, given Wright’s teaching that a “stable execution environment” is “part of the policy setting module 212,” a POSITA would have recognized such an execution environment would be provided by “policy engine 832,” and that this execution environment facilitates communication of “execution instructions” in the form of policies exchanged between the client device and the policy server. *See* SAMSUNG-1005, ¶¶286-287. Wright further teaches secure (e.g., through use of encryption) receipt and management of policies received from the policy server and by the policy engine 832 and the layer services provider 814. SAMSUNG-1005, ¶¶274-283, 296.

124. The ‘144 Patent’ recognizes that a partition encompasses a secure execution environment. *See, e.g.,* SAMSUNG-1001, 3:3-8 (“various secure execution environments for device assisted services are provided using various hardware partition techniques (e.g., secure memory, secure modems, secure memory partition(s) in the CPU/processor), as described herein”). Consistent with this disclosure, a POSITA would have understood or found obvious that the policy engine 832 and the layer services provider are partitioned from other components in the user space (such as, e.g., applications 830). That’s because Wright contemplates an execution environment implemented at the policy engine 832 and utilizes the

policy engine 832 and layered services provider 814 for facilitating the secure communication and management of policy information at the client device. SAMSUNG-1005, ¶137, 169, 22.

125. Additionally, it was well-known in the relevant timeframe to partition environments, particularly when doing so would enable securing sensitive data and functionality of a system. Wright already contemplates the importance of secure handling of policies and ensuring its system is functioning correctly (e.g., has not been corrupted or tampered with). SAMSUNG-1005, ¶138. A POSITA would have therefore been motivated to apply well-known partitioning techniques to Wright's system to ensure secure and proper functioning of its policy storing and management operations. SAMSUNG-1005, ¶138. Partitioning computing environments, including, e.g., using memory partitions, was a known and conventional technique, and it would be predictable to combine such known techniques with known system (such as, e.g., Wright's computing environment) to achieve predictable results of partitioned environments that are secured and separated from operations of other applications in the user space. Further still, a POSITA would have expected reasonable success in so partitioning the user space, given that it was well-known and straightforward to partition environments and Wright's system already contemplates a type of partitioning with respect to its implementation of an execution environment for the above-contemplated secure handling and

management of policies. SAMSUNG-1005, ¶¶228, 237-238; *see, e.g.*, SAMSUNG-1014, 69 (“Programs should have only as much privilege as necessary: the principle of least privilege.”); SAMSUNG-1022, 1283 (“f) Least privilege: Every program and every user of the system should operate using the least set of privileges necessary to complete the job. Primarily, this principle limits the damage that can result from an accident or error. It also reduces the number of potential interactions among privileged programs to the minimum for correct operation, so that unintentional, unwanted, or improper uses of privilege are less likely to occur. Thus, if a question arises related to misuse of a privilege, the number of programs that must be audited is minimized. Put another way, if a mechanism can provide "firewalls," the principle of least privilege provides a rationale for where to install the firewalls. The military security rule of "need-to-know" is an example of this principle.”); SAMSUNG-1015, 41, 108 (“Each process has its own address space, that defines the memory addresses that it can access;” and “The Solaris operating system implements a system of privileges that provide fine-grained control over the actions of processes.”) A POSITA would recognize from these disclosures that not only are processes axiomatically partitioned in computing system (via their own virtual address space), but they can be partitioned across privilege levels as well (which is enforced by the operating system).

126. Thus, based on the above, a POSITA would have understood or found obvious that Wright's user space would be partitioned such that the (1) policy engine 832 and layer services manager would be in a secure/protected execution partition (i.e., ***protected execution partition***) and (2) applications 830 would be implemented in a separate partition, i.e., ***application execution partition***. See, e.g., SAMSUNG-1015, 41 ("A protection ring boundary isolates the UNIX kernel from the processes. Each process has its own address space, that defines the memory addresses that it can access.").

127. Additionally, applications 830 (e.g., browsers, email applications) would be understood as ***one or more processes in an application execution partition***. SAMSUNG-1005, ¶¶187, 239. A POSITA understood that a process includes programs or instructions being executed by a processor/CPU; applications are programs that, when executed, are computing processes. SAMSUNG-1040, 46; SAMSUNG-1041, 5:41-61. Additionally, a POSITA understood or found obvious that user interface module would be implemented in the application execution partition for the reasons explained in claim 18. See my analysis at ¶¶258-264; see *supra* [18]. Similarly, a POSITA understood that LSP 814 and the policy engine 832 are processes executed by the processor that perform the operations and functionality (i.e., policy setting and management) of these components, as described in Wright. SAMSUNG-1005, ¶¶237-243; see, e.g., SAMSUNG-1015, 41

(“ Each process has its own address space, that defines the memory addresses that it can access. “); SAMSUNG-1015, 27 (“The supervisor is isolated from other processes by protection rings”); SAMSUNG-1015, 29; SAMSUNG-1022, 1284 (“Each partition contains a collection of information, all of which is intended to be protected uniformly.”).

[1.4] the one or more processes executed in the kernel execution partition including an operating system packet network stack to pass Internet data packet traffic between the WWAN modem and one or more applications executing in the application execution partition,

128. Wright teaches or renders obvious that *Internet data packet traffic* passes between the *WWAN modem* and one or more applications executing in the *application execution partition*. See, e.g., SAMSUNG-1005, ¶¶73, 231-232, 281-282, 293. As explained above, Wright’s applications (executing in the application execution partition) can send/receive data in the form of network packets over the Internet, and such transmissions would happen from the respective applications via the NIC/NAP of the mobile device (i.e., *WWAN modem*, as explained in [1.2] *supra*). See *id.*; see also SAMSUNG-1005, ¶¶282 (enabling a TLS/SSL socket for client-server communications to/from device applications). For example, Wright teaches that an application (e.g., an email application) can send data over a network, such as a file (e.g., email, text message, TIFF file, or a Word document) or can use

a browser to access a website of an Internet shopping site. SAMSUNG-1005, ¶¶51, 73, 187.

129. Wright further contemplates that network traffic passes between the NIC and applications, given its teaching of “connection type [being] indicated by the association of a local IP address with data communicated over a wireless or wired network adapter or network interface card (NIC)” SAMSUNG-1005, ¶73.

130. Moreover, as explained above for [1.3], Wright’s kernel space and associated processes therein correspond to the *one or more processes executed in the kernel execution partition*.

131. Wright’s kernel space includes *an operating system packet network stack*. In Wright, the mobile device “processes” incoming and outgoing “network traffic” devices using “information in one or more layers of a model for communications between computers in a network,” such as, e.g., the seven-layer OSI data communication model or the four-layer Internet Protocol Stack. SAMSUNG-1005, ¶¶67, 220, 229 (teaching “model for communications between computer in a network,” including OSI data communication model). A POSITA would have understood that such multi-layer network communication models are network stacks (or are implemented as network stacks on computing devices, such as Wright’s mobile device). *See, e.g.*, SAMSUNG-1016, 20-21 (“The OSI protocols developed slowly, and because running the full protocol stack is resource intensive,

they have not been widely deployed, especially in the desktop and small computer market. In the meantime, TCP/IP and the Internet were developing rapidly, with deployment occurring at a very high rate.”); SAMSUNG-1016, 6.

132. Moreover, a POSITA understood or would find obvious that this network stack is typically implemented using drivers, components, and associated functionality in the kernel. For example, in Wright, a Network Driver Interface Specification (NDIS) driver “operat[es] within the kernel of the operating system of the mobile device” and enables network communications. SAMSUNG-1005, ¶¶228, 237; *see id.*, ¶239 (“explaining that “network parameters” are obtained by “the NDIS driver for OSI layers 2-5”). Wright additionally teaches that “[n]etwork packets from the network interface card (NIC) are received by a Miniport Driver” and a transport driver determines information associated with the OSI model’s transport layer. SAMSUNG-1005, ¶229. A POSITA understood that a miniport driver manages interactions between a NIC and the operating system. *See, e.g.*, SAMSUNG-1034, 8 (“Network interface cards use an “NDIS miniport driver,” which relies exclusively on a special API exported by NDIS.SYS or NDIS.VXD.”) Wright additionally teaches that the filter engine 818 in the kernel can “act[] as a layer manager in processing parameters at the network layer in the OSI or IPS models” with respect to “network packets” received from the NIC. SAMSUNG-1005, ¶229.

133. A POSITA additionally understood that the above-described Internet packet network traffic between the WWAN modem and the applications would pass through the above-described ***operating system packet network stack*** to facilitate the network communications contemplated in Wright. See, e.g., SAMSUNG-1016, 6 (“By dividing the communication software into layers, the protocol stack allows for division of labor, ease of implementation and code testing, and the ability to develop alternative layer implementations. Layers communicate with those above and below via concise interfaces. In this regard, a layer provides a service for the layer directly above it and makes use of services provided by the layer directly below it. For example, the IP layer provides the ability to transfer data from one host to another without any guarantee to reliable delivery or duplicate suppression.”); see also SAMSUNG-1017, Abstract, 3.4; SAMSUNG-1024, 2, 10.

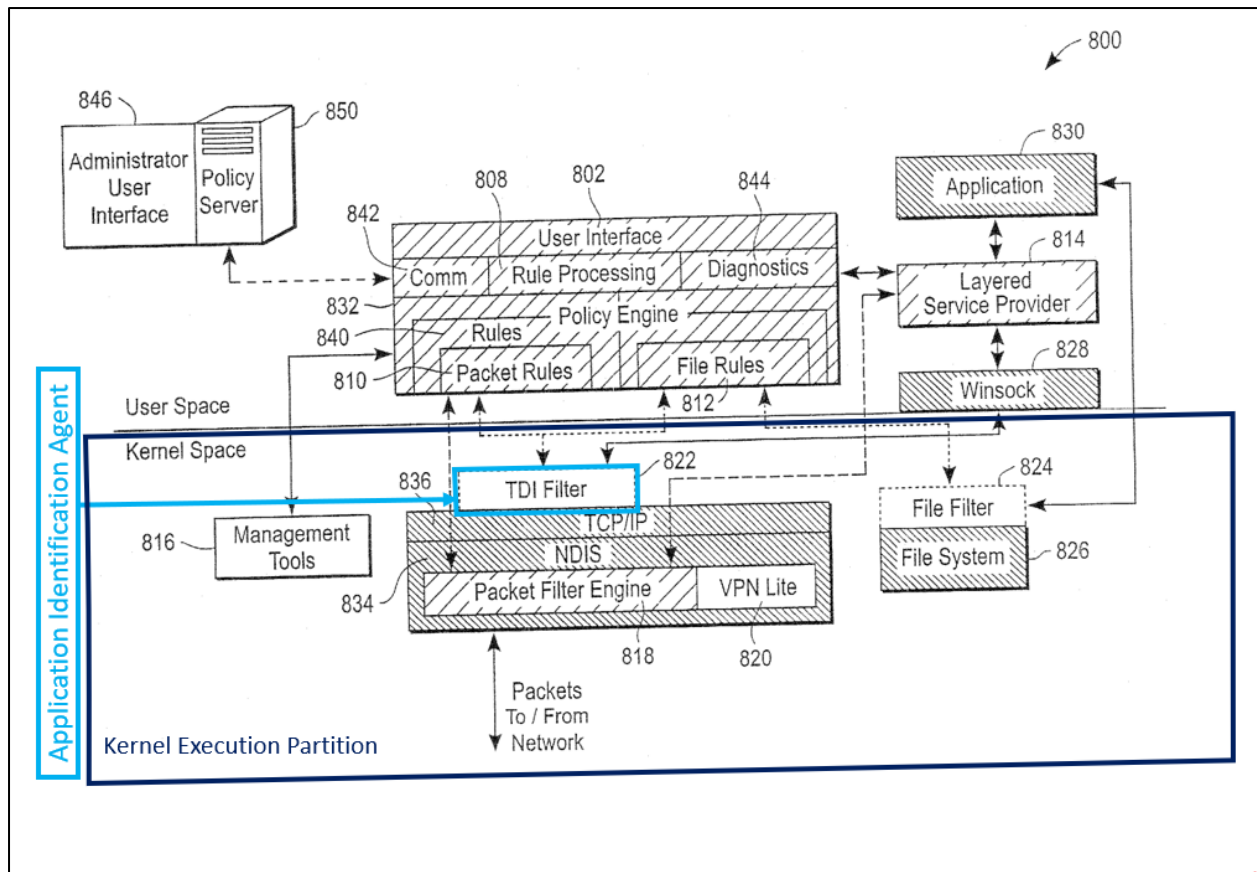
134. In this manner, a POSITA understood that the ***operating system packet network stack*** (e.g., drivers, components, and associated functionality implementing the OSI/IPS network model) is configured ***to pass Internet data packet traffic*** (i.e., data related to “a file,” “an e-mail, a text message, a TIFF file, or a Word document”) ***between the WWAN modem and one or more applications*** (e.g., an application 830 handling such data) ***executing in the application execution partition***. SAMSUNG-1005, ¶187.

[1.5] the one or more processes executed in the kernel execution partition including ... an application identification agent to classify individual flows of the Internet data packet traffic passing through the stack according to which of the one or more applications is associated with a particular individual flow,

135. As explained above, Wright teaches that its ***kernel execution partition*** (e.g., kernel space) is configured to execute ***one or more processes*** (e.g., the driver 834, packet filter engine 818, a TDI Filter 822, management tools 816, a file filter 824, file system 826, and VPN Lite 820). *See supra*, [1.3].

136. A POSITA understood that the application/TDI filter 822 in the kernel space corresponds to the claimed application identification agent that classifies individual flows of the Internet traffic through the stack according to which of the one or more applications is associated with a particular individual flow, for the reasons described below.

137. As a starting point, application/TDI filter 822 corresponds to the ***application identification agent***. SAMSUNG-1005, ¶238.



SAMSUNG-1005, FIG. 8 (annotated)

138. TDI filter 822 is configured to apply multiple classifications of packets / traffic passing therethrough. Wright discloses that “[t]he NDIS filter driver” will act on the low level information *i.e.* TCP or UDP port numbers” and “[w]hen the packet arrives at the TDI layer 822, the TDI filter driver 822 determines based on one or more application parameters for which ... application a packet is destined and if the packet should be forwarded or filtered.” SAMSUNG-1005, ¶241. Additionally, Wright’s TDI filter 822 perform application-specific filtering on the individual packets associated with particular applications. For example, rules

determined by policy engine 832 “are passed to ... the TDI filter 822 for application specific filtering.” SAMSUNG-1005, ¶269; *see also id.*, ¶297 (“TDI filter 822 also monitors application parameters and allows the traffic associated with this application to be forwarded for further processing by the mobile device”). TDI Filter 822 thus serves as an “enforcement mechanism[,]” similar to the enforcement control module 214” of FIG. 2, which performs “stateful filtering.” SAMSUNG-1005, ¶¶264, 283; *see id.*, ¶¶264-265, ¶266 (“adaptive port blocking”), ¶¶268-271.

139. By determining which application corresponds to a particular packet traffic and applying application-specific policy rules on those packets, a POSITA understood or would find obvious that TDI filter 822 is an ***application identification agent*** included ***in the kernel execution partition*** that ***classifies individual flows*** of packets or traffic ***according to which of the one or more applications is associated with a particular individual flow***.

140. Moreover, the traffic flow relates to the Internet data traffic between the WWAN modem and the one or more applications, which a POSITA understood would implicate ***traffic passing through the OS network stack*** (which, as explained in [1.3] facilitates such network communications. SAMSUNG-1005, ¶¶215, 229, 267, 283, 67 and 237 (managing network communications at different layers of OSI/ISP model), 297 (“TDI filter 822 also monitors application parameters to block any network traffic at the upper layers for the prohibited application that may have

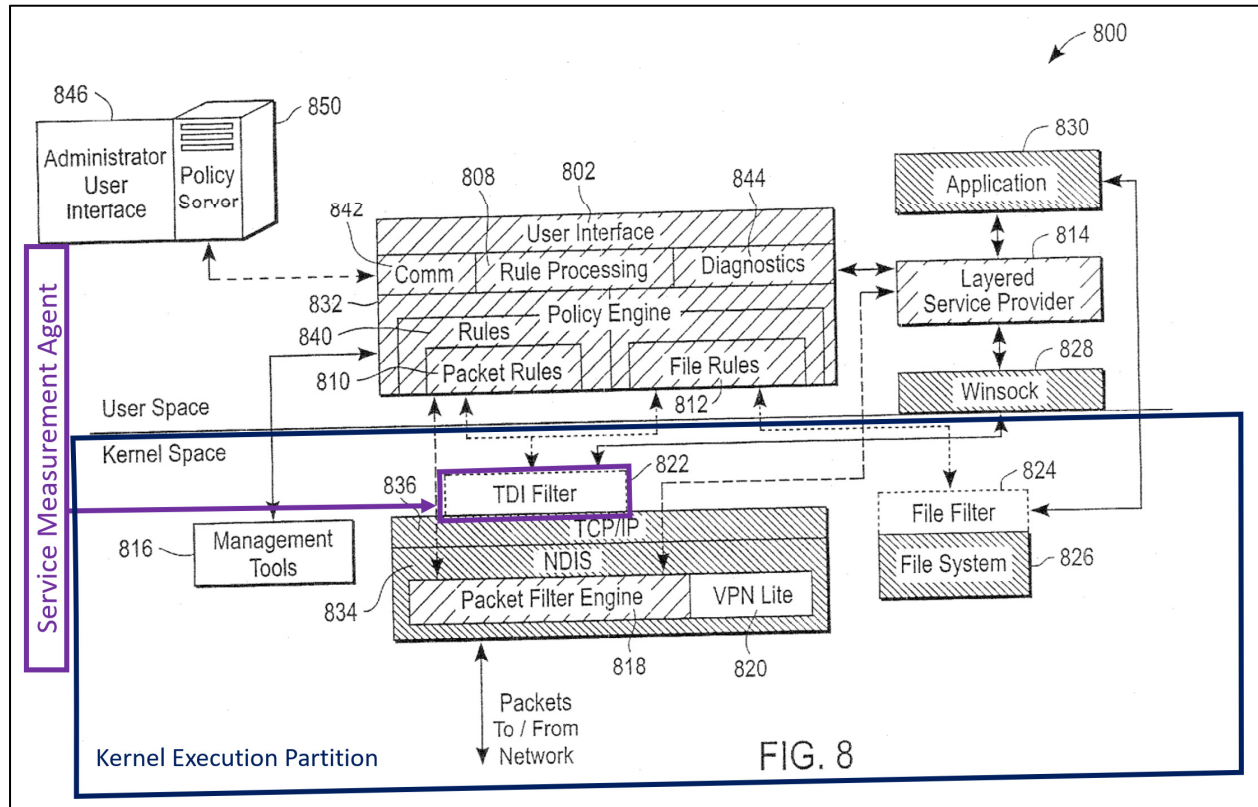
bypassed the packet filter engine 818.”). Thus, a POSITA would have found obvious that Wright’s network packets or traffic that is application-specific *passes through the stack*. See *supra*, [1.4].

[1.6] the one or more processes executed in the kernel execution partition including ... a service measurement agent to measure, for classified individual flows, an amount of the Internet data packet traffic associated with each of the one or more applications, and

141. As explained in [1.3] *supra*, Wright teaches one or more processes, which include the TDI filter 822 and the packet filter engine 818, which are executed in the kernel execution partition.

142. Additionally, TDI filter 822 and/or packet filter engine 818 perform “stateful filtering,” which Wright also refers to as “stateful packet filtering” when “filtering is performed on a packet basis.” SAMSUNG-1005, ¶¶265, 268. Such “[s]tateful filtering” may be “based on application parameters,” which provides “the benefit of more refined application filtering.” *Id.* In some implementations, the “TDI filter 822 also monitors application parameters and allows the traffic associated with this application to be forwarded for further processing by the mobile device” and in some instances, “block[s] any network traffic at the upper layers for the prohibited application that may have bypassed the packet filter engine 818.” SAMSUNG-1005, ¶297; see *id.*, ¶193 (explaining that “application parameter”

includes, e.g., “application identifier or information derived by an application”),
 ¶199 (location is an application parameter).



SAMSUNG-1005, FIG. 8 (annotated)

143. Wright further contemplates “peer to peer” communication in which files are exchanged between devices, or be downloaded from the server during a communication session. *See* SAMSUNG-1005, ¶¶135-181; FIGS. 3A-3F. In one example, Wright explains that policies defined for such communications can control the “type, size, format, and other attributes of transferred files.” *See* SAMSUNG-1005, ¶169. Moreover, Wright contemplates such file attribute-based policies being applicable for email applications. *See id.* (“Files transfers and/or user-to-user

communications may be subject to other policies.... For email between users who are on the same corporate domain, email APIs may be used to redirect email and attachments through an encrypted channel to the policy server to the corporate mail server.”); *see also id.* at ¶163 (“Use the “software update size” or “policy size” to calculate an estimated download time.”); *see also id.*, ¶158-162. In this regard, a POSITA would have understood that attributes of files being transferred by an application is an application parameter at least because such information would be derived by the application layer of the network stack/OSI model (e.g., when an application indicates that it is sending such files over a network). SAMSNG-1005, ¶193 (“application parameter [is] typically associated with the application layer in the OSI or IPS models”).

144. Given the above disclosures, a POSITA understood that Wright contemplates application-specific filtering of traffic, which can be dictated by policies based on application parameters such as size of packet traffic (e.g., size of file being transmitted) by the application.

145. For Wright’s system to perform packet filtering based on traffic size (e.g., size of a file or attachment to be sent over a network), a POSITA would have understood that the system would measure the file/traffic size, or at least would have found it obvious to do so. Indeed, in the relevant timeframe, it was well-known to perform application-specific packet filtering at least in part by measuring/monitoring

network packet traffic to/from the application. Indeed, conventional email and file transfer application for which policies controlled traffic transmission exceeding certain amount of network or service usage. And such packet traffic/file size measuring functionality was conventionally implemented as part of kernel processes using techniques such as Deep Packet Inspection (DPI). DPI is a form of (most frequently) in-kernel data processing that examines the contents of data packets transmitted over a network. For example, McCanne teaches “Because network monitors run as user-level processes, packets must be copied across the kernel/user-space protection boundary. This copying can be minimized by deploying a kernel agent called a packet filter, which discards unwanted packets as early as possible.” SAMSUNG-1017, 1. The data packet payloads are then analyzed to determine application state. Unlike traditional packet filtering, DPI enables the identification, classification, and control of application traffic such as the security policies of Wright.

146. More broadly, a POSITA would know that the inspection of data for the purposes of implementing policy at the kernel level was widely known and deployed at the crucial date. For example, Takeuchi discloses a network packet monitoring unit implemented in the kernel that measures and analyzes network packets. SAMSUNG-1009, ¶69. As another example, Oester describes a detection device in operating system that measure “resource consumption” of applications.

SAMSUNG-1010, ¶¶1, 10, 18, 21, 30-35. As another example, Wang discloses a system that “performs application-message-based inspection rather than packet-level inspection” and that the system “operates at the application protocol layer, above the transport layer.” SAMSUNG-1024, 2.

147. Thus, implementing a network traffic measuring unit to measure size of network traffic would have been an obvious, conventional way to implement what Wright describes, and is nothing more than utilizing familiar, known techniques to achieve a predictable result of application-specific filtering based on packet size, as already contemplated by Wright. Alone or in combination, Wright’s TDI filter 822 and filter engine 818 provide an obvious kernel implementation of measuring an amount of the Internet data packet traffic associated with each of Wright’s one or more applications, e.g., applications 830. While there may be other techniques for measuring a file size, a POSITA would understand certain advantages for measuring by the TDI filter 822 / packet filter engine 818. *See* SAMSUNG-1017, Abstract (“Because network monitors run as user-level processes, packets must be copied across the kernel/user-space protection boundary. This copying can be minimized by deploying a kernel agent called a *packet filter*, which discards unwanted packets as early as possible.”); see also SAMSUNG-1017, §3.4 (“This next filter accepts all IP packets which did not originate from two particular IP networks, 128.3.112 or 128.3.254.”); SAMSUNG-1024, 2, 10. A POSITA would understand McCanne and

Wang, at least in part, as describing techniques and devices for packet measurement. SAMSUNG-1017, Abstract, §§2.2, 3.4, 3.6; SAMSUNG-1024, 2 (“...performs application-message-based inspection rather than packet-level inspection;” “Shield operates at the application protocol layer, above the transport layer.”), 4, 5, 10 (“well-designed kernel implementation of Shield”), 11.

148. For example, a POSITA would understand that any traffic measurement done outside the kernel space is less secure and by contrast, measuring traffic (e.g., at the TDI filter 822 or packet filter engine 818) would have provided a more secure measurement point in the kernel space and would further enable these components’ enforcement of policies. *See, e.g.*, SAMSUNG-1005, ¶239; *see also id.*, 167 (“, a software application on the client mobile device processes all incoming file and email transfers,” and “...calculates signatures of each file received and forwards these to a centralized server (e.g. corporate server).”). Moreover, a POSITA would have been motivated and would have had a reasonable expectation of success to implement this functionality as part of the TDI Filter 822 / Packet Filter 818 at least because (1) these are kernel processes (which is a conventional obvious place to implement such functionality, per Takeuchi), (2) these processes are already tasked with application-specific filtering of network traffic based on traffic/packet size, and (3) traffic size measurements would enable and further Wright’s goal of more “refined” application filtering, which would be able to implement refined traffic

filtering based on size or volume (rather than just allow or block all traffic). See, e.g., SAMSUNG-1005, ¶¶242, 268-269.

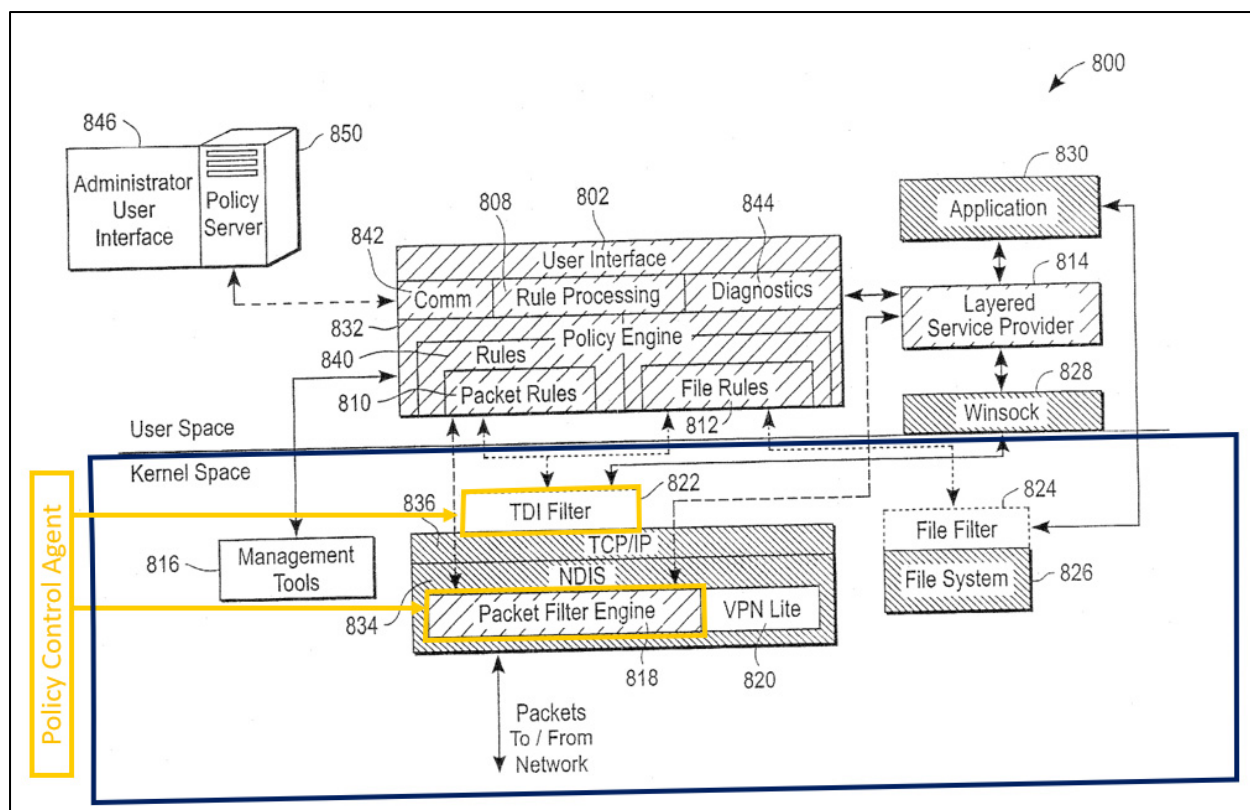
149. In summary, Wright in view of conventional knowledge of a POSITA teaches or renders obvious implementing *a service measurement agent* (e.g., TDI Filter and/or packet filter that would include packet size measurement functionality) executing in the kernel execution partition *that measures, for classified individual flows, an amount of the Internet data packet traffic associated with each of the one or more applications.*

[1.7] the one or more processes executed in the kernel execution partition including ... a policy control agent to apply application-specific traffic policy controls to classified individual flows,

150. As explained above, Wright's kernel space (*kernel execution partition*) is configured to execute one or more processes, such as packet filter engine 818 and TDI filter 822 (*see supra* [1.3]) and the TDI filter classifies individual flows of Internet packet traffic associated with applications (*see supra* [1.5]).

151. A POSITA understood or would find obvious that the TDI filter and packet filter engine correspond to the claimed *policy control agent that appl[ies] application-specific traffic policy controls to classified individual flows*, as explained below.

152. As explained in [1.5], TDI filter 822 and the packet engine 818 perform application-specific filtering using rules 840 from the policy engine 832. With reference to FIG. 8, which describes aspects of stateful filtering that can be implemented by the packet filter engine 818 and TDI filter 822, Wright teaches that policy filtering happens in one of three modes as they to relate ports over which traffic is sent or received: “open, closed, or stateful.” SAMSUNG-1005, ¶267; see also *id.*, ¶¶263-266, 268-274. As Wright explains, all traffic flows when port is open; all traffic is blocked when port is closed; some traffic is permitted and some blocked when port is stateful. “[S]tateful filtering [is] performed by the packet filter engine 818 and/or the TDI filter 822 in accordance with rules 840, for example rules in the subset of the packet rules 810, applicable to the current policy....” SAMSUNG-1005, ¶268; *see id.*, ¶297, ¶231 (in context of the FIG. 7 embodiment, policy rule engine 704 (which is akin to policy rule engine 832 of FIG. 8) provides “different rules with respect to inbound packets than for outbound packets”).



SAMSUNG-1005, FIG. 8 (annotated)

153. In some implementations, packets, which have been classified previously and filtered at the TDI filter 822, are forwarded to the packet filter engine 818 (see SAMSUNG-1005, ¶297), which performs “stateful packet filtering” to “filters packets based on a state set by the currently enforced policy with respect to a parameter.” SAMSUNG-1005, ¶265. Wright also explains that the rules, from the policy engine, are additionally “passed to ... the TDI filter 822 for application specific filtering.” SAMSUNG-1005, ¶269; see FIG. 8 (showing packet rules from rules 840 being provided to TDI Filter 822 and Packet Filter Engine 818 for packet filtering).

154. The TDI filter 822 filters the upper layers of network traffic, e.g., by “monitor[ing] application parameters” for applications for which packets are being transmitted “to block any network traffic at the upper layers for the prohibited application that may have bypassed the packet filter engine 818.” See SAMSUNG-1005, ¶297. Thus, in view of the above, a POSITA understood or would find obvious that, the packets for which the TDI filter 822 applies application-specific filtering (*applying application-specific traffic policy controls*) to classified individual flows of Internet data packet traffic associated with applications (as explained in [1.5] above).

155. Wright’s application-specific traffic policy controls include application-specific traffic rules or policies. See, e.g., SAMSUNG-1005, ¶¶169, 239, 269. For example, the application-specific traffic policy controls can be rules or policies for or related to application-specific filtering or application-specific network access. See, e.g., *id.*, ¶¶238-239, 269. The application-specific traffic policy controls can also include application-specific rules or policies as they relate to attributes of a file or object, such as size, type, etc. See SAMSUNG-1005, ¶169. These rules and policies have permissions that enable modifications to Wright’s *application-specific traffic policy controls*. See, e.g., SAMSUNG-1005, ¶85; see also *id.*, ¶¶55, 68, 72, 85-109, 158-163, 218.

156. Additionally, a POSITA would understand or find it obvious that the packet engine 818, embedded in the NDIS driver 834, handles lower layer filtering/processing of network packets. Per Wright, “network environment location detection is performed by the policy engine 832, in accordance with rules implementing one or more location detection tests in the Rules set 840, based on network parameters obtained by-the NDIS driver [which is part of packet engine 818] for OSI layers 2-5....” SAMSUNG-1005, ¶239. As also noted in the analysis of Elements [1.5]-[1.6] above, a POSITA would understand or find obvious that the TDI filter 822 classifies and measures individual flows of Internet data packet traffic. Thus, in view of the above, a POSITA understood or would find obvious that, the packets for which the TDI filter 822 classifies and measures individual flows of Internet data packet traffic, are provided for application-specific filtering (*applying application-specific policy controls to classified individual flows*) at packet engine filter 818.

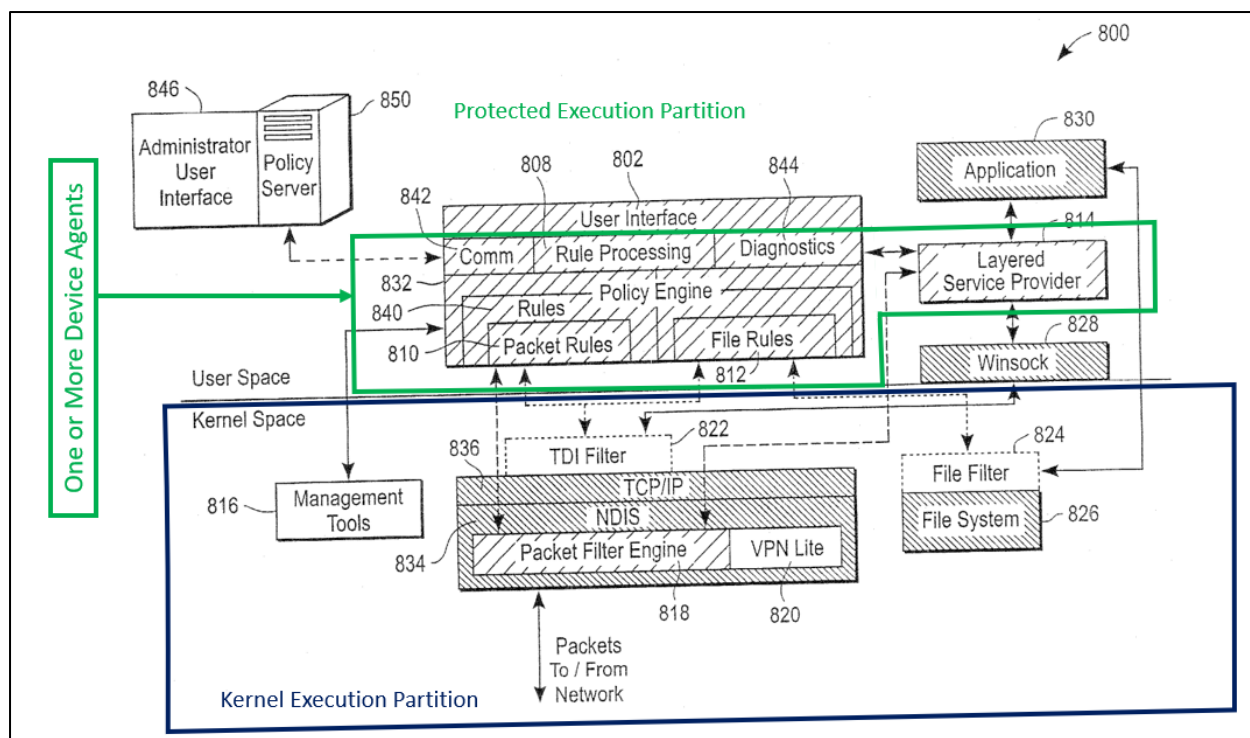
[1.8] the one or more processes executed in the protected execution partition including one or more device agents with limited privileges to access processes executing in the kernel execution partition, including

157. As explained in [1.3] above, the *protected execution partition (PEP)* in Wright includes LSP 814 and policy engine 832.

158. Each of these PEP components are *one or more device agents* because they are resident on Wright’s end-user device and are software agents that are configured to perform specific functions. *See, e.g.,* SAMSUNG-1029, 3 (providing dictionary definition of “agent,” as “[a]n autonomous system that receives information from its environment, processes it, and performs actions on that environment.” such as a “[s]oftware agents [that] operate in symbolic environments, and perceive and act upon strings of symbols;” explaining that software agents are programs that perform certain tasks, much like the components of policy engine 832 and LSP 814; Downing defines an agent as “a piece of software that performs a service... [f]or example, an agent might run on a client computer to keep the server informed of its needs;” SAMSUNG-1030, 3”);. Based on these descriptions and definitions, a POSITA understood that software agents are programs that perform certain tasks, much like the components of policy engine 832 and LSP 814. . SAMSUNG-1005, ¶¶236-281, 296 FIG. 8.

159. A POSITA would have understood or found obvious that each of the LSP 814 and policy engine 832 have *privileges to access processes executing in the kernel execution partition*. For example, LSP 814 has privileges to access the packet filter engine in the kernel space. As Wright explains, LSP 814 “diverts the outgoing traffic from the application to a VPN client piece [in the kernel space], and incoming traffic from the VPN client piece to the application.” SAMSUNG-1005,

¶281, ¶238 (Winsock 828 has a communication interface with the TDI filter 822 in the kernel space). Additionally, LSP 814 “captures information about network applications starting and stopping and what ports the applications will be using.” SAMSUNG-1005, ¶239; *see id.* (“LSP 814 “is a windows socket filter [that] is used to determine which application (e.g. browser e-mail application such as Outlook Exchange®) is accessing the network and what networking services the application will be using.”). “An example of [Wright’s] application’s network service usage could include Outlook Express opening a specific set of Winsock ports, each using a separate protocol and target I.P. address. The layered service provider 814 will pass this information to the packet filter engine 818, which then informs the policy engine 832 using an event signaling mechanism.” SAMSUNG-1005, ¶239. LSP 814 has “a communication interface” with the packet filter engine 818 to facilitate the above described communications between these components. SAMSUNG-1005, ¶237.



SAMSUNG-1005, FIG. 8 (annotated)

160. With respect to policy engine 832, Wright explains that this component has a “communication interface[s]” to processes in the kernel space, such as (1) “management tools 816 of the operating system” that provide information about running applications/processes and types of adapters connected to the device, (2) “file filter 824 operating in the kernel space that controls access to the file system 826,” and (3) “packet filter engine 818 operating within a driver 834” to which rules are passed, from rules 840, to perform, e.g., stateful packet filtering. See SAMSUNG-1005, ¶237, 243, 268-269.

161. A POSITA would understand or find obvious that the above-described *one or more device agents have limited privileges to access processes executing in*

the kernel execution environment. In general, a POSITA understood that the kernel is responsible for managing system resources, memory, and drivers for the mobile device, and that processes in the user space can only access processes executing in the kernel if they have permission to do so. Additionally, it is well-known, in the context of operating systems on computing devices, for processes in the user space (such as LSP 814 and policy engine 832) to have limited privileges to processes in the kernel space, because the limiting of privileges is required to ensure the security and stability of the overall system. Indeed, the structure of the operating system and the underlying hardware is built on the limiting of privileges of the user space, e.g., where the kernel runs at hardware RING0 (which is the highest privilege and all user space processes run at RING3: “The security kernel runs in the most privileged protection ring, ring 0, and user software runs in the least privileged ring, ring 3.” SAMSUNG-1015, 79.

162. In view of the above, and when considering Wright’s disclosure of only limited kernel processes with which LSP 814 and policy engine 832 communicate (as described above), a POSITA would understand or find obvious that LSP 814 and policy engine 832 have ***limited privileges*** to access processes executing in the kernel execution environment. *See, e.g.,* SAMSUNG-1015, 34 (“Some ring 0 code must respond to calls from untrusted processes (e.g., system calls). The only way that ring 0 can be accessed by an untrusted process is via a gate. As described above, gates

check the format of the arguments to higher-privileged, supervisor code to block malicious inputs. Thus, if the gates are correct, then untrusted processes cannot compromise any ring 0 code, thus protecting the supervisor.”); SAMSUNG-1015, 17 (“The trivial approach is to mediate all system calls, as these are the entry points from user-level processes.”).

163. For the above reasons, a POSITA would understand or find obvious that Wright’s *one or more device agents* (e.g., LSP 814, policy control engine 832) have *limited privileges to access processes executing in the kernel execution partition* (e.g., kernel space).

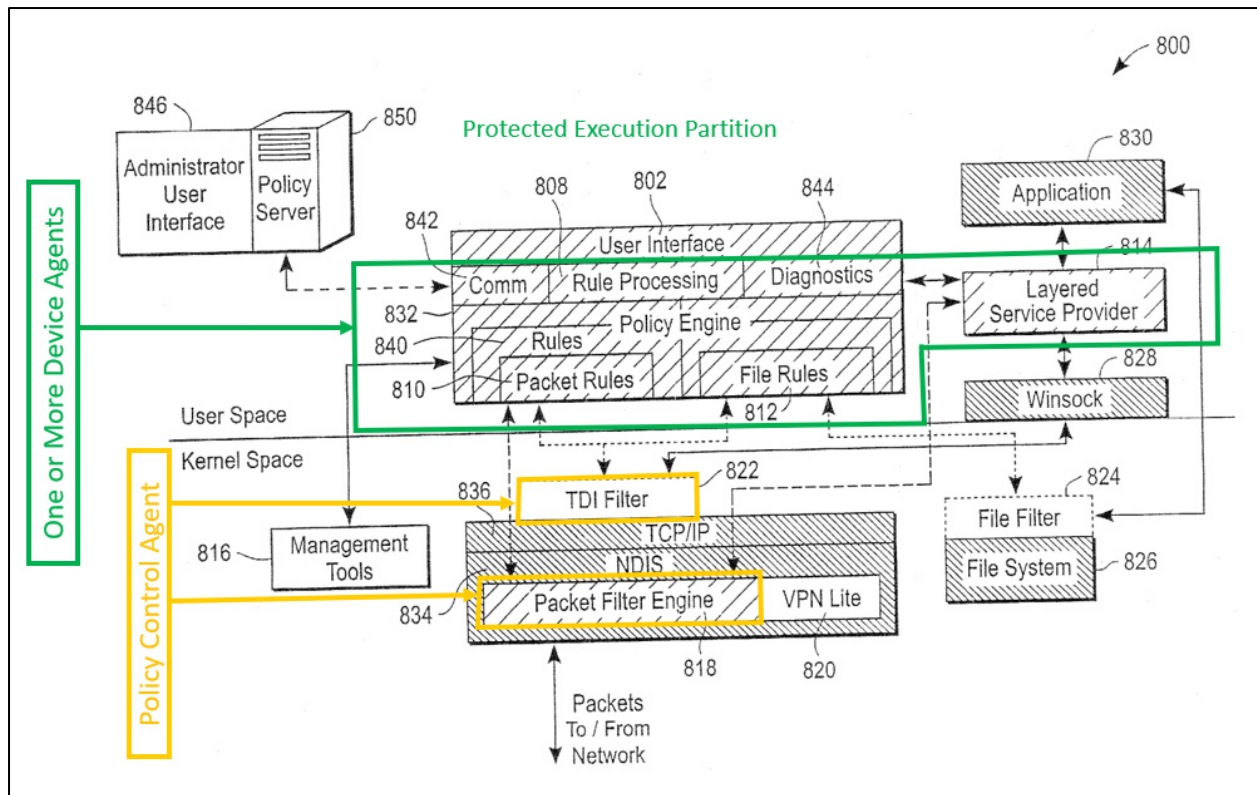
[1.9] One or more device agents ... including a privilege to configure the application-specific traffic policy controls of the policy control agent, and

164. Further to the analysis above in [1.8]. Wright teaches or suggests that one of the device agents (e.g., LSP, policy engine) has *a privilege to configure the application specific traffic policy controls of policy control agent*.

165. As explained below, Wright’s policy engine 832 (i.e., *device agent in the PEP*) has a privilege to configure *application-specific traffic policy controls* of packet filter engine and/or TDI filter (i.e., *policy control agent*). *See supra* [1.7]. Wright discloses that rules 840 component of policy engine 832 “define[s] one or more security policies” (SAMSUNG-1005, ¶245) and that “stateful filtering,” which is “performed by the packet filter engine 818 and/or the TDI filter 822 in accordance

with rules 840” (e.g., rules in packet rules 810 that may be “applicable to the current policy”). SAMSUNG-1005, ¶268 (further noting that “forward and filter decisions” are made in “accordance with rules in the set of rules 840 or the subset of the packet rules 810). In other words, “[t]hese rules [840, 810] are passed to the packet filter engine [818] as well as the TDI filter 822 for application specific filtering.” SAMSUNG-1005, ¶269.

166. Wright further teaches, in the context of the embodiment described and depicted in FIG. 2, that the “the policy setting module 212 receives, installs and updates the security information including security policies and/or software updates received from the policy management module 236 via the policy distribution module 234 over the network connection 204.” SAMSUNG-1005, ¶68. Similarly, in the context of FIG. 8, Wright teaches that “policy engine 832 performs implementation examples of functions ... of determining policies analogous to those of the policy setting module 212” of FIG. 2 (SAMSUNG-1005, ¶283), and that policy engine “download[s]” policies “from the policy server 850.” SAMSUNG-1005, ¶¶244, 286-287. Given these disclosures, a POSITA would recognize that the functionality of policy setting module would be implemented as part of the policy engine 832 and would be used to configure (e.g., update and store) the policies in rules 840.



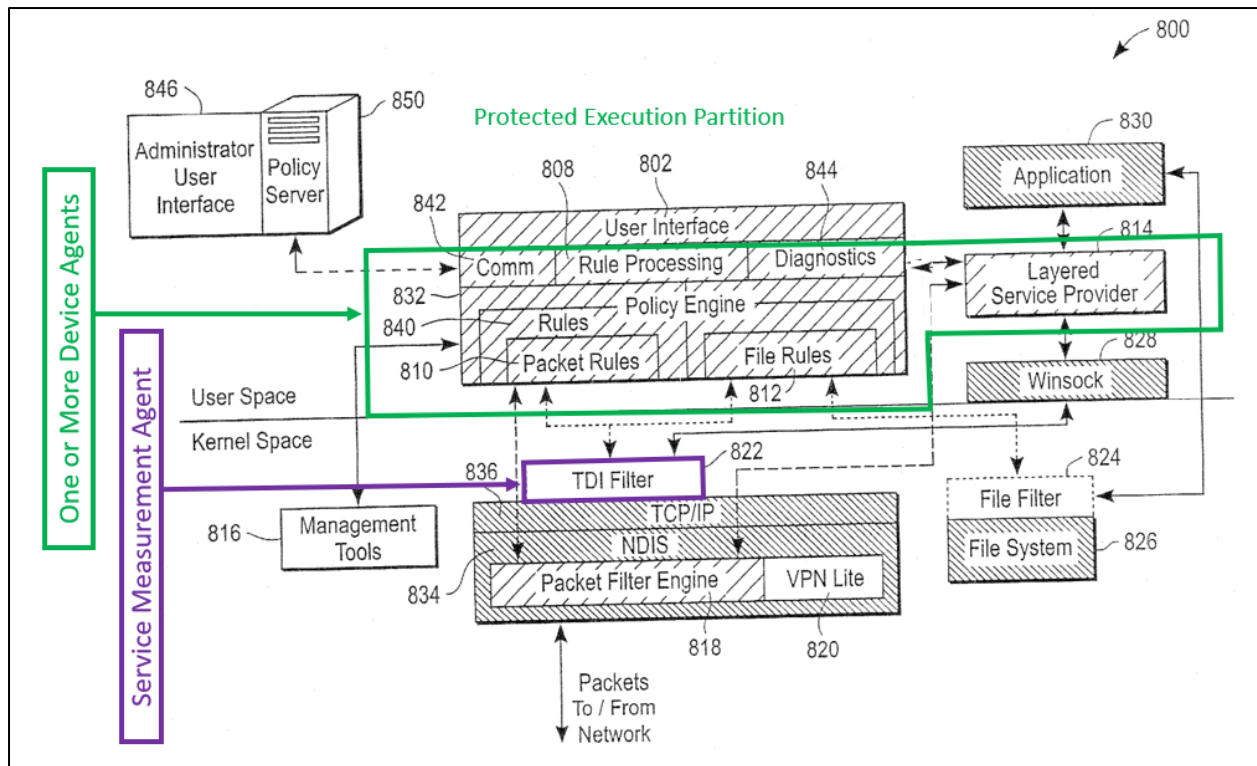
SAMSUNG-1005, FIG. 8 (annotated)

167. Thus, a POSITA would have understood or found obvious that Wright's policy engine 832 (*one or more device agents*) includes a rules 840 component that is configured with policies/rules used for application-specific filtering or network access (i.e., *a privilege to configure application-specific traffic policy controls*) performed by the packet filter engine 818 and TDI filter 822 (*policy control agent*). SAMSUNG-1005, ¶¶238, 245, 268.

[1.10] *One or more device agents ... including ... a privilege to receive at least one of Internet data packet traffic passing between the operating system packet network stack and the WWAN modem, and traffic information including the*

measured amount of the Internet data packet traffic associated with each of the one or more applications.

168. This limitation recites “a privilege to receive *at least one of* Internet data packet traffic... *and* traffic information.” This limitation would thus be satisfied by a showing that the claimed “one or more device agents” has a privilege to receive the “Internet data packet traffic...,” “traffic information including the measured amount,” or both. In other words, the present element is satisfied by showing the device agent’s privilege to either receive the “Internet data packet traffic...” or “traffic information.” Nevertheless, even if the claim is interpreted to require that the agent has the privilege to receive both “Internet data packet traffic” and “traffic information,” that is disclosed or rendered obvious by Wright’s teachings, as further explained below.



SAMSUNG-1005, FIG. 8 (annotated)

169. **First**, Wright teaches or renders obvious that the one or more device agents include a privilege to receive at least one of Internet data packet traffic passing between the operating system packet network stack and the WWAN modem. As explained above, Wright teaches or renders obvious that the protected execution partition (PEP) includes one or more device agents (e.g., LSP 814 and policy engine 832) (see supra [1.8]), and that Internet data packet traffic passes between the operating system packet network stack and the WWAN modem (see supra, analysis of [1.4]).

170. LSP 814 (i.e., one or more device agents) has a *privilege to receive Internet data packet traffic*. Per Wright, LSP 814 has a communication interface with one or more applications 830, obtains “information about network applications starting and stopping and what ports the applications will be using,” and passes “this information to the packet filter engine 818, which then informs the policy engine 832 using an event signaling mechanism.” SAMSUNG-1005, ¶¶228, 239-240. LSP 814 also “diverts the outgoing traffic from the application to a VPN client piece, and incoming traffic from the VPN client piece to the application.” SAMSUNG-1005, ¶281.

171. Wright further explains that LSP 814 provides “implementation examples of functionality analogous to that of layer manager 206” (SAMSUNG-1005, ¶283), which Wright explains is responsible for “process[ing] network traffic which may include inbound data ... accessible over a network and outbound copies of data objects from resident memory 220.” SAMSUNG-1005, ¶67. As explained in the context of elements [1.2]-[1.3], such “information,” e.g., ingoing and outgoing traffic, would be understood to correspond to the *Internet data packet traffic* (e.g., packets, traffic, network traffic, data, and/or data objects 242), which would pass *between the operating system packet network stack and the WWAN modem*. Accordingly, Wright’s *one or more device agents* (e.g., LSP 814) has a privilege to

receive *Internet data packet traffic passing between the operating system packet network stack and the WWAN modem*.

172. Additionally, or alternatively, Wright also renders obvious *a privilege to receive ... Internet data packet traffic* based on such traffic being received by one or more components of the policy engine 832 (*one or more device agents*). For example, “policy engine 832” downloads policy data from a policy over a network (as explained in [1.9] above). Additionally, diagnostics module 844 (which is part of policy engine 832) performs similar functions [as] the client diagnostics module 246,” and is configured to “provide[] status, configuration, error logs, audit logs, and debug information to the server system” over a network. See SAMSUNG-1005, ¶¶284, 235-243; see *id.*, ¶61 (explaining that “[t]his information may be received periodically over a network 204 from the client diagnostics module 246”).

173. Given the above disclosures, a POSITA would understand that policy data and diagnostics-related data received over a network is *Internet data packet traffic*, and such network traffic would pass through the operating system network stack and the WWAN modem (which facilitate network communications to/from the device). A POSITA understood that traffic received over a network is in the form of data packets. SAMSUNG-1026, 20 (“IP packets are the bundles of data that form the foundation for the TCP/IP protocol suite. Every packet carries a source and destination address, some option bits, a header checksum, and a payload of data.”).

Therefore, Wright's policy engine 832 has *a privilege to receive Internet data packet traffic passing between the operating system network stack and the WWAN modem.*

174. *Second*, Wright teaches or renders obvious that the one or more device agents include a privilege to receive traffic information including the measured amount of the Internet data packet traffic associated with each of the one or more applications. For example, Wright teaches that “policy engine 832, in accordance with the current security policy, provides to the layered service provider 814 rules with respect to applications.” SAMSUNG-1005, ¶239. LSP 814, in turn, can “enforce[] the current policy in accordance with these rules applicable to applications.” Id. As noted above in [1.6], rules specific to applications can include size of files (traffic information including measured amount of Internet data packet traffic) being transmitted by/to applications. Thus, if LSP 814 enforces such a size-based rule, a POSITA would have understood or found obvious that the measured amount (e.g., file size) would be provided to (and thus received by) LSP 814 to facilitate such policy enforcement.

[2] The wireless end-user device of claim 1, the one or more processors further configured to execute, in the application execution partition, at least one device-assisted-services agent to communicate with at least one device agent executed in the protected execution partition.

175. As explained above, Wright’s system includes *one or more processors* that are configured to execute *an application execution partition* that includes one or more applications 830. *See supra* analysis for Element [1.3].

176. Wright’s layered service provider 814 (*at least one device agent executed in the protected execution partition*) communicates with applications 830 via a communication interface. SAMSUNG-1005, ¶238; ¶68 (“a communication interface, [can be] a software interface or buffers for storing parameters”); *see id.*, ¶¶58, 69. Via this interface, LSP 814 “determine[s] which application... accessing the network and what networking services the application will be using” and provides this information to other system components for application-specific policy enforcement. SAMSUNG-1005, ¶¶240-241. A POSITA would recognize that application-specific policy enforcement is a service provided by the agents/components of the device. Because providing application-specific information to the LSP 814, via the communication interface, assists in providing this policy enforcement device services, a POSITA would have recognized that the communication interface is a *device assisted services agent*. In the relevant timeframe, a POSITA understood that application interfaces are types of software agents tasked with facilitating inter-application communications. *See, e.g.*, SAMSUNG-1015, 153 (“Operating systems define an application programmer interface (API) consisting of a set of system calls it supports. New operating systems”).

often define new APIs, resulting in the need to port applications and/or write new applications. This has been a major problem in gaining acceptance for secure operating systems based on emerging kernels, such as Trusted Mach [35] and Flask [295], and the reason for so much focus on securing commercial systems.”). Thus, a POSITA would understand that an API is software that allows applications to communicate with other applications (in this context, applications who communicate with the operating system, which is effectively a (big) application. Moreover, a POSITA would have understood or found obvious that this communication interface (*device assisted services agent*) would be *executed within the application execution partition*. Wright teaches that the communication interface can be implemented in different ways, such as: (1) “a bus between a processor executing one or more of the modules and a memory controller responsible for memory reads/writes,” (2) “one module reading a parameter stored in a memory location by another module,” and (3) “other communication interfaces known to those of ordinary skill in the art.” SAMSUNG-1005, ¶58; see also id., ¶¶68-69, 228. Indeed, in the relevant timeframe, applications were known to communicate using application programming interfaces (APIs). Implementing an API as part of or in relation to an application would have amounted to a known technique being applied to a known system, as taught by Wright, to achieve predictable results of facilitating the application-to-LSP 814 communications contemplated by Wright. See, e.g.,

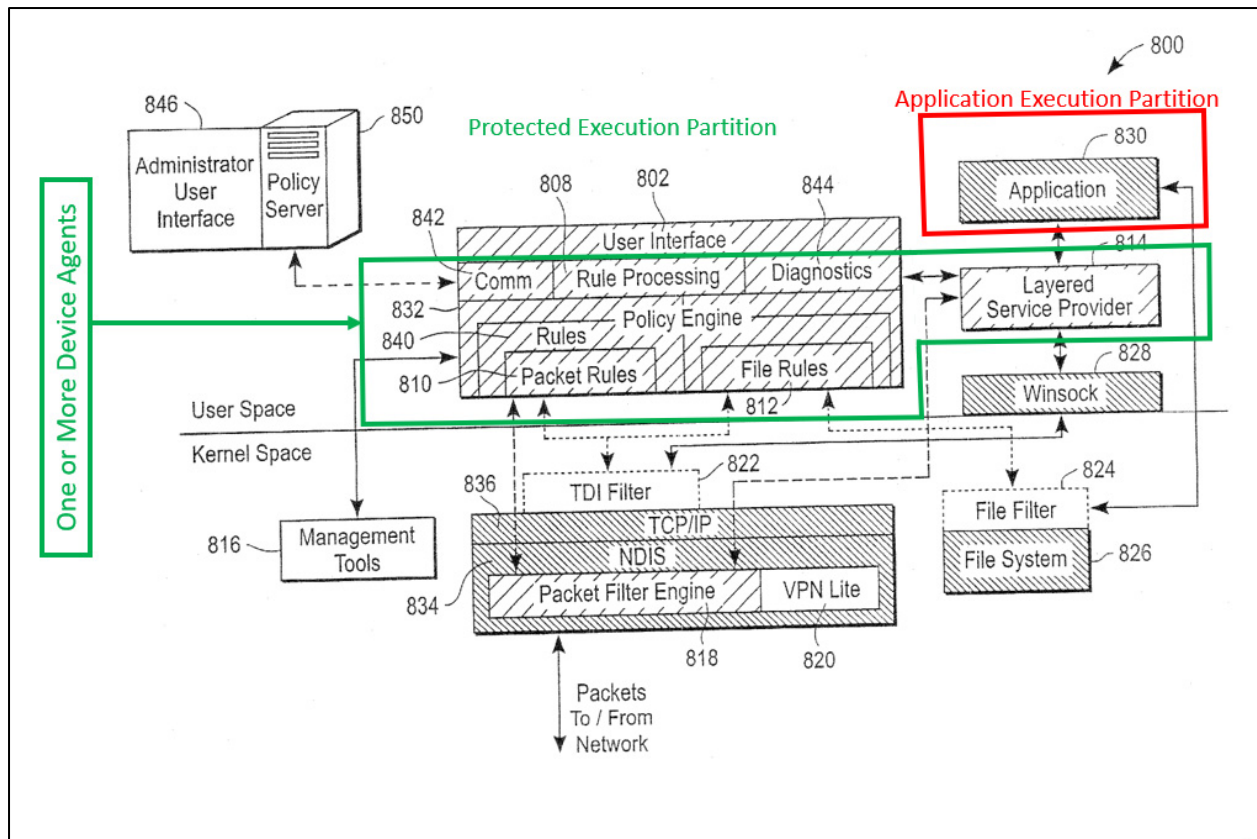
SAMSUNG-1018, 36 (“API (Application Program Interface, aka Application Programming Interface) 1. A set of routines by which an application program can call specific programs or services of a computer operating system (OS) or network operating system (NOS). See also application, NOS, OS, and routine. 2. A set of routines by which an application program allows another application program to work directly with it. See also application, program, and routine.”).

177. Such an implementation is further motivated by Wright, which contemplates use of other known communication interfaces (e.g., SAMSUNG-1005, ¶¶58, 68, 268), and would have been straightforward and readily implemented by a POSITA, who would recognize APIs implemented by an application as a common and conventional communication interface for enabling communications of other applications with that application. *See, e.g.*, SAMSUNG-1005 (“For email between users who are on the same corporate domain, email APIs may be used to redirect email and attachments through an encrypted channel to the policy server to the corporate mail server.”)

178. In this vein, Wright also teaches that “a communication interface, [can be,] for example a software interface or buffers for storing parameters.” SAMSUNG-1005, ¶68, 268 (“application parameters [are] received via [LSP] 814”); *see also id.*, ¶228; ¶285 (“Furthermore, policy management, compilation and interpretation is performed by policy aware application interfaces.”), ¶187

(contemplating storage of data such as “e-mail[s], a text message[s], a TIFF file[s], or a Word document”). This additional teaching suggests local storage of application-specific parameters, which then would have motivated a POSITA to implement the communication interface (e.g., APIs for the applications) at the respective applications or in an environment localized to applications (such as, e.g., in the application execution partition). For this and the above-described reasons, a POSITA would have found obvious to implement the above-described communication interface (device assisted services agent) as part of the application execution partition.

179. Further, Wright discusses the use of a device assisted service agent as implemented as the popular Winsock library, where “The TDI filter 822 further comprises a communication interface with a Windows Socket (Winsock) layer 828 in application space.” SAMSUNG-1005, ¶238. Here the agent ensures that “using the Winsock Environment, all communication between client and server is routed through a secure channel.” SAMSUNG-1005, ¶282. A POSITA would understand from Wright that the Winsock layer would receive policy from *device agent executed in the protected execution partition* as it is needed to know how to handle application traffic.



SAMSUNG-1005, FIG. 8 (annotated)

[3] The wireless end-user device of claim 1, further comprising one or more memory partitions to store program code for the respective processes of the operating system packet network stack, the application identification agent, the service measurement agent, the policy control agent, and the one or more device agents, the one or more processors configured to retrieve the stored program code from the one or more memory partitions to execute the respective processes.

180. As discussed above, Wright discloses or renders obvious one or more processes—namely, an *operating system packet network stack*, *application identification agent* (e.g., TDI filter 822), a *service measurement agent* (e.g., TDI filter 822), a *policy control agent* (e.g., TDI filter 822 and packet filter engine 818), *one or more device agents* (e.g., policy engine 832, LSP 814)—executing on one or

more processors (*e.g.*, a “processor”). *See supra* analysis at [1.3], [1.5]-[1.7]. SAMSUNG-1005, ¶¶73, 228, 302-303; FIGS 1, 8.

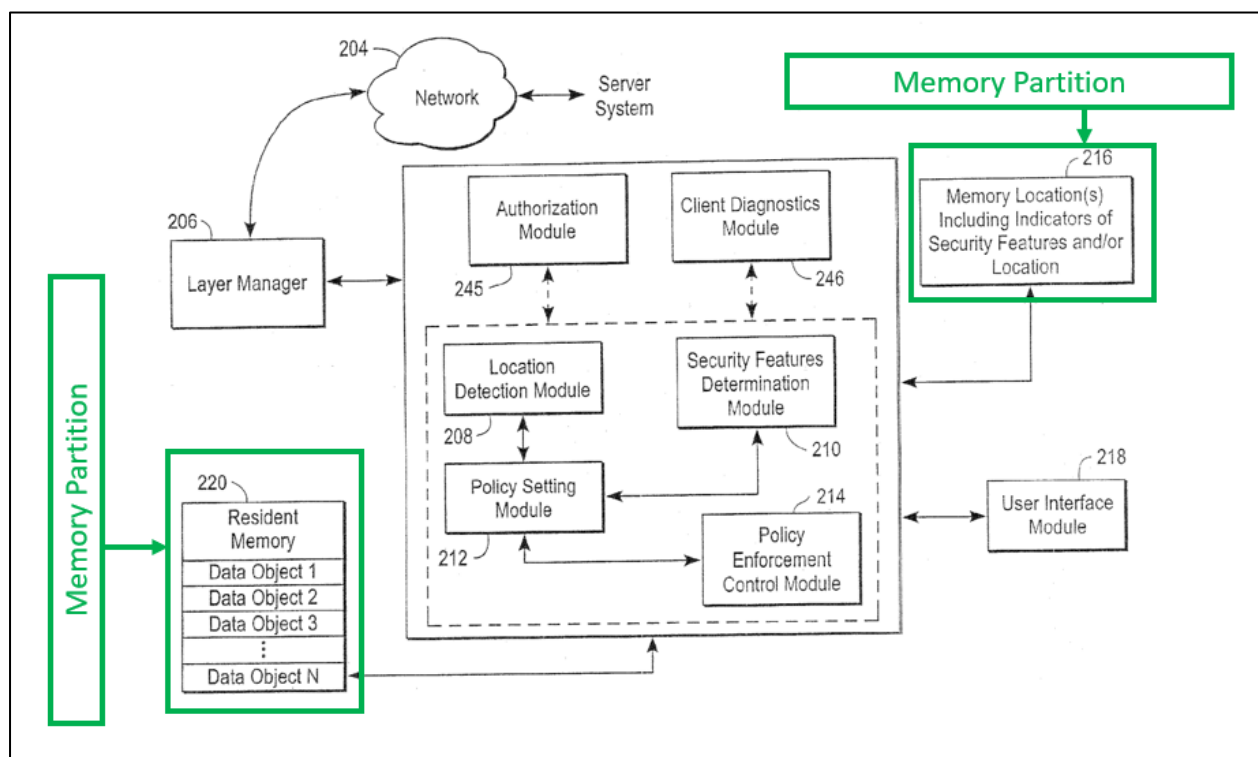
181. Wright further teaches or renders obvious that these processes correspond to computer programs including programming instructions. For example, Wright teaches that TDI filter 822 and packet filter engine 818 (as depicted in FIG. 8) perform some functions similar to the policy enforcement module 214 shown in Wright’s FIG. 2. *See* SAMSUNG-1005, ¶¶283, 301. Similarly, policy engine 832 (in FIG. 8) performs functions similar to those of FIG. 2 modules, such as location detection module 208, security features determination module 210, and a policy setting module 212. *See* SAMSUNG-1005, ¶¶283, 301. Wright teaches that these modules “can be implemented as software” or as part of a computer “program.” *See* SAMSUNG-1005, ¶¶302-303 (“wherever a component, an example of which is a module, of the present invention is implemented as software, the component can be implemented as a standalone program, as part of a larger program, as a plurality of separate programs...”). Indeed, Wright recognizes that these modules and other components/processes of the kernel and user space, *e.g.*, which are part of the embodiments of FIGS. 8 and 2, are “embodied as instructions stored in one or more computer usable mediums some examples of which are a memory....” SAMSUNG-1005, ¶13; *see also id.*, ¶65 (“system 201 protects data accessible ...

in resident memory 220,” which “may be accessed by one or more of these modules”) SAUMSUNG-1005, ¶65;

182. Given these disclosures, a POSITA would have understood or found obvious that Wright’s processes would be implemented as computer programs stored in its device memory (i.e., the device *store[s] program code for the respective processes*). Based on Wright’s teachings, a POSITA would understand that modules 208-214 in FIG. 2 store program code (e.g., “software” or “programming language” for modules).

183. Wright teaches or renders obvious *one or more memory partitions* for storing the program code for the above-described processes. Wright teaches that its device includes memory (e.g., memory locations 216 and resident memory 220), and as explained above, program code corresponding to the above-described processes would be stored in such memory. SAMSUNG-1005, ¶¶73, 228, 302-303. As described above, the device includes separate partitioned environments, with the processes in the kernel space being partitioned from those in the user space. One obvious and conventional way to achieve such partitioning would be to partition the memory allocated to each partition and its underlying programs. Such memory partitioning was well-known in the relevant timeframe and leveraging such memory partitioning would amount to applying known techniques to known system to achieve predictable results (partitioned environment as rendered obvious by Wright).

SAMSUNG-1014, 69; SAMSUNG-1022, 1283; SAMSUNG-1015, 41, 108; SAMSUNG-1015, 27. Indeed, a POSITA would know that a computer program “has its own address space, that defines the memory addresses that it can access.” SAMSUNG-1015, 41. Moreover, processes also run in *protection rings*, where “protection rings form a hierarchical layering from the most privileged ring, ring 0 where the most-privileged code in the supervisor runs, to the least privileged ring.” SAMSUNG-1015, 27. Taken together, a POSITA would know from these disclosures that programs in computer systems are partitioned in a variety of ways, ranging from memory isolation via virtual address spaces and through security policies (i.e., ring levels) both of which restrict programs to the memory locations they can access.



SAMSUNG-1005, FIG. 8 (annotated)

184. A POSITA would therefore understand or find obvious that the program code for the respective processes of the operating system packet network stack, the application identification agent (e.g., TDI filter 822), the service measurement agent (e.g., TDI filter 822), the policy control agent (e.g., TDI filter 822, packet filter engine 818), and the one or more device agents (e.g., the LSP 814, policy engine 832) would be stored in one or more memory partitions.

185. Moreover, and as explained in [1.3], Wright discloses one or more processors to execute the above-described partitions. A POSITA would have understood that a processor would execute these processes by retrieving and executing the underlying program code from where it is stored in memory. A POSITA would therefore understand that Wright's *stored program code* (e.g., "software" or "programming language," for modules) would be retrieved from the *one or more memory partitions to execute the respective processes*, and that the *one or more processors* (e.g., a "processor") were configured to store and retrieve such *stored program code* the above-identified processes.

[4.1] The wireless end-user device of claim 3, wherein the one or more memory partitions include a separately protected device-assisted services partition to store the program code for the one or more device agents,

186. As explained above for claim 2, Wright renders obvious *one or more memory partitions*, and one such memory partition would *store the program code*

for the one or more device agents in Wright’s user space. This memory partition would be a *device-assisted services partition* because it stores program code for facilitating “service usage policy implementation” (given the policy control and management functions of LSP 814 and policy engine 832), which the ’144 Patent explains is a device-assisted service. SAMSUNG-1001, 3:11.

187. A POSITA would understand or find obvious that this partition is *separately protected*, e.g., from other memory partitions. It was conventional and well-known that memory partitions to be secured. *See, e.g.*, SAMSUNG-1011, ¶¶17-36; FIGS. 2-3. Given that the one or more device agents are programs directed to policy control and management, a POSITA would have been motivated to provide security over this portion of memory, e.g., to avoid tampering with and ensure proper operation of the device’s policy-related functionality. There would also be a reasonable expectation of success in securing this memory partition, given the prevalent use of security mechanisms for memory partitions (e.g., encrypting information stored in a memory and/or limiting access). *See, e.g.*, SAMSUNG-1015, 17 (“First, a secure operating system must provide complete mediation of all security-sensitive operations. If all these operations are not mediated, then a security requirement may not be enforced (i.e., a secret may be leaked or trusted data may be modified by an untrusted process). Second, the reference monitor system, which includes its implementation and the protection system, must all be tamperproof.”).

Otherwise, an attacker could modify the enforcement function of the system, again circumventing its security.”); *id.*, 34 (“Tamperproof: How does the system protect the reference monitor, including its protection system, from modification? . . . Some ring 0 code must respond to calls from untrusted processes (e.g., system calls). The only way that ring 0 can be accessed by an untrusted process is via a gate. As described above, gates check the format of the arguments to higher-privileged, supervisor code to block malicious inputs. Thus, if the gates are correct, then untrusted processes cannot compromise any ring 0 code, thus protecting the supervisor.”). Thus, securing this memory partition would have been a known technique to improve known systems (such as those disclosed in Wright) to achieve predictable results (memory partitions storing program code that are separately secure). In this manner, Wright’s device-assisted services memory partition would be separately secured, e.g., from other memory partitions that are commonly secured as well (e.g., memory partitions dedicated to kernel processes). *See, e.g.*, SAMSUNG-1015, 17 (“First, a secure operating system must provide complete mediation of all security-sensitive operations. If all these operations are not mediated, then a security requirement may not be enforced (i.e., a secret may be leaked or trusted data may be modified by an untrusted process). Second, the reference monitor system, which includes its implementation and the protection system, must all be tamperproof. Otherwise, an attacker could modify the

enforcement function of the system, again circumventing its security.”); SAMSUNG-1014, 69 (“Programs should have only as much privilege as necessary: the principle of least privilege [662]”).

[4.2] The wireless end-user device of claim 3, wherein ... the one or more device agents further comprising an access control integrity agent having exclusive permission to update program code stored in the separately protected device-assisted services partition.

188. Wright’s policy engine 832 (*one or more device agents*) includes a diagnostic module 844 and policy-setting and management functionality, which are akin to the functionality of the policy setting module 212 and the diagnostic module 246 described with reference to Wright’s FIG. 2. SAMSUNG-1005, ¶¶283-284.

189. In Wright, the diagnostic module observes the operations and service usage of device components and provides the same to a server over the network for further evaluation, which the ’144 explains as “access control integrity” functions. SAMSUNG-1001, 9:55-65 (“In some embodiments, the access control integrity agent functions include one or more of hashing other device assisted security agents, querying other device assisted security agents, observing the operation of other device assisted security agents or monitoring service measures and then either evaluating the results locally on the device to determine if they are within pre-defined allowable parameters or sending at least some of the results to the service controller for further analysis via the service control link”). In particular, “the policy engine 832, the layered service provider 814, the file filter 824, and the packet filter

engine 818 report whether any of the other services have been disabled or removed from the system to the diagnostics module 844.” SAMSUNG-1005, ¶270. This information is “relayed to the policy server 850 as it indicates a possible compromise of the protective system” in which case “policy server 850 ...provides diagnostic support information to the diagnostics module 844” to remedy these issues. *Id.* The “support information may be in the form of instructions or code to the client device to repair a problem or perform maintenance,” which a POSITA would understand as being implemented by the diagnostics module given that “no user intervention or action” is required to perform such “repair” or “maintenance.” *Id.*, ¶62.

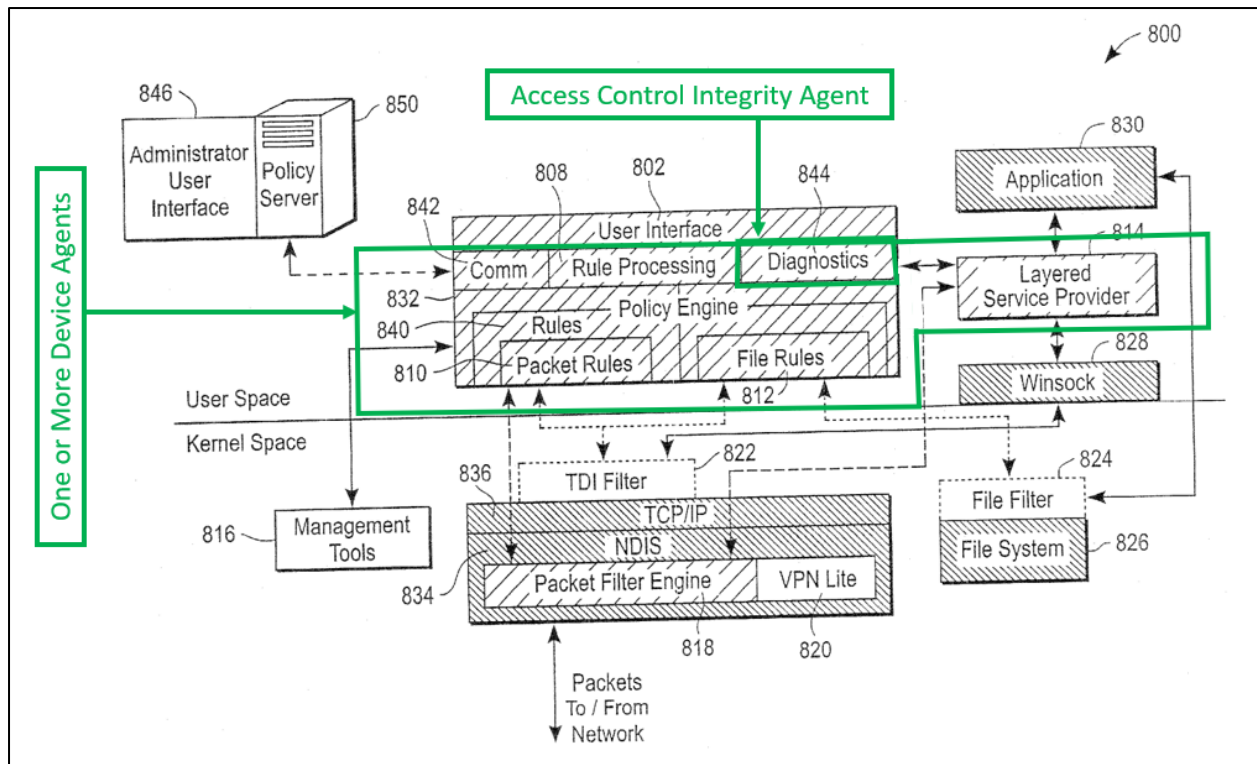
190. In view of the above disclosures, a POSITA would have understood that Wright’s diagnostic module is an *access control integrity agent* because it is an agent that performs access control integrity functions of the type disclosed in the ’144 Patent’s specification.

191. Moreover, the diagnostic module would have *exclusive permission to update program code stored in the separately protected device-assisted services partition*. As explained above, the diagnostic module receives and automatically applies “code” to “repair the problem” or “perform maintenance,” which a POSITA would understand as updating the stored program code with the received code to remedy the issue. Because the received “code” update may be for LSP 814, which is one of the device agents for which program code is stored in the device assisted

services memory partition (*see supra* analysis for claim 3), a POSITA would have understood that the diagnostic module has permission to update the program code for LSP 814 that is stored in this memory partition. Moreover, Wright only teaches that diagnostic module performs this, and conversely, no other module in Wright's system (other than the diagnostic module 844) is described as providing the above function of the receiving the code in the form of the "support information" and updating the same. A POSITA would have therefore understood that the diagnostic module 844 had an *exclusive permission* to update the above-described program code. Indeed, a POSITA would understand the *principle of least privilege*, which ensures that "Programs should have only as much privilege as necessary." SAMSUNG-1014, 69. Said differently, a POSITA would know that diagnostic module 844 had an exclusive permission to update the above-described program code because other programs would not have such permissions (in accordance with the principle of least privilege, which computer systems at that time (and today) comply with).

[5] The wireless end-user device of claim 4, the access control integrity agent further configured to validate the program code for the one or more device agents.

192. As described above, Wright teaches or suggests an *access control integrity agent* (e.g., client diagnostics module 246/diagnostic module 844). *See supra* analysis for claim 4.



SAMSUNG-1005, FIG. 8 (annotated)

193. As explained below, the diagnostic module is *further configured to validate the program code for the one or more device agents*, which as explained in [1.8] include the LSP 814, and one or more agents in the policy engine 832.

194. In more detail, Wright teaches “the policy engine 832, the layered service provider 814, the file filter 824, and the packet filter engine 818 report whether any of the other services have been disabled or removed from the system to the diagnostics module 844.” SAMSUNG-1005, ¶270. Additionally, the diagnostic module is configured to “provides status, configuration, error logs, audit logs, and debug information to the server system.” SAMSUNG-1005, ¶284; *see supra*,

analysis of Element. [1.10]. For example, Wright discloses that “files transferred between users may be checked for malicious software...” and that “[e]ach component of the system may also provide health checks on the others.” SAMSUNG-1005, ¶¶179, 270. Additionally, the “diagnostics module” runs “tests with respect to the security policies,” such as, e.g., running tests that “communicate to the outside world on blocked ports or an IP address,” with the output being “saved in a debug output file and sent to ... the remote diagnostics module 224.” SAMSUNG-1005, ¶83. “This information [from the diagnostic module 246/844] may be relayed to the policy server 850 as it indicates a possible compromise of the protective system.” SAMSUNG-1005, ¶270. Indeed, a POSITA understood that such a compromise would indicate incorrect or faulty operation of the underlying program code related to policy control and management. And, policy server 850 in that case provides, to the diagnostic module “support information” e.g., “in the form of instructions or code to the client device to repair a problem or perform maintenance.” SAMSUNG-1005, ¶¶270, 62; *see id.*, ¶¶77-80.

195. Given the above disclosures, a POSITA would have understood or found obvious that the diagnostic module evaluates the operation of components such as the LSP 814 (one or more device agents) and this evaluation would be understood as a validation of the program code corresponding to this device agent.

196. Additionally, or alternatively, the diagnostic module would also be configured *to validate the program code for the one or more device agents* in view of Wright’s teaching of a security information received by the policy engine 832 being authenticated. In Wright, when it is determined that a new “policy, software, or a message is available,” this “new security information is downloaded.” SAMSUNG-1005, ¶137. Thereafter, the “security information is authenticated,” e.g., by the authorization module 245,” which uses a hash comparison technique to compare the hash of the security information received with the hash included at the end of the decrypted file. *Id.*, ¶138 (“On the client side, a hash is performed with the key, and a comparison of the result made,” and “[i]f the comparison succeeds, then the client can safely conclude that the policy was generated on an authorized server, and that the [security information] has not been corrupted (nor tampered with)”); *see also id.*, ¶¶247-252, 285 (“the policies may also be protected from hacking by encryption or signatures”), ¶169 (“files transferred between users may be checked for malicious software by the policy server prior to being forwarded to the recipient”).

197. Given the above disclosures, a POSITA would understand that the received code is authenticated/validated. And because this code is security/policy-related code, it would be program code for the policy engine 832 (given that the underlying disclosure in Wright teaches that this received code would be

“install[ed]” and “update[d]” by the policy setting module, which has functions analogous to those of the policy engine 832). SAMSUNG-1005, ¶¶68 (policy setting module has functionality analogous to policy engine 832), 283 (“policy setting module 212 receives, installs and updates the security information including security policies and/or software updates received from the policy management module 236 via the policy distribution module 234 over the network connection 204”).

198. Additionally, Wright says that authorization module 245 performs the above-described validation of information received from the server system 200 (which is akin to the policy server 850, given its roles of disseminating policies). SAMSUNG-1005, ¶138. Although the authorization module is not depicted in the FIG. 8 embodiment, a POSITA would have found it obvious for this module to be implemented as part of the diagnostics module of FIG. 8. That’s at least because both modules perform similar functionality—e.g., they both communicate with the policy server (SAMSUNG-1005, ¶¶66-67, 270, 284) and both perform validations of code or other system functionality (SAMSUNG-1005, ¶¶62, 66, 84, 270, 284). Given that policy and other security information is conveyed to the policy engine in FIG. 8 and to the corresponding elements in FIG. 2B (e.g., policy setting module), and that diagnostic module of FIG. 8 is described as applying received code (similar to receipt and application of policy-related security information), a POSITA would

have been motivated and would have reasonably expected success to implement authorization module's functionality as part of the diagnostic module in FIG. 8's embodiment. Indeed, as noted previously, Wright's system aims to reduce tampering or modification of information exchanged between the device and the policy server (SAMSUNG-1005, ¶138) and the entity responsible for managing policy-related information on the client device is policy engine 832. This would provide further motivation to implement the authorization module's functionality as part of the policy engine 832 (and in one implementation, as part of the diagnostic module). Such a system configuration would be straightforward to implement and would result in Wright's system continuing to operate in the manner it already teaches. Thus, for these additional reasons, it would have been obvious that the diagnostic module (i.e., *access control integrity agent*) is configured to *validate the program code for the one or more device agents*.

[6] The wireless end-user device of claim 3, wherein the one or more memory partitions comprise a device-assisted services partition to store device service state information, the device service state information only modifiable by the one or more device agents.

199. Wright renders obvious that this memory partition would *store device service state information*. "Device service state information," as used in the '144 Patent, broadly encompasses various types of device information including, e.g., service usage reporting, reporting regarding service control state (e.g., service plan

settings, usage, preferences, key information, device agent reports, transaction records, threat reports, user status reports, device and/or application status information, device location, device QOS, and other state/settings information. SAMSUNG-1001, 17:25-58.

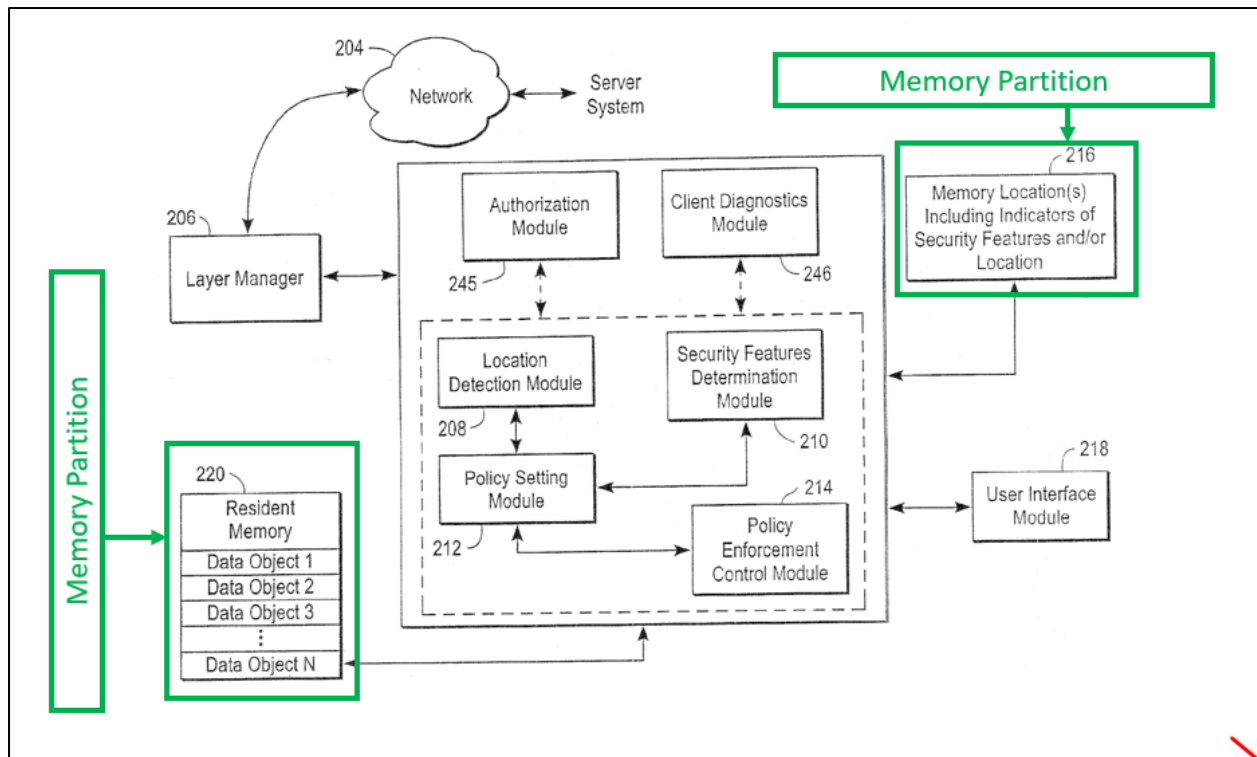
200. Consistent with this description, Wright teaches or renders obvious storing such *device service state information*. As explained below, Wright teaches storing *device service state information* in the form of device information (about its wireless end-user device), which can include diagnostic information, device configuration information, parameters, identifiers, information, reports, security features, policies, health checks or associated data and objects.

201. As shown below, Wright's FIG. 2B shows on-device memory locations 216 and 220 that store "indicators of security features and/or location" and data objects. SAMSUNG-1005, FIG. 2B, ¶¶54, 184, 69. Per Wright, such data stored in memory includes "security policies, corporate data, group, files indicating the organization of personnel into various groups, client device management data, and diagnostic information related to the mobile devices or computers internal to the corporate network." SAMSUNG-1005, ¶¶54, 184, 188, 193, 203-206, 213-216. Wright further teaches that one or more network parameters can be stored in memory 242 or resident memory 220. SAMSUNG-1005, ¶192. Wright additionally teaches that the diagnostic module is configured to collect device diagnostic information and

receive diagnostic support information (SAMSUNG-1005, ¶¶60-66, 82-84 (collecting debug reports, logs)), which includes software or policy information, and which is used to “repair” or “perform maintenance” on Wright’s other modules. SAMSUNG, ¶62; *see id.*, ¶284 (explaining that diagnostic module 844 in FIG. 8 provides functionality analogous to the diagnostics module 246). A POSITA would have understood that debug information that is collected and sent to a remote server would be obvious to store in buffers on device before being transmitted (such as the memory locations already described in Wright).

202. “Examples of network parameters are identifications of network services such as Gateways, Domain Name System (DNS) Servers, (Dynamic Host Control Protocol) DHCP and Domain servers. Other examples include identifications for Windows® Internet Naming Service (WINS) servers.” SAMSUNG-1005, ¶188. “... network parameters such as the examples of a Domain, Gateway, DHCP, DNS1, DNS2, DNS3, and WINS servers . . . Examples of such parameters include a MAC address associated with the data-link layer of the OSI model or the subnetwork layer of the Internet Protocol Stack (IPS), an IP address typically associated with the network layer of the OSI model and the Internet layer of the IPS model, a port value typically associated with the transport layer in the OSI or IPS models, and an application parameter (e.g. an application identifier or information derived by an application) typically associated with the application layer

in the OSI or IPS models. SAMSUNG-1005, ¶193. “This information is a layer 7 or application layer parameter upon which a location detection decision may be based. As seen in this example, different layers can provide additional verification of parameters received at other layers. Here the layer 7 parameter is verifying the layer 4 parameter. Here are other examples of criteria to be used for identifying a location: [0203] 1) The Default Gateway IP address and MAC address must both match the stored values, or [0204] 2) The DHCP server IP address and MAC address must both match the stored values, or [0205] 3) The Default Gateway IP address and the DHCP server IP address must both match the stored values, or [0206] 4) The local IP address must match the stored value.” SAMSUNG-1005, ¶¶203-206. “In this example, three parameters v_j , x_i and z_k are used to detect a given location. The variable v_j corresponds to a value, for example a numbered index, corresponding to a policy predefined environmental parameter such as an IP or MAC address from which the client device has received traffic while at the given location.” 214, SAMSUNG-1005, ¶¶213-216.



SAMSUNG-1005, FIG. 2B (annotated)

203. Wright teaches that the stored network parameters can be used for network environment location detection. See, e.g., SAMSUNG-1005, ¶239 (“network environment location detection is performed by the policy engine 832, in accordance with rules implementing one or more location detection tests in the Rules set 840, based on network parameters obtained by-the NDIS driver for OSI layers 2-5, and by the layered service provider for OSI layers 6 and 7”). Wright also teaches determining different aspects of security features, which as noted above, are also stored in device memory. See, e.g., SAMSUNG-1005, ¶¶283-2931 (with reference to FIGS. 9A-9D, Wright discloses several methods 900, 920, 930, and 940 for

determining aspects of a security feature, *e.g.*, 900: determining: whether a security feature of a connection type of wireless or wired is in effect, 920: whether the security feature of a security software program is currently executing, 930: determining one or more security features of a network access point; and 940: determining whether the security feature of a connection type of wireless or wired is in effect)). Additionally, Wright's policy engine 832 determines and "assigns" a "value" to a field in a data object "associated with each policy," *e.g.*, fields corresponding security software. *See, e.g.*, SAMSUNG-1005, ¶¶289-293; (describing storing values for different data objects). Because Wright explains that the data objects are stored in resident memory, a POSITA would have understood that assign values to fields in data objects would result in such values being stored in the corresponding memory locations for those data objects. *See, e.g.*, SAMSUNG-1005, ¶¶61, 67, 184, 283-293, 239, 192.

204. More specifically, Wright discloses that "[t]he policy engine 832 associates 904 the connection with an adapter type value of wired or wireless in an adapter data object associated with each policy. In a similar manner, the policy engine 832 may also determine 906 from the operating system the class (*e.g.* 802.11b, 802.3, 802.11a, 802.11g, GSM, GPRS, 3G) of the network adapters on the mobile device, and assign 908 a value to an adapter class field in the adapter data object associated with each policy. Furthermore, the policy engine 832 may

determine 910 from the operating system the hardware brand of the network adapters on the mobile device, and assign 912 a value to an adapter brand field in the adapter data object associated with each policy.” SAMSUNG-1005, ¶289.

205. Wright also teaches that “[t]he policy engine 832 then assigns 924 a value in a field corresponding to the security software in a data object associated with each policy;” Wright, SAMSUNG-1005, ¶290; and “[t]he policy engine 832 assigns 934 a value in a field corresponding to the security feature such as the security software, in a data object associated with each policy;” SAMSUNG-1005, ¶292. In addition, “[t]he policy engine 832 associates 944 the connection with a NAP type value of wired or wireless in a NAP [network access point] data object associated with each policy. In a similar manner, the policy engine 832 may also determine 946 from the operating system the class (e.g. 802.11b, 802.3, 802.11a, 802.11g, GSM, GPRS, 3G) of the NAP with which the mobile device is communicating, and assign 948 a value to a NAP class field in the NAP data object associated with each policy. Furthermore, the policy engine 832 may determine 950 from the operating system the brand of the NAP with which the mobile device is communicating, and assign 952 a value to a NAP brand field in the NAP data object associated with each policy.” SAMSUNG-1005, ¶293.

206. Moreover, as explained in the context of claim 4, Wright renders obvious a *device-assisted services memory partition* that stores program code for

the one or more device agents, which include LSP 814 and policy engine 832. Wright further teaches that client system 201 (which a POSITA would recognize includes Wright’s mobile device) “protects data ... in memory 220” that is used for management and enforcement of policies. SAMSUNG-1005, ¶65, 68-71 (policy setting and enforcement modules leveraging this stored data for their respective policy setting operations); *see id.*, ¶230 (explaining that policy rule engine 832 provides functionality analogous to policy setting module 212), 283 (explaining that policy rule engine 832 provides functionality analogous to location detection module, security feature determination module—which access location and security information data). A POSITA would have found it obvious for the above-described device state information that is leveraged by these device agents (e.g., for policy management), to be stored in the same memory region where the code for these agents is stored as well. A POSITA would have been motivated to so implement Wright’s system in view of Wright’s goal of “to protect” such device state data and given that this stored data is utilized by the policy rule engine, for which the code is already provided in a secure manner in the device assisted services memory partition. Such an implementation would have been straightforward and readily implementable by one skilled in the art.

207. Wright renders obvious that *the device state information is only modifiable by the one or more device agents*. As explained above, the policy engine

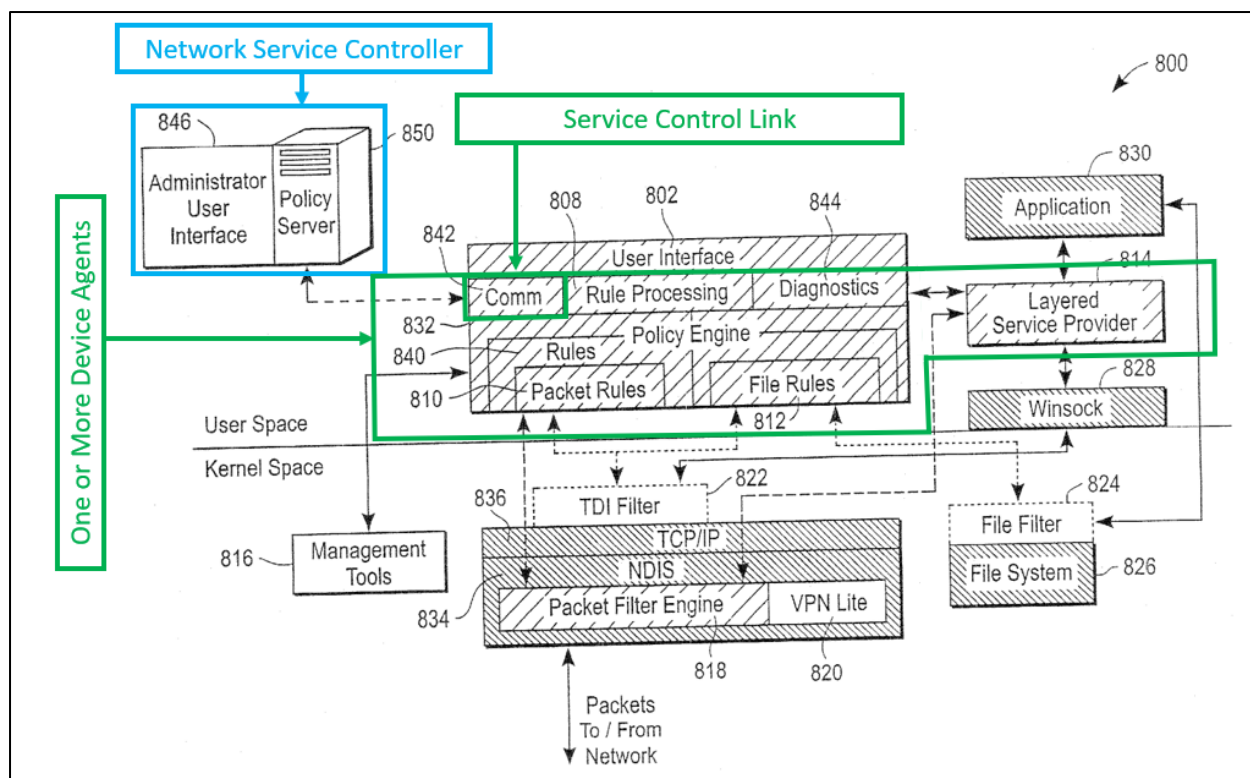
832 and its components are the only components that collect the above-described state information and update the values for the same in system memory. Similarly, a POSITA would understand Wright’s “repair” and “maintenance,” in the form of instructions or code, sent to the diagnostic module 844 to update stored program code as *device service state information*, which is *modifiable by the one or more device agents*. While Wright contemplates communication interfaces via which some of the security feature and location data is accessible by other system components (see SAMSUNG-1005, ¶66), Wright does not teach or suggest that any system components other than the policy engine 832 (or its components) as having the privilege to modify the stored device information. See, e.g., SAMSUNG-1014, 69; SAMSUNG-1015, 26, 41, 79. A POSITA would have found obvious for the policy engine 832 to be the component with such exclusive modify capability (1) to improve security of the system, e.g., by limiting the modification capability by multiple other system components, and (2) to further Wright’s goal of protecting such device state information (SAMSUNG-1005, ¶¶85-109, 244). A POSITA would have a reasonable expectation of success in so configuring Wright’s system given that a memory partition already limits access to the partition by an agent not privileged to access that partition and here, the one or more device agents already have access to and include code stored in this partition.

[7] The wireless end-user device of claim 6, the one or more device agents further comprising a service control link configured to securely communicate with a network service controller.

208. As explained below, Wright teaches or suggests that the one or more device agents further comprising a service control link configured to securely communicate with a network service controller.

209. Wright teaches that policy engine 832 (*one or more device agents*) includes communication port 842 to facilitate communication with a policy server 850. SAMSUNG-1005, ¶236. As shown in FIG. 8, “policy server 850 ... is communicatively coupled over a network with the client side system,” and in particular, the policy engine 832, “through a communication port 842.” *Id.* Here, a POSITA would understand the communication port 842 would contain connection interfaces, e.g., UNIX socket data, file handles, and APIs, that are used to support communication with the policy server (“A socket is defined as an endpoint for communication. A pair of processes communicating over a network employ a pair of sockets—one for each process. A socket is identified by an IP address concatenated with a port number.” SAMSUNG-1025, 108.). A POSITA would understand that Communication Port 842 is in fact part of Policy Engine 832 for at least two reasons. Firstly, a POSITA would know that a communication port represents a communication link between two entities. Figure 8 in Wright clearly demonstrates that the communication port, which is part of the Policy Engine 832,

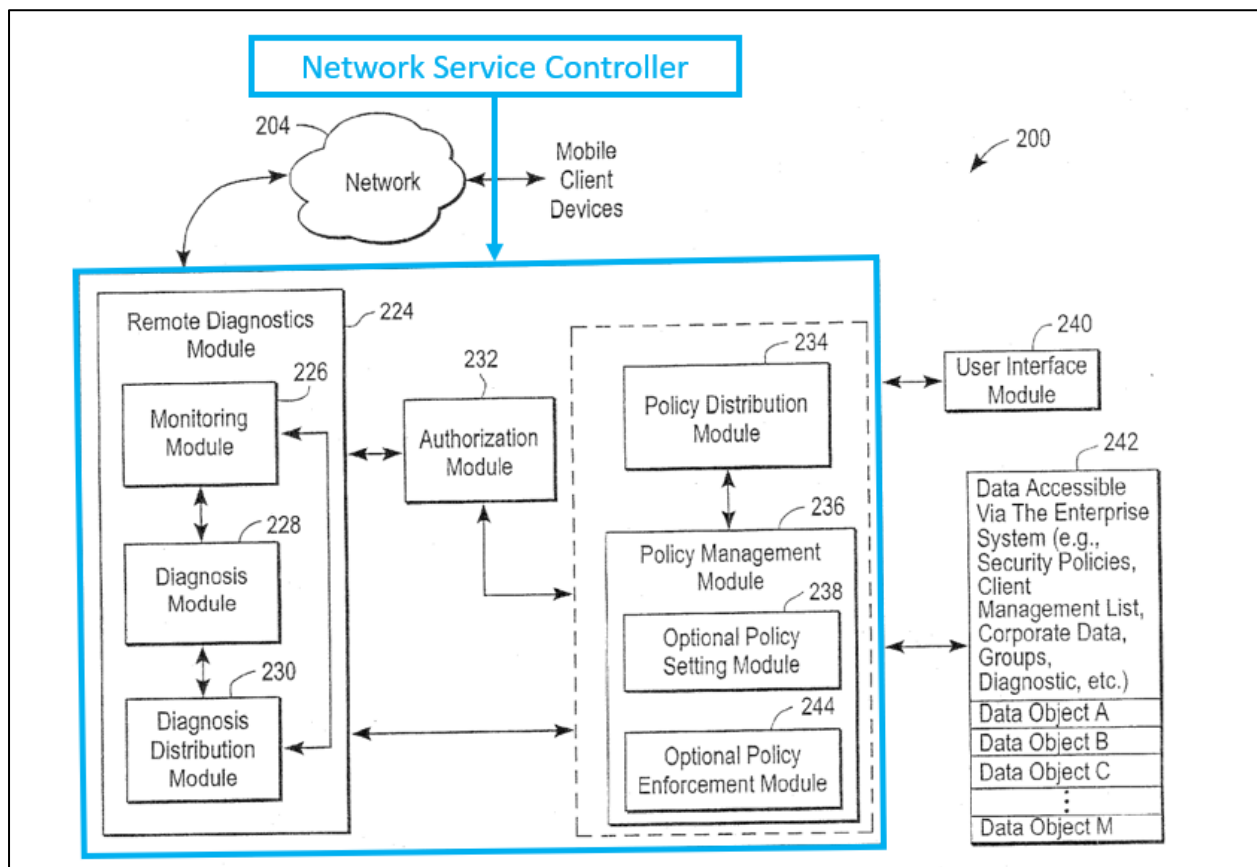
communicates with the Policy Server 850 (and therefore, Communication Port 842 (service control link), which is part of the Policy Engine (device agent), is configured to securely communicate with a Policy Server (network service controller)). Additionally, Wright explicitly teaches how Policy Engine 832 communicates with Policy Server 850, namely: “As illustrated, the server system embodiment comprises a policy server 850 having an administrator user interface 846. This system is communicatively coupled over a network with the client side system through a communication port 842.” SAMSUNG-1005, ¶236. “The Policy Engine 832 may create a security policy that is not inconsistent with the policies downloaded from the policy server 850. Additionally, modification and local management of policies as allowed, for example in accordance with permissions of policies set by the policy server 850.” SAMSUNG-1005, ¶244. Specifically, a POSITA would understand from these disclosures that: (1) the Policy Server 850 is has a link to the client side system, and (2) this link is a *service control link* to Policy Engine 832 because Policy Engine 832 may “download [policies] from the policy server 850.” That is, there would be a control link between Policy Engine 832 and Policy Server 850 for the engine to download information from the server. And, because the information downloaded is in the form of network policies, such as control link is also necessarily a service control link because such policies govern the behavior of the network.



SAMSUNG-1005, FIG. 8 (annotated)

210. Policy server 850 is a **network service controller** because it is an entity accessible by the client/mobile device over a network that provides and/or controls services used by the client/mobile device, given that it provides policies and other security information to be enforced and/or used by the mobile device (which may be based on security features and one or more parameters)—which in turn can control access of the device to network services (e.g., access to servers in the network and exchanging data therewith). SAMSUNG-1005, Abstract, ¶236 (“the server system embodiment comprises a policy server 850” and “...is communicatively coupled over a network with the client side system through a communication port 842;” and

the policy server 850 “... may comprise an authority server executing on one machine for establishing communication sessions ... with which client devices [that] may communicate in order to obtain their updates and receive diagnostic assistance.”); see also id., 244 (“...modification and local management of policies [by the policy engine 832] as allowed, for example in accordance with permissions of policies set by the policy server 850.”); *see also id.*, ¶¶270, 281-287.



SAMSUNG-1005, FIG. 2A (annotated)

211. Additionally, the communication port 842 is a *service control link* because it is a network link over which the above-described service-related control

information (e.g., policy-related security information that controls network communications by the device) are exchanged between the policy server and the device's policy engine 832.

212. A POSITA would have understood that this service control link facilitates *secure communications* given that Wright teaches or at least renders obvious that this link facilitates encrypted communications between the policy server 850 and the policy engine 832. As explained above, policy engine 832 has analogous functionality as that provided by policy setting module 212, which “determines 338 if the security information” (which includes software or policy information) received from the policy server/server system 200 “is encrypted 338.” SAMSUNG-1005, ¶137; *see* SAMSUNG-1001, 6:61-7:316:66-67 (describing that “the service control link is a secure link (e.g., an encrypted communication link)”). “If so, the security information is decrypted 340” by the policy setting module 212 “using cryptographic information stored in memory (e.g. 220) that was previously downloaded in a secure session within the enterprise’s internal network.” *Id.* (describing hashing of received file, and encryption/decryption using encryption protocols such as AES). Because the policy engine 832 enables encrypted communications with policy server and such communications happen over communication port 842, a POSITA would have recognized that the communication

port 842 is configured to and enables *secure communications* with the policy server (*network service controller*).

213. A POSITA would understand that the *service control link* (communication port 842) is configured to transmit information to/from other modules in the client device 201 and system 700/800, as shown in FIGS. 2B, 7, and 8. For example, a layer manger 206, as shown in FIG. 2, communicates via a “communication interface [with] one or more modules 208, 210, 212, 214, 245, and 246,” and receives “network traffic which may include inbound data [from server system 200] accessible over a network and outbound copies of data objects from resident memory 220.” SAMSUNG-1005, ¶67. Similarly, communication port 842 (in FIG. 8) “is communicatively coupled over a network with” policy server 850. SAMSUNG-1005, ¶236.

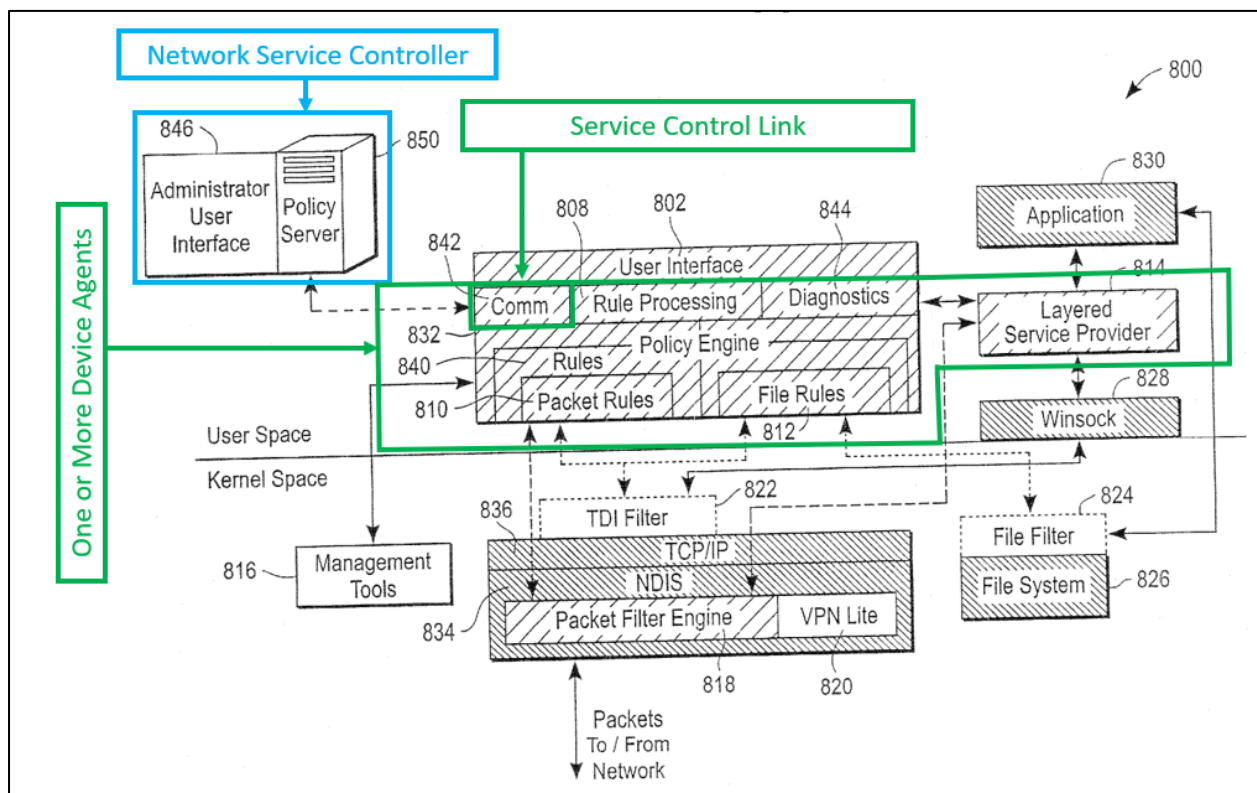
214. Additionally, or alternatively, a POSITA would find obvious that Wright’s authorization module 232 in combination with the communication port 842 corresponds to a *service control link* configured to securely communicate with the Wright’s policy server. Wright teaches that “security information” received from the server “is authenticated 341 by the client system, for example by the authorization module 245” of the mobile device. SAMSUNG-1005, ¶138, FIG. 2B.

215. A POSITA would have found it advantageous for the policy engine 832, using Wright’s communication port 842 (alone or in combination with the authorization module), to securely communicate with the ***network service controller*** (e.g., policy server 850) to prevent security information, policies, and other support information downloaded from the policy server 850 and implemented at the policy engine 832 from being corrupted, tampered with, or subject to malicious software—and in this manner, enhancing security of the system. SAMSUNG-1005, ¶138; *see also id.*, ¶¶247-252. A POSITA would have reasonably expected success in so implementing the system given that Wright contemplates updating a mobile device with encrypted information over “peer to peer” communication or an “encrypted” or “secure” channel from a server using a secure device, such as authorization module 232. *See* SAMSUNG-1005, ¶¶66, 169, 274-281, 282, 298-300.

216. Thus, Wright renders obvious ***one or more device agents*** (e.g., policy engine 832) further comprising ***a service control link*** (e.g., a communication port 842 alone or in combination with authorization module 245) ***configured to securely communicate with a network service controller*** (e.g., policy server 850).

[8] The wireless end-user device of claim 7, wherein the service control link is further configured to receive at least one of the application-specific traffic policy controls from the network service controller.

217. As explained with reference to element 1.7, Wright's policy engine 832 receives, from a policy server and over a network, *application specific traffic policy controls* in the form of policies and other security information that are used for application-specific traffic filtering. Additionally, as explained in claim 7, Wright's policy engine 832 communicates with the policy server (i.e., network service controller) via at least communication port 842, which is part of the *service control link*.



SAMSUNG-1005, FIG. 8 (annotated)

218. Given that (1) Wright’s policy setting module “receives, installs and updates the security information including” the above-described security policies (which are *application-specific traffic policy controls*) a policy server (*network service controller*) “over the network connection 204,” (SAMSUNG-1005, ¶¶68, 239, 269), (2) Wright policy setting module’s functionality is analogously provided by policy engine 832 of FIG. 8 (SAMSUNG-1005, ¶¶238, 230), and (3) Wright’s policy engine communicates with the policy server via its communication port 842 (*service control link*) (*see claim 7 supra*), a POSITA would have understood or found obvious that Wright’s *service control link is configured to receive at least one of the application-specific traffic policy controls from the network service controller*.

219. Wright teaches that a security policy can be downloaded by the mobile device from a policy server with assistance of the policy setting module. *See* SAMSUNG-1005, ¶68. With reference to FIG. 2B, Wright teaches that using “criteria defined in a downloaded policy from the server system 200,” the client system “detects or determines the location associated with the current network environment.” SAMSUNG-1005, ¶68. The client system 201 has a “policy setting module 212 [that] receives, installs and updates the security information including security policies and/or software updates received from the policy management module 236 [in the computer/server system 200] via the policy distribution module

234 over the network connection 204.” SAMSUNG-1005, ¶68. The client system has “[a] policy setting module 212 [that] may define criteria or if permissions set by the policy management module 236 allow, supplemental policy definitions or customization of policy definitions based upon user input,” or “define[] an aspect of a policy such as location criteria or security features criteria.” SAMSUNG-1005, ¶68.

220. For example, Wright discloses several modules that enforce application-specific rules in the system 800. In one example, Wright teaches that *application-specific traffic policy controls* (e.g., rules for or related to application-specific filtering or application-specific network access, such as filtering of packets/traffic) include “rules [that] are passed to ... the TDI filter 822 for *application specific filtering*.” SAMSUNG-1005, ¶269. A POSITA would understand that these rules are received in the form of packets or internet traffic and that the communications port 842 was suitable for receiving these packets/traffic.

221. In another example, the policy engine 832 receives *application-specific traffic policy controls* (e.g., a policy(e.g., filtering of packets/traffic)). In more detail, (“The policy engine 832 may also pass rules about application-specific network access to the layered service provider 814 via an IOCTL interface.”). SAMSUNG-1005, ¶269 “[T]he policy engine 832, in accordance with the current security policy, provides to the layered service provider 814 rules with respect to applications,” and

“may provide a list of which applications can or cannot access the network,” and provide rules to “[t]he layered service provider 814 then enforces the current policy in accordance with these rules applicable to applications.” SAMSUNG-1005, ¶239.

222. In this regard, and as an example, a POSITA would understand that a mobile device configured to communicate over the internet can have processors, memory, and modems, of various sizes. A mobile device, running an email application, may have a different processing and memory capacity than a personal computer (PC) or a server. A POSITA would find it obvious that a maximum size of a file that can open or execute on a mobile device could be different than a maximum size of a file that can open on the PC or server. Thus, a POSITA would find it beneficial, where permitted by policy, to enable the user to limit the size of an application running on a mobile device. By enabling the user to limit the size of a file/object that could open in an application on a mobile device, a POSITA would mitigate against crashes, errors, and non-responsive applications that attempt to handle files/objects that force the processor to operate beyond operating limits.

223. Wright therefore renders obvious a *service control link* (e.g., communication port 842) that is *configured to receive at least one of the application-specific traffic policy controls* (i.e., packets/traffic related to rules or policies for a given application (e.g., filtering of packets/traffic)) from the *network service controller* (i.e., a policy server 850 or computer system 200).

[9] The wireless end-user device of claim 7, wherein the service control link is further configured to transmit at least some of the device service state information to the network service controller.

224. As explained below, Wright discloses wherein the ***service control link*** (e.g., communication port 842) is further ***configured to transmit*** at least some of the ***device service state information*** (e.g., authorization information, diagnostic information, policies, version numbers, and health checks) to the ***network service controller*** (e.g., a policy server 850).

225. As explained above for claims 7 and 8, communication port 842 (***service control link***) is used to facilitate communications from the policy engine 832 to the policy server (***network service controller***). SAMSUNG-1005, FIGS. 2B, 7-8.

226. Wright further teaches that diagnostic information (device service state information) is transmitted to the policy server. SAMSUNG-1005, ¶61 (providing examples of diagnostic information), 188-190, 27; SAMSUNG-1001, 17:25-58 (defining “state information” as including “current DAS agent status reports, current DAS agent security state reports, current ambient service usage and/or transaction records, current service control integrity threat reports, user status information, device status information, application status information, device location, device QOS state, and/or other state and/or settings information”).

227. The policy server’s “monitoring module 226 receives diagnostic information such as events or audit logs from a client device and stores the information in a data object (242) for the client device.” SAMSUNG-1005, ¶61. In addition, “a client diagnostics module (e.g., FIG. 2B, 246) periodically and automatically initiates tests” and results “including errors from these tests are reported over a network 204 (e.g., Internet) to the remote diagnostics module 224.” SAMSUNG-1005, ¶61; *see id.* (“other examples of diagnostic information retrieved from the client are debug output files, examples of which include system event logs, crash dumps, and diagnostic outputs from a client diagnostics module (e.g., 246, FIG. 2B).”).

228. Consistent with the ’144 Patent, Wright also discloses transmitting, to the policy server, ***device service state information*** that includes “client configuration information may be retrieved such as the current version of the security policy software components on the client device, the current policy settings on the device, and attributes in accordance with those settings, for example, which ports are blocked. SAMSUNG-1005, ¶64. Per Wright, “remote diagnostic module 224 receives 1102 a notification of an error occurrence on a client mobile device,” and where “error or errors have occurred . . . downloads 1106 a debug version for each of the effected software modules or portions.” SAMSUNG-1005, ¶82.

229. Wright additionally teaches transmitting device status information to the policy server, including sending “a status request upon network access starting up to the server system” where the “status request provides information regarding the client mobile device, examples of which are authorization information, policies and their version numbers resident on the device, and diagnostic information.” SAMSUNG-1005, ¶134. This is similar to ’144 Patent’s teaching that “state information” includes policy settings, user preference settings, and/or channel key information.

230. Additionally, Wright teaches that “authorization module 232 [in the server/ *network service controller*] receives 322 the status request with authorization information from a client mobile device, and it determines 324 whether a communication exchange with this mobile device is authorized. If not, the status request is ignored 326 or an error message is sent 326.” SAMSUNG-1005, ¶134. “One example of authorization information is an encrypted token provided by the authorization module 232 to the mobile client device during a trusted connection between the two.” SAMSUNG-1005, ¶85.

231. Additionally, as noted above, diagnostic information can include: “events or audit logs,” “errors,” and “debug output files, examples of which include system event logs, crash dumps, and diagnostic outputs.” SAMSUNG-1005, ¶61; *see also id.*, ¶¶188-190 (describing a service identifier designation). Wright’s policy

engine 832 performs “health checks” that results in “report[ing] whether any of the other services have been disabled or removed from the system to the diagnostics module 844” and “...relayed to the policy server 850.” SAMSUNG-1005,¶270.

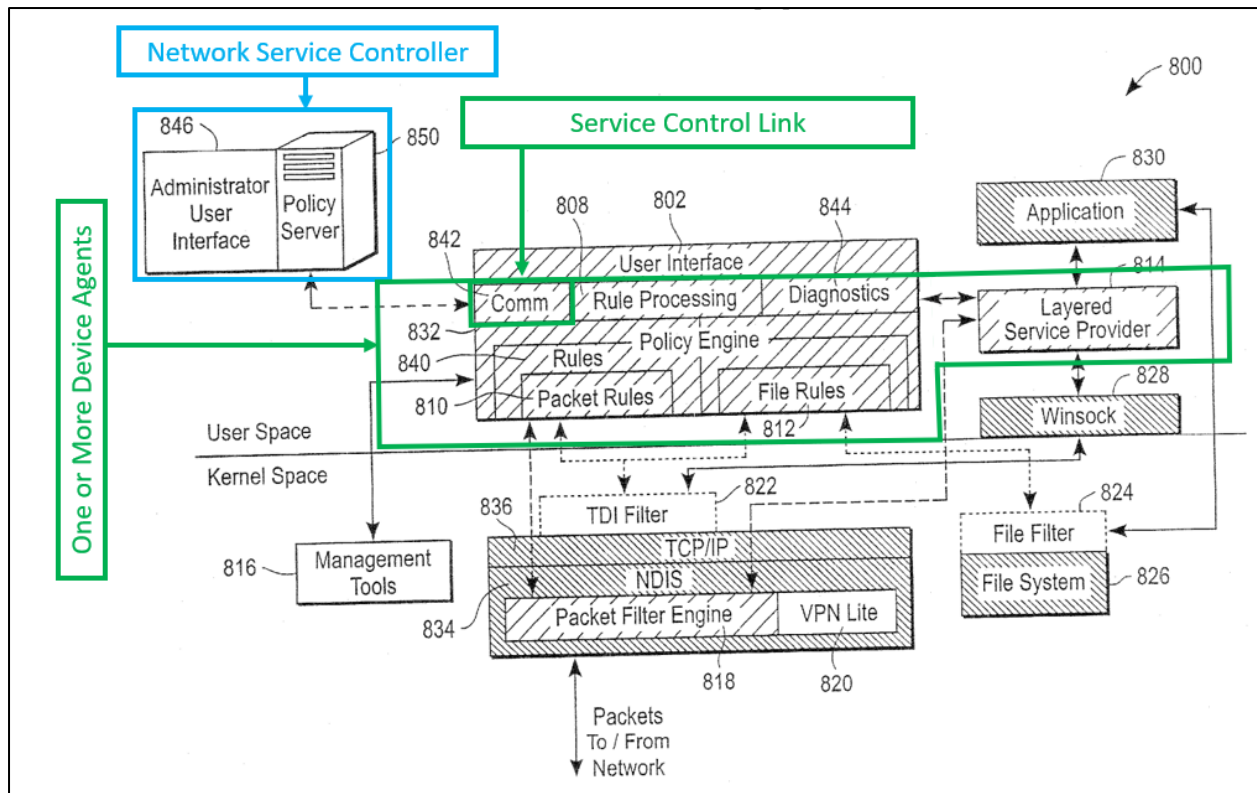
232. Accordingly, a POSITA would understand that Wright teaches wherein *the service control link* (e.g., communication port 842 and/or authorization module 245) is configured to *transmit some of the device service state information* (e.g., authorization information, diagnostic information, policies, version numbers, and health checks) to the *network service controller* (e.g., a policy server 850).

[10] The wireless end-user device of claim 9, wherein the one or more device agents are further configured to retain the transmitted device service state information until after an instruction is received from the network service controller acknowledging that the device service state information can be deleted.

233. Wright renders obvious that the *one or more device agents* (e.g., policy engine 832) are further configured *to retain the transmitted device service state information* (e.g., one or more of Wright’s diagnostic information, device configuration information, parameters, identifiers, information, reports, and health checks) received from the *network service controller* (e.g., policy server 850).

234. In more detail, policy engine 832 (*one or more device agents*) stores (retains) the *device service state information* (e.g., authorization information, diagnostic information, policies, version numbers, and health checks), which it transmits to the policy server (network services controller, per claim 9). For

example, “policy engine 832 assigns 934 a value in a field corresponding to the security feature such as the security software, in a data object associated with each policy.” SAMSUNG, 1005, ¶293; *see also id.*, ¶291. Similarly, “policy engine 832 may also determine 906 from the operating system the class (e.g. 802.11b, 802.3, 802.11a, 802.11g, GSM, GPRS, 3G) of the network adapters on the mobile device, and assign 908 a value to an adapter class field in the adapter data object associated with each policy.” SAMSUNG, 1005, ¶290. In the context of mobile device shown in FIG. 2, Wright’s “[s]ecurity policies may be stored as data objects accessible over a network 242 in FIG. 2A or in resident memory 220.” SAMSUNG, 1005, ¶185, FIG. 2A-2B; *see id.*, ¶193 (“[t]hese one or more network parameters identifying the location are stored in a data object (e.g. look-up table) in memory 242,” and “...may be stored in a data object in resident memory 220, particularly for locations defined by the client system 201.”); *see* FIG. 2B.



SAMSUNG-1005, FIG. 8 (annotated)

235. In another example, Wright teaches that “[d]ebug versions of the software in which the error occurred are downloaded from the server system and installed 1108 on the client device. The output from these debug versions, for example time stamped values of parameters, are captured 1110” (which a POSITA would have understood as being stored at the client device) and transmitted “to the remote diagnostics module 224” of the computer/server system 200. SAMSUNG-1005, ¶82; see *id.*, ¶¶81-83 (providing examples of diagnostic information), ¶285 (explaining that diagnostic module 844 in FIG. 8 performs

functions similar to the diagnostic module in embodiment of FIG. 2B), FIGS. 2A-2B, 8, 11, ¶¶188-190.

236. Given the above disclosures, a POSITA would have understood that the policy engine's 832 ability to capture and/or assign values in data objects, store policies and network parameters in data objects and resident memory 220, as well as provide status reports and logs, corresponds to *one or more device agents* (e.g., policy engine 832) being configured to retain the *transmitted device service state information* (e.g., policies).

237. A POSITA would understand that the transmitted *device service state information* (e.g., authorization information, diagnostic information, policies, version numbers, and health checks) would be retained in the *one or more device agents* (e.g., policy engine 832) *until after an instruction is received that the device service state information can be deleted*. For example, Wright teaches that a "client mobile device[]" sends a status request upon network access starting up to the server system" that includes device information, e.g., "authorization information, policies and their version numbers resident on the device, and diagnostic information." SAMSUNG-1005, ¶134. If the server determines updated information is available for the device, (e.g., a new policy, a new version, or new software being available), the server "distributes 327 the updated security information to the mobile device over the network (e.g. Internet) 204" (*id.*) and this security information is then

“receive[d], install[ed] and update[d]” by the policy setting module 212 of the device (see SAMSUNG-1005, ¶68). Because the previously-stored and transmitted information (*transmitted device service state information*) is updated with the newly-received security information, a POSITA would have understood or found obvious that this update would result in deletion/overwriting of the previously-stored information with the newly-received information. Thus such receipt of the newly-received and updated security information is akin to an instruction by the server system/policy server acknowledging that the previously transmitted and retained information can be deleted (e.g., by being overwritten or updated with new information).

238. As another example, Wright teaches that the diagnostics module (*one or more device agents*) provides “status, configuration, error logs, audit logs, and debug information to the server system” (SAMSUNG-1005, ¶¶285, 270; FIGS. 2A, 8, 11), which then provides “a repair to a problem in the client device, determine that a trend is occurring for the device, or determine that preventive maintenance is to be scheduled for the client device.” SAMSUNG-1005, ¶62. “Once a diagnosis has been made [by the diagnosis module 228 of the computer/server system 200] with respect to a problem, support information may be distributed to the mobile device under the control of the diagnosis distribution module 230.” SAMSUNG-1005, ¶¶62, 82. A POSITA would understand Wright’s support information, included

“instructions or code” for the software update, to delete, overwrite, or alter previous versions or a portion of the installed software. SAMSUNG-1005, ¶62. In this regard, and as described in the preceding paragraph, the receipt of the newly-received instructions or software is akin to an instruction to delete the previously transmitted and retained diagnostic information.

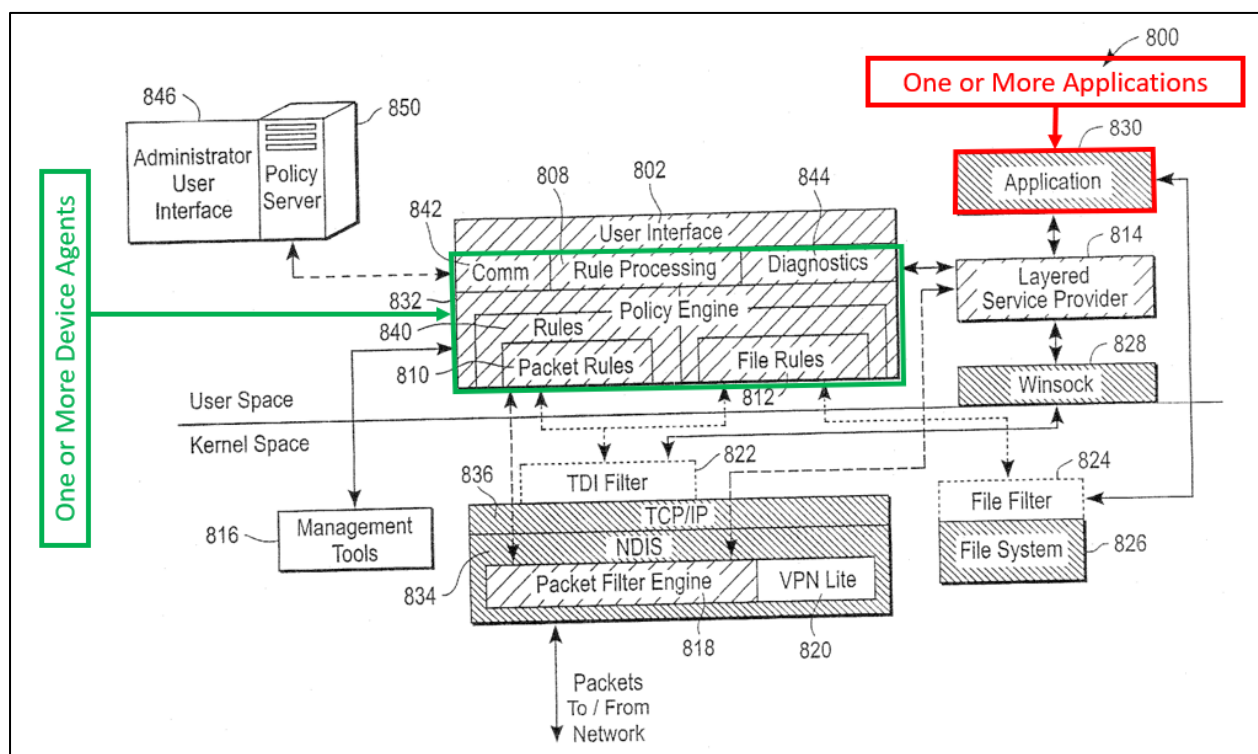
239. Thus, Wright renders obvious that the *one or more device agents* (e.g., policy engine 832) are configured to retain the *transmitted device service state information* (e.g., authorization information, diagnostic information, policies, version numbers, and health checks) until after the *instruction* (e.g., instructions or code) is received from the *network service controller* (e.g., policy server 850) acknowledging that *the device service state information* (e.g., parameters or software updates related to policies) can be deleted.

[11] The wireless end-user device of claim 1, the one or more device agents further configured to, based on the application-specific traffic policy controls, configure application settings for the one or more applications.

240. As described above, Wright’s policy engine 832 (*one or more device agents*) configures *application-specific traffic policy controls* for the client mobile device. *see supra* analysis for elements [1.7] and [1.9].

241. In view of Wright’s additional teachings, a POSITA would have understood or found obvious that policy engine 832 (*one or more device agents*)

configures, based on the above-described application-specific traffic policy controls, application settings for the one or more applications (e.g., permissions to modify policies, e.g., permissions to modify settings, such as the ability to modify visual settings, create/modify global objects, allows access to a specific file). SAMSUNG-1005, ¶¶55, 68, 72, 85-109.



SAMSUNG-1005, FIG. 8 (annotated)

242. In a first example, application-specific traffic policy controls correspond to application-specific traffic rules and policies, which include permissions “relat[ing] to the allowable modification that may be made to a downloaded policy or client software by the client mobile device.” SAMSUNG-

1005, ¶85. see id., ¶¶93, 106-107 (describing that permission may be “modifiable” and “recommended,” settings which grant permission to the mobile phone user to modify a parameter associated with a policy.) See also id., (describing permissions as a “policy override”) id., ¶218 (describing override criteria). An example of a such a policy-specific permission includes an “[a]bility to modify visual settings.” SAMSUNG-1005, ¶94; Id, ¶55 (“a policy includes specification of rules and permissions (e.g. policy override), defining one or more locations associated with...applications to be monitored (emphasis added);” id., ¶72 (“user interface module 218 receives input from a user input device Such as a keyboard, mouse, or touchpad, and causes user interfaces to be displayed for use by a user for establishing criteria for defining an aspect of a security policy as allowed by permissions....”))

243. Because Wright’s policies are application-specific, a POSITA understood or found obvious that a permission to adjust visual settings for a policy would be an *application setting* (i.e., a visual setting for an application) that is configurable based on *application-specific traffic policy controls* (Wright’s application-specific policies). Given these disclosures, a POSITA would understand that Wright’s policies include permissions to adjust settings related to the policy, such changes to visual settings. Because policies in Wright can be application-specific, a POSITA would have understood or found obvious that a permission to adjust visual settings would be an *application setting* (i.e., a visual setting for an

application). A POSITA understood or found obvious that adjustment of visual settings, in an application context, would entail, e.g., adjustment of appearance of text, images, or other visual attributes as provided for display within an application.

244. In a second example, *application-specific traffic policy controls* correspond to application-specific rules or policies as they relate to attributes of a file or object, such as size, type, etc., that may be transferred by applications. See SAMSUNG-1005, ¶169 (defining policies applicable to “type, size, format, and other attributes of transferred files”); see also *id.* at ¶163 (“Us[ing] the “software update size” or “policy size” to calculate an estimated download time.”); ¶¶158-163; see *supra* analysis for element [1.6].

245. Because (1) Wright describes application-specific policies can be based on file attributes of files being transferred and (2) Wright’s policies can have permissions that allow modification of certain settings of policies, a POSITA would have found obvious to implement policies with permissions to modify settings (application settings) based on attributes application to files being transferred by applications (*based on application-specific traffic policy controls*). In the context of Wright and what was known in the art, application-specific data transfer policies would include configurable permissions to modify attributes (e.g., attributes of file(s) being transferred by applications). See, e.g., SAMSUNG-1019, 41 (“Application traffic (at the end host) or incoming traffic from other hosts (at the

intermediate device) is identified by the classification mechanism and is forwarded to the appropriate queue awaiting service from other mechanisms such as the packet scheduler. The granularity level of the classification mechanism can be per-user, per-flow, or per-class depending on the type of QoS services provided. For example, per-flow QoS service requires per-flow classification while per-class QoS service requires per-class classification.”); SAMSUNG-1019, 44 (“The application or user can be uniquely identified by using user/application identification (ID).”), 50 (“Some applications may require low bandwidth and low delay. In this case WFQ will allocate high bandwidth to these applications in order to guarantee the low delay bound.” “Most multimedia application traffic (voice, video) is generated by a standard codec which generally provides certain knowledge of the quantitative traffic parameters.”).

246. A POSITA would understand that this disclosure teaches that QoS policies (which is an example of a data transfer policy) could operate at the application-level. For example, one configuration could be that the files associated with one application (such as streaming video) would receive priority over other applications (such as transferring Word documents). As specific file types are associated with specific applications, QoS policies at the application-level provide configurable permissions (which applications receive priority) relating to the attributes of files being transferred (i.e., large video files associated with a media

streaming application versus small documents associated with document processing applications).

247. Indeed, user-configurable policies enable flexibility in policy implementation and enforcement for particular scenarios. Thus, implementing such file attribute based permissions as part of application-specific policies would have amounted to a known technique to known systems to achieve the predictable results of configurable application settings for particular application policies. A POSITA would have been further motivated and would have been well within the skillset of a POSITA, given that Wright's discloses of its file attribute based policies and explains how permissions would be configured for policies.

[13] The wireless end-user device of claim 1, the one or more device agents further configured to encrypt Internet data packet traffic passing from the operating system packet network stack to the WWAN modem.

248. Wright discloses or renders obvious the one or more device agents further configured to encrypt Internet data packet traffic passing from the operating system packet network stack to the WWAN modem.

249. As described above, Wright teaches or renders obvious Internet data packet traffic passing from the operating system network stack to the WWAN modem. *See supra* analysis for elements [1.2] and [1.3].

250. A POSITA would have found obvious that Wright’s policy engine 832 (one or more device agents) would be *configured to encrypt at least some of this Internet data packet traffic*. In Wright, the *wireless end-user device* (client system 201) has an authorization module 245 that receives, stores, and transmits encrypted data (*e.g.*, an encrypted file or token). *See* SAMSUNG-1005, ¶¶66, 85, 135, 222. The “authorization module 245 provides authorization information to the authorization module 232 of the server system 200,” which a POSITA would understand is Internet packet data traffic (given that it is data traffic sent as packets over the Internet to the server system 200). SAMSUNG-1005, ¶66. In one example, “responsive to a network connection being activated, the authorization module 245 on the client device sends 334 sends a status request checking if new security information is available along with authorization information to the server system 200.” SAMSUNG-1005, ¶135. Wright’s “authorization module 245 may perform the cryptographic authentication protocol.” SAMSUNG-1005, ¶222; *see id.*, ¶85 (explaining that “[o]ne example of authorization information is an encrypted token”). Wright explains that “cryptographic authentication protocols” include “employing private keys, public keys or both” as well as “Secure Sockets Layer (SSL) certificates.” SAMSUNG-1005, ¶222. A POSITA understood that SSL/TLS connections are encrypted communications channels between an application running on a device to another endpoint on the Internet. SAMSUNG-1046, 1:38-42;

SAMSUNG-1047, 3. Such encrypted communications are decrypted at the application layer (e.g., inside a browser), and therefore, Internet data traffic is encrypted as it passes to/from the operating system and WWAN modem that Secure Sockets Layer (SSL) certificates.

251. Given these disclosures, a POSITA would have found obvious that the above-described packet data traffic would be encrypted according to a cryptographic protocol.

252. Moreover, as explained above, a POSITA would have found obvious to implement the authorization module as part of the diagnostic module of Wright's policy engine 832. *See analysis supra* for claim 5.

253. In view of the above, a POSITA would have found obvious that Wright's diagnostic module, which is part of the policy engine 832 (*one or more device agents*) is configured *to encrypt Internet data packet traffic from the operating system packet network stack to the WWAN modem*, which is subsequently transmitted to the server system 200/policy server 850.

[15] The wireless end-user device of claim 1, wherein the privilege to configure the application-specific traffic policy controls of the policy control agent is an exclusive privilege for one or more of the device agents executed in the protected execution partition.

254. As explained with reference to claim 1, Wright renders obvious that the *one or more of the device agents* (e.g., policy engine 832) has a *privilege to configure the application-specific traffic policy controls* (e.g., rules for or related to application-specific filtering or application-specific network access) of *the policy control agent* (e.g., packet filter engine 818, TDI filter 822). *See supra*, analysis of element [1.9]; SAMSUNG-1005, ¶¶268-269, FIG. 8.

SAMSUNG-1005, FIG. 8 (annotated)

255. In light of Wright’s disclosure, a POSITA would understand or find obvious that the above-described *privilege to configure the application-specific traffic policy controls* is *an exclusive* privilege. Wright teaches that the policy

engine 832 is the only entity that receives/downloads the policies from the policy server (*see* SAMSUNG-1005, ¶¶68, 244), and it is the only entity that provides the rules underpinning the policies to the TDI filter and packet filter engine, for enforcement of those policies (*see id.*, ¶¶239, 245; *see supra* Elements [1.7]-[1.9]). Given that no other entity in Wright is described as have the privilege to receive and configure policies that are enforced by the packet filter engine and the TDI filter, a POSITA would have understood that this is an exclusive privilege of Wright's policy engine.

256. Alternatively, a POSITA would have found it obvious for the policy engine 832 to have the exclusive privilege to receive and configure the application-specific policies enforced by the TDI filter and packet filter engine. A POSITA would have been motivated to limit or make exclusive the policy configuration privilege of the policy engine 832 to ensure that policies received from an external policy server are not corrupted or tampered with (*see* SAMSUNG-1005, ¶138), and in doing so, ensure proper operation of the mobile device. Indeed, a POSITA understood that when multiple system components have the capability to configure policies applicable to a device's operations, that increases the likelihood of the policy being corrupted and/or not being enforced correctly. Here a POSITA would understand that restricting access the creation policy to the policy engine 832 would represent the best practice isolating security functionality to a small, protected part

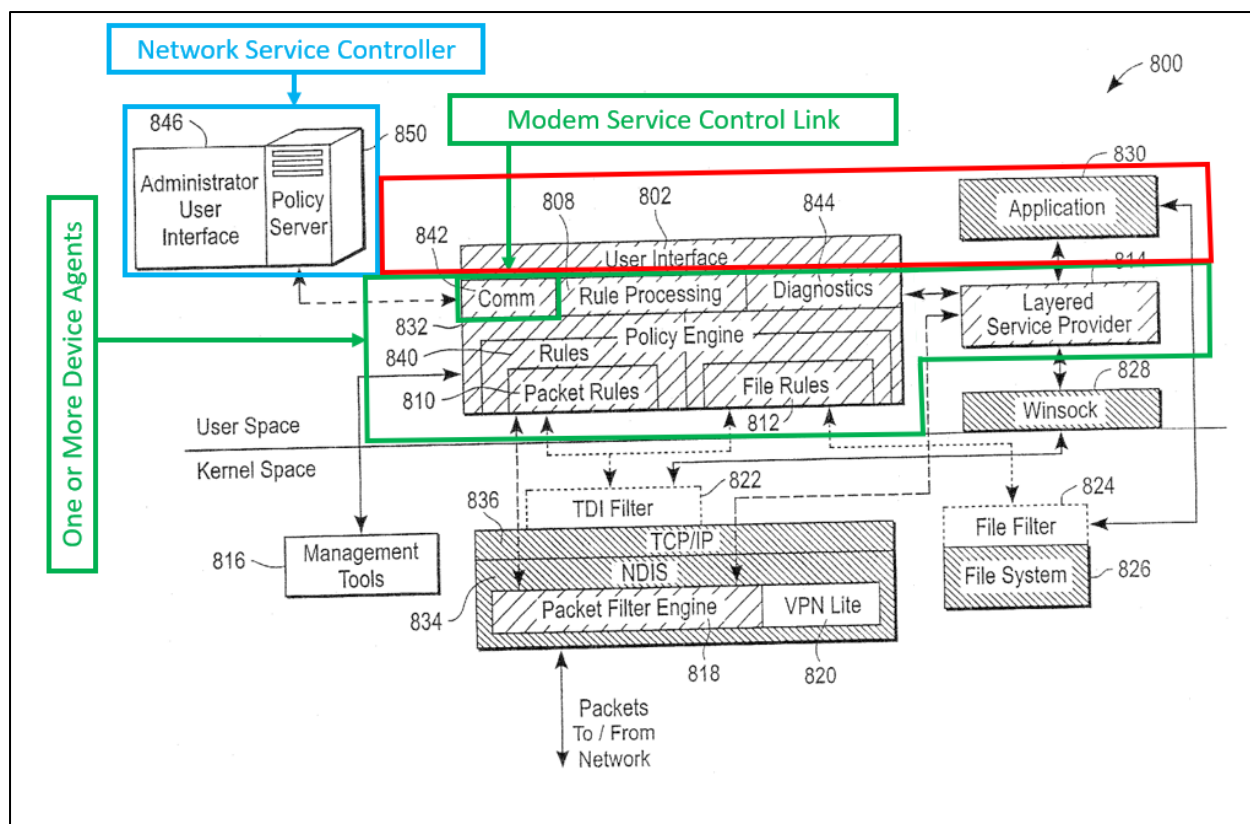
of the system (e.g., reduce the trusted computing base (TCB) to the smallest set of components possible; “Ideally, a system TCB should consist of the minimal amount of software necessary to enforce the security goals correctly.” SAMSUNG-1015, 6.).

257. Additionally, it was well-known in the context of policy management systems, for a single policy setting entity to perform the policy setting and configuration of the policies that are to be enforced, e.g., by a policy enforcement entity. *See, e.g.*, SAMSUNG-1005, ¶¶235-287 (describing policy engine 832), FIG. 8; SAMSUNG-1044, FIG. 2, ¶40 (“a firmware agent may set network filters to a predefined default policy, 210. The default policy may be, for example, to allow the host platform to access only specified network resources until an acceptable security policy can be verified.”). Thus, implementing Wright’s system to provide the policy engine 832 the exclusive privilege to perform the policy configuration would have amounted to an implementation of a known technique to a known system to achieve predictable results and functionality, as already performed by Wright’s system, and a POSITA would have thus had a reasonable expectation of success in so implementing Wright’s system.

[18]The wireless end-user device of claim 2, further comprising a user interface, the at least one device-assisted-services agent comprising a user-interface agent to display to a user, and allow modification to, at least one policy setting affecting at least one of the application-specific policy controls.

258. Wright teaches that the *wireless end-user device compris[es] a user interface*. Wright teaches sending information and instructions, via a user interface module, to “a screen on the mobile device.” SAMSUNG-1005, ¶¶82-83. Wright also provides examples of “graphical user interface[s]” that are displayed to the user and with which the user interacts. *See id.*, ¶¶27-34, 182-193, FIGS. 4-5. Wright describes one such interaction, in which “user interface module 218 receives input from a user input device such as a keyboard, mouse, or touchpad, and causes user interfaces to be displayed for use by a user for establishing criteria for defining an aspect of a security policy as allowed by permissions....” SAMSUNG-1005, ¶72; *see id.*, ¶82 (“client diagnostics module 246 may request that the user interface module 218 [to] cause the debug output to be displayed 1107 on a screen of the mobile device,” which “... allows the mobile device user to possibly assist in the diagnostic/debugging process”). Given these disclosures, a POSITA would understand that Wright’s mobile device includes a user interface. See FIGS. 4A-5F, ¶¶181-192. It was conventional that a user interface includes an input device (e.g., a screen) and software via which the user interacts with a computing device. *See, e.g.*, SAMSUNG-1005, ¶83 (“tests may be run that attempt to communicate to the outside world on blocked ports or an IP address may be pinged. Output may be saved in a debug output file and sent to a screen on the mobile computing device”); *id.*, ¶82, FIGS. 4A-5F, ¶¶181-192 (describing and illustrating various screens on a

mobile device that enable user input); see also SAMSUNG-1043, 607, FIG. 1 (“a graphical user interface application on the endpoint system 104 or a graphical user interface application on the host system 102, when users or local administrators of those computing devices are responsible for configuring their own policy settings locally.”). Note that a POSITA would understand that the user interface 802 would partially be external to the protected execution partition (PEP). User interfaces (UI) must interact closely with other parts of the Windowing system that communicates with all other user interfaces of other applications. Thus, the UI must run outside of a protected environment to allow the system to draw windows, receive input or other UI operations.



SAMSUNG-1005, FIG. 8 (annotated)

259. Wright’s user interface module (218, 702, 802) corresponds to **a user-interface agent that is configured display to a user at least one policy setting** (e.g., permissions and other user-configurable aspects of policies) **affecting at least one of the application-specific policies** (e.g., application-specific policies). See, e.g., SAMSUNG-1005, ¶228. Wright’s user interface module 218 enables “defining an aspect of a policy” (SAMSUNG-1005, ¶186) and “receives input from a user input device such as a keyboard, mouse, or touchpad, and causes user interfaces to be displayed for use by a user for establishing criteria for defining an aspect of a security policy as allowed by permissions....” SAMSUNG-1005, ¶72, 236. User

interface module 218 is configured to “set[s] 310 the permissions” that “typically relate to the allowable modification that may be made to a downloaded policy or client software by the client mobile device.” SAMSUNG-1005, ¶¶85, 186 (describing this function in the context of policy management module at the server side and explaining that policy setting module at the client device similarly implements aspects of this module). Similarly, in the context of FIG. 8, “policy Engine 832 receives user input and sends output via a communication interface with the user interface module 802 to display ... policy settings.” SAMSUNG-1005, ¶244.

260. Wright’s user interface module (user-interface agent) is configured to allow modification to at least one policy setting (e.g., permissions) affecting at least one of the application-specific policies (e.g., application-specific policies). In Wright, “policy setting module 212 may define criteria or if permissions set by the policy management module 236 allow, supplemental policy definitions or customization of policy definitions based upon user input processed by the mobile device user interface module 218.” SAMSUNG-1005, ¶68; see id., ¶¶85, 181 (“[a] notification of a change in policy, location, security feature or a combination of any of these may be sent by the policy setting module 212 or policy enforcement module 214 to the user interface module 218 which may then cause a notification indicating the change to be displayed in order to provide the user an opportunity to intervene

in the policy change if he desires and is allowed to do so”), 186. Similarly, “policy setting module 212 defines an aspect of a policy such as location criteria or security features criteria based upon user input processed by user interface module 218.” Id. Policy Engine 832, which provides functionality analogous to that of policy setting module 212, similarly “receives user input and sends output via a communication interface with the user interface module 802 to ...change policy settings responsive to user input.” SAMSUNG-1005, ¶244.

261. Wright explains that the user interface module has a communication interface to multiple system components, including policy setting module 212 (which provides functionality analogous to policy engine in FIGS. 7-8). SAMSUNG-1005, ¶¶72, 244, 228, 186, 188.

262. A POSITA would have understood the user interface module, together with its communication interface, correspond to one or more *device services assisted agent(s)*. That’s at least because these components together enable obtaining policy related information and displaying and/or modifying it—thus assisting in the device’s performance of services such as execution and enforcement of application-specific policies on the device.

263. Additionally, a POSITA would have found obvious to implement these components so that they *are executed within the application execution partition*. As a starting point, Wright teaches that the user interface module is in the user space

of the mobile device's environment, which as explained in claim 1 would be partitioned into protected execution partition and an application execution partition *See* SAMSUNG-1005, FIG. 8. Also, as explained above, the user interface module facilitates displaying of interfaces and the associated interaction of users with these interfaces. A POSITA would thus recognize that the services offered by the user interface module would not only facilitate display and interactions with user interface screens corresponding to policy management functions of the device, but would also enable such functions for other device applications, such as applications 830. In this regard, FIG. 2B shows the user interface module as a separate component from the group of components (245, 246, 208, 210, 212, and 214) that are responsible for the policy management and diagnostic information collection functionality of the system. Given the more general applicability of the user interface module, a POSITA would have sought to implement it as part of the application execution partition and not within the protected execution partition that provides only the policy management and diagnostic-related functions of Wright. Such an implementation would also be obvious to try given that there would be two partitions within user space within which the user interface module could be implemented. Moreover, user interface agents/modules typically do not need secure environments in which they are implemented and are often implemented in the less secure execution environments within which applications execute. *See, e.g.,*

SAMSUNG-1042, 54 (“in Unix, the GUI (e.g., Windows and Motif) runs entirely in the user space, so the only system calls needed for writing on the screen are write and a few other minor ones. Of course, there are larger number of calls to X Windows and the GUI, but these are not system calls in any sense.”).

264. A POSITA would understand that software for providing a user interface is, by design, to allow users to interface with the operating system. The mechanism by which a user interfaces with the operating system through the user interface is via system calls. Finally, a POSITA would know that interfacing with the operating system via system calls necessarily means the software (that wishes to interface with the operating system, e.g., user interface software) is running in user space (because if the software was running in kernel space, it would not need to leverage system calls to execute code within kernel space—it would just call that code directly if it was already in kernel space). *See, e.g.,* SAMSUNG-1015, 7. (“Since user processes are untrusted, we cannot depend on them, but we can protect them from threats. For example, secure operating system can prevent a user process with access to secret data from leaking that data, by limiting the interactions of that process.”) *see also e.g.,* SAMSUNG-1025, 42 (“A second strategy for interfacing with the operating system is through a user-friendly graphical user interface or GUI. Rather than having users directly enter commands via a command-line interface, a GUI allows provides a mouse-based window-and-menu system as an interface.”);

see also e.g., SAMSUNG-1025, 43 (“System calls provide an interface to the services made available by an operating system.”), 18 (“However, when a user application requests a service from the operating system (via a system call), it must transition from user to kernel mode to fulfill the request.”). Thus, implementing the user interface module in the execution environment where applications 830 are executed would therefore be an implementing of known techniques to known systems to achieve the predictable result of enabling user interface functionality for all types of applications (those with higher privileges or not) on the computing device.

B. Ground 1B: Combination of Wright and Tzannes Renders Obvious Claims 12, 16, and 17

[12] The wireless end-user device of claim 1, the one or more device agents further configured to calculate, separate from the traffic information, a service usage measurement for the Internet data packet traffic passing between the operating system packet network stack and the WWAN modem.

265. As described in §X.C, Wright-Tzannes would, in some implementation, include packet counting functionality at LSP 814 (*one or more device agents*). As also described in §X.C, such packet counting functionality (as provided by packet counter module 260 in Tzannes and as implemented at LSP 814 in the combination) includes calculating a packet count value that is appended to the packet being transmitted. SAMSUNG-1006, ¶212. As Tzannes explains:

packet retransmission is enabled by “transmitting packets with an appended bit field that contains a counter indicating the place of each packet in a stream of packets. This counter value is also known as a Sequence ID (SID). For example, a bit field containing a 16-bit counter could be appended to each packet and the counter module 260 would be incremented by one after each packet was transmitted. In cooperation with the packet assignment module 240, a packet counter field could be appended to the packet in a number of places, for example, at the beginning or end of the packet, or at the beginning or end of the packet header. SAMSUNG-1006, ¶216.

266. The packet count calculations corresponds to a *service usage measurement* for the transmitted packets because it provides a calculation of the place of each packet in the stream of packets (and by extension, an amount of transmitted traffic) being transmitted. This calculation as implemented at LSP 814 is *separate from* the previously-calculated *traffic information* that includes the previously-calculated measured amount by the TDI filter/packet filter engine (*see supra* analysis for element [1.6], [1.10]).

267. A POSITA would understand from this disclosure that traffic measurement models commonly use the “number of packets” to associate traffic measurement (service usage measurement) with an accountable entity (i.e., in the

context of cellular networks, a subscriber). *See, e.g.,* SAMSUNG-1021, 4 (“At the heart of the traffic measurement model are network entities called traffic METERS. Meters count certain attributes (such as numbers of packets and bytes) and classify them as belonging to ACCOUNTABLE ENTITIES using other attributes (such as source and destination addresses). An accountable entity is someone who (or something which) is responsible for some activity on the network. It may be a user, a host system, a network, a group of networks, etc, depending on the granularity specified by the meter's configuration.”)

268. Moreover, a POSITA would have understood that this packet count calculation, as performed by LSP 814 in the Wright-Tzannes combination, is a service usage measurement *for the Internet data packet traffic passing between the operating system packet network stack and the WWAN modem*. That’s because (1) the packet traffic being transmitted in Wright is Internet data packet traffic transmitted from the OS packet network stack to the WWAN (*see supra* analysis for [1.2]), and (2) Tzannes further teaches that packets for transmission are received, from higher layers, and are for delivery by the modem/transceiver to external devices. SAMSUNG-1006, ¶¶192-216.

[16] The wireless end-user device of claim 1, the WWAN modem comprising a modem execution partition configured to calculate, separate from the traffic information, a service usage measurement for the Internet data packet traffic

passing between the operating system packet network stack and the WWAN modem.

269. As explained in §X.C *supra* and in claim 12, Wright-Tzannes packet counting functionality (as provided by packet counter module 260) enables calculation of a packet count value that is appended to the packet being transmitted and corresponds to a *service usage measurement*. Additionally, this packet count value, which is different and *separate from the traffic amount* as described in element [1.6], is a measurement *for the Internet data packet traffic passing between the operating system packet network stack and the WWAN modem*, as explained in claim 12 *supra*.

270. Moreover, as explained in §X.C, a POSITA would have found obvious to implement the components of Tzannes modem/transceiver, including the packet counter module 260, as part of the WWAN modem in the Wright-Tzannes combination. Additionally, in the relevant timeframe, it was well-known for a mobile device's modem to be implemented using a separate, dedicated processor (i.e., separate from the general processor of the device) with its own dedicated execution environment. Wright teaches that its WWAN modem is implemented using a network interface card (NIC), which typically includes a specialized processor that is different from the general processor of the mobile device. *See, e.g., SAMSUNG-1028*, 1 (“Multimedia telephony is a delay-sensitive application. Packet losses, relatively less critical than delay, are allowed up to a certain threshold. They

represent the QoS constraints that have to be respected to guarantee the operation of the telephony service and user satisfaction. In this work we introduce a new smartphone architecture characterized by two process levels called application processor (AP) and mobile termination (MT), respectively. Here, they communicate through a serial channel.”); *see, e.g.*, SAMSUNG-1028, 2 (“Two processors can be distinguished called, respectively, AP (application processor) and MT (mobile termination).”); SAMSUNG-1031, 748 (“It integrated a 3G WCDMA digital baseband processor (DBB) and an application processor (AP) on a single chip.”); SAMSUNG-1032, 260 (“In the application portion, an ARM1176 runs the OS and a SH-X2 runs realtime media applications. In the baseband, an ARM926 controls the baseband IPs including DSPs that handle modem protocols”). A POSITA would have thus understood or found obvious that Wright’s WWAN modem would provide a specialized and separate execution environment (a modem execution partition). A POSITA would have thus understood or found obvious that the above-described packet counter module would be implemented within Wright’s modem execution partition.

[17] The wireless end-user device of claim 16, the modem execution partition further comprising a modem control link to communicate the calculated service usage measure to at least one of a network service controller and at least one device agent executed in the protected execution partition.

271. As explained for claim 16 and in §X.C *supra*, Wright-Tzannes teaches a WWAN modem with a ***modem execution partition*** that would implement the components and functionality of Tzannes' modem/transceiver. *see supra* analysis for claim 16 and §X.C. One such component is the transmission management module 220/320 (*see id.*), which is responsible for transmitting packets appended with the calculated packet count values (***service usage measure***, per [16] *supra*) to other computing devices. SAMSUNG-1006, ¶¶212, 214, claims 72, 85, FIG. 1, 192.

272. Moreover, a POSITA would have understood that the transmission management module is a ***modem control link*** because it facilitates the communication with external devices (and their respective modems), and controls the packets that are so transmitted by the combination's WWAN modem in Wright-Tzannes over communication channel 10 (as depicted in Tzannes), which would be akin to the communication channel between communication port 842 and policy server 850 in Wright.

273. Additionally, Tzannes teaches communicating with other device transceivers/modem, just as Wright similarly teaches packet transmission between the device and other devices, including policy server 850 (***network service controller***, per analysis for claim 7). In combination, a POSITA would have understood or found obvious that the modem's transmission management module of the modem facilitates network packet transmissions to the policy server (i.e., *a*

network service controller). A POSITA would have been motivated to so implement Wright-Tzannes' system and would have had a reasonable expectation of success in doing so, for the reasons described in §X.C *supra* and at least the following additional reasons.

274. **First**, a POSITA would recognize that communicating the *calculated service usage measure* to the Wright's *network service controller* (policy server 850) would enable any subsequent packet retransmission, subject to applicable QOS rules, that may be requested by the policy server 850 (which may not have made it to the server). For example, Tzannes explains that “[o]nce a packet(s) is found to be in error, there are several exemplary ways in which a receiving modem can communicate information to the transmitting modem indicating that a retransmission of one or more packets is required.” SAMSUNG-1006, ¶228.

275. **Second**, because Wright's mobile device sends and receives (to/from policy server 850) critical policy information and diagnostic information that controls the operation of Wright's device (*see supra* §§X.B-X.C; SAMSUNG-1005, ¶¶284, 238), a POSITA would have been motivated to implement Tzannes' teachings in the combination to facilitate reliable network communications between Wright's mobile device and its policy server (i.e., communications that are not subject to packet loss). These modifications, i.e., configuring permissions to limit

access exclusively to Wright's policy engine 832, would have been relatively straightforward and well within skill of a POSITA.

276. For these reasons, Wright-Tzannes renders obvious the *modem execution partition* comprising a *modem control link* (e.g., transmission management module) to communicate *the calculated service usage measure* (e.g., determine or calculate a value, such as a "packet count value,") *to at least one of a network service controller* (e.g., policy server 850).

C. Ground 1C: Wright in view of Smith renders claims 14 obvious

[14] The wireless end-user device of claim 1, wherein the protected execution partition comprises a secure virtual execution environment.

277. As explained in [1.3] *supra*, LSP 814 and policy engine 832 would be implemented as part of *protected execution partition (PEP)*. As explained in §X.E *supra*, it would be obvious to implement the protected execution partition using one or more virtual machines, thus rendering obvious the protected execution partition as *comprising a secure virtual execution environment*.

XII. CONCLUSION

278. For all the reasons I have noted in the foregoing paragraphs, claims 1-18 of the '144 patent are obvious in view of the references discussed above.

279. 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.

APPENDIX A

PATRICK DREW MCDANIEL

Tsun-Ming Shih Professor of Computer Sciences
School of Computer, Data and Information Sciences ◊ University of Wisconsin-Madison
Office : Room 7387, 1210 W. Dayton St. ◊ Madison, WI 53706 ◊ (608) 263-1008
email: mcdaniel@cs.wisc.edu ◊ *Homepage*: <https://patrickmcdaniel.org/>

ACADEMIC AND RESEARCH APPOINTMENTS

Tsun-Ming Shih Professor of Computer Sciences, University of Wisconsin-Madison, Madison, WI	2022-Present
William L. Weiss Professor of Information and Communications Technology Computer Science and Engineering, Pennsylvania State University, University Park, PA	2017-2022
Distinguished Professor Computer Science and Engineering, Pennsylvania State University, University Park, PA	2016-2017
Professor Computer Science and Engineering, Pennsylvania State University, University Park, PA	2011-2015
Associate Professor Computer Science and Engineering, Pennsylvania State University, University Park, PA	2007-2011
Hartz Family Career Development Assistant Professor Computer Science and Engineering, Pennsylvania State University, University Park, PA	2004-2007
Adjunct Professor Stern School of Business, New York University, New York, NY	2003-2009
Senior Research Staff Member AT&T Labs - Research, Florham Park, NJ	2001-2004

RESEARCH LEADERSHIP APPOINTMENTS

Director, National Science Foundation, Center for Trustworthy Machine Learning Participants: Penn State, Stanford, UC Berkeley, UC San Diego, Univ. of Wisconsin, Univ. of Virginia	2018-present
Director, Institute for Networking and Security Research College of Engineering, Pennsylvania State University, University Park, Pennsylvania	2016-2022
Program Manager, Cyber-Security Collaborative Research Alliance (CRA) College of Engineering, Pennsylvania State University, University Park, Pennsylvania	2013-2018
Co-Director, Systems and Internet Infrastructure Laboratory College of Engineering, Pennsylvania State University, University Park, Pennsylvania	2004-2022

EDUCATION

Ph.D., Computer Science and Engineering, University of Michigan , Ann Arbor, MI	2001
M.S., Computer Science, Ball State University , Muncie, IN	1991
B.S., Computer Science Ohio University , Athens, OH	1989

AFFILIATIONS

Association for Computing Machinery (ACM), *Fellow*
The Institute of Electrical and Electronics Engineers (IEEE), *Fellow*
American Association for the Advancement of Science (AAAS), *Fellow*
USENIX Advanced Computing Systems Association (USENIX)

STUDENTS

Past PhD Students

- *Z. Berkay Celik*, Spring 2019, now **Assistant Professor**, Purdue University

-
- *Nicolas Papernot*, Spring 2018, now **Assistant Professor**, University of Toronto
 - *Wenhui Hu*, Fall 2016, now **Senior Member of Technical Staff**, Oracle
 - *Devin Pohly*, Spring 2016, now **Assistant Professor**, Wheaton College
 - *Damien Oceau*, Summer 2014, now **Software Engineer in Security**, Google
 - *Steve McLaughlin*, Spring 2014, now **Senior Software Engineer**, Samsung Research America
 - *Thomas Moyer*, Summer 2011, now **Assistant Professor**, University of North Carolina-Charlotte
 - *William Enck*, Spring 2011, now **Professor**, North Carolina State University
 - *Kevin Butler*, Summer 2010, now **Professor**, University of Florida
 - *Machigar Ongtang*, Summer 2010, now **Assistant Professor**, Dhurakij Pundit University
 - *Patrick Traynor*, Spring 2008, now **John and Mary Lou Dasberg Preeminent Chair, Professor**, University of Florida, co-advisor
 - *Fr. Boniface Hicks*, Fall 2007, now **Assistant Professor**, St. Vincent College

Current PhD Students

- *Jean-Charles Noirot Ferrand*, University of Wisconsin, Spring 2028
- *Kyle Domico*, University of Wisconsin, Spring 2028
- *Kunyang Li*, University of Wisconsin, Spring 2027
- *Rachel King*, University of Wisconsin, Spring 2027
- *Yohan Beugin*, University of Wisconsin, Spring 2026
- *Blaine Hoak*, University of Wisconsin, Spring 2026
- *Quinn Burke*, University of Wisconsin, Spring 2026
- *Eric Pauley*, University of Wisconsin, Spring 2025
- *Ryan Sheatsley*, University of Wisconsin, Spring 2024

Past Masters Students

- *Blaine Hoak*, Pennsylvania State University, Summer 2022
- *Rachel King*, Pennsylvania State University, Summer 2022
- *Alban Heon*, Pennsylvania State University, Summer 2022
- *Yohan Beugin*, Pennsylvania State University, Spring 2021
- *Ahmed Abdou*, Pennsylvania State University, Spring 2021
- *Adrien Cosson*, Pennsylvania State University, Summer 2020
- *Bolor Zolbayar*, Pennsylvania State University, Summer 2020
- *Alejandro Andrade*, Pennsylvania State University, Summer 2020
- *Quinn Burke*, Pennsylvania State University, Spring 2020
- *Eric Pauley*, Pennsylvania State University, Spring 2020
- *Sushrut Shringarputal*, Pennsylvania State University, Fall 2019
- *Raquel Alvarez*, Pennsylvania State University, Spring 2019
- *Valentin Vie*, Pennsylvania State University, Spring 2019
- *Ryan Sheatsley*, Pennsylvania State University, Fall 2018
- *Eric Kilmer*, Pennsylvania State University, Spring 2016
- *Nathan Lagerman*, Pennsylvania State University, Spring 2016
- *Matthew Dering*, Pennsylvania State University, Spring 2014
- *Phil Koshy*, M.S. Pennsylvania State University, Fall 2013
- *Diana Koshy*, M.S. Pennsylvania State University, Fall 2013
- *Steve McLaughlin*, M.S. Pennsylvania State University, Spring 2011
- *Sergei Miadzvezhanka*, M.S. Pennsylvania State University, Spring 2011
- *Adam Delozier*, M.S. Pennsylvania State University, Spring 2011
- *Juliet Uhloft*, M.Eng. Pennsylvania State University, Fall 2010

-
- *Damien Octeau*, M.S. Pennsylvania State University, Spring 2010
 - *Thomas Moyer*, M.S. Pennsylvania State University, Spring 2009
 - *Luke St. Clair*, M.S. Pennsylvania State University, Summer 2008
 - *Lisa Johansen*, M.S. Pennsylvania State University, Spring 2008
 - *Sunam Ryu*, M.S. Pennsylvania State University, Spring 2007
 - *Dhananjay Bapat*, M.S. Pennsylvania State University (Electrical Engineering), Fall 2006
 - *Jennifer Plasterr*, M.Eng. Pennsylvania State University, Summer 2006
 - *Adam Kerr*, M.Eng. Pennsylvania State University, Fall 2006
 - *William Enck*, M.S. Pennsylvania State University, Spring 2006
 - *Wesam Lootah*, M.S. Pennsylvania State University, Spring 2006
 - *Jon Hansford*, M.Eng. Pennsylvania State University, Fall 2005
 - *John van Bremer*, M.Eng. Pennsylvania State University, Spring 2005

Past Post-Docs

- *Vaibhav Rastogi*, graduated Northwestern University
- *Robert Walls*, graduated University of Massachusetts, Amherst

TEACHING

University of Wisconsin-Madison, College of Letters and Science

- **Intro to Information Security** - Fall 2023

Pennsylvania State University, College of Engineering

- **CMPSC311 - Introduction to Systems Programming** - Fall 2013, Fall 2014, Fall 2015, Fall 2016, Summer 2019, Spring 2020, Fall 2020, Fall 2021
- **CMPSC443 - Introduction to Computer and Network Security** - Spring 2006, Spring 2009, Fall 2017, Fall 2018
- **CSE543 - Computer and Network Security** - Fall 2004, Fall 2005, Fall 2008, Fall 2009, Fall 2011, Fall 2014
- **CSE544 - Advanced System Security** - Spring 2005, Spring 2007
- **CSE545 - Advanced Network Security** - Spring 2006, Spring 2008, Spring 2011
- **CSE597g - Principles, Analysis, and Applications of Computer Security** - Fall 2015
- **Security and Privacy of Machine Learning** - Fall 2016
- **Advanced Topics in the Security and Privacy of Machine Learning** - Spring 2017
- **CSE598 - Cell Phone Operating Systems** - Spring 2009
- **CSE598i - Web 2.0 Security** - Spring 2010
- **CSE598d - Topics in Applied Systems Security** - Fall 2010
- **CSE598e - Critical Infrastructure Security** - Fall 2011
- **CSE 597 – Emerging Trends in Computer Security** - Fall 2021
- **CMPSC297 – Introduction to C Programming in UNIX** - Fall 2021

New York University, Stern School of Business

- **B20.3157 Computer and Network Security** - Spring 2003, Summer 2004, Summer 2005
- **B20.3156 - Online Privacy** - Spring 2003, Summer 2004

HONORS, AWARDS, AND KEYNOTE ADDRESSES

SIGSAC Outstanding Innovation Award, for innovative research in mobile device security, trustworthiness of machine learning, and systems security, November 2021

Penn State Engineering Society Premier Research Award, Given to one faculty member per year, the Penn State Engineering Alumni Society Premier Research Award recognizes and rewards an individual whose contributions to scientific knowledge through research are exemplary and internationally acclaimed, April 2021

AAAS Fellow, for distinguished contributions to the field of computational security and privacy, particularly for advancing algorithms for the formal analysis of mobile devices and applications, November 2020

SIGOPS Hall of Fame Award, recognizing the paper "TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones" (Will Enck first author), "which sparked an important research agenda on smartphone privacy that continues to this day", November 2020

J. D. Williams student paper award, Nuclear Security and Physical Protection division, recognizing the best student papers by area in Proceedings of the Institute of Nuclear Materials Management Annual Meeting (INMM), July 2019

Penn State Engineering Society Outstanding Advising Award, highly selective award by the Penn State Engineering Society given to faculty in the College of Engineering who have made significant contributions as advisor, October 2018

Best Paper, 2017 EAI SECURECOMM 2018, with Sayed M. Saghaian, Tom La Porta, Trent Jaeger and Z. Berkay Celik, August 2018

Best Student Paper, 2017 ACM Symposium on SDN Research (SOSR), with Stefan Achleitner, Thomas La Porta and Trent Jaeger, April 2017

IEEE Technical Committee on Security and Privacy Outstanding Community Service Award, in recognition for leadership of the Technical Committee on Security and Privacy, May 2016

ACM Fellow, for contributions to computer and mobile systems security, December 2015

Science of Security Index of Significant Research in Cyber Security, acknowledging paper 'Toward a Science of Secure Environments', Science of Security Virtual Organization (SOS-VO), August 2015

IEEE Fellow, for contributions to the security of mobile communications, November 2014

Best Artifact Award, 20th International Symposium on the Foundations of Software Engineering (FSE), with advisee Damien Ocateau and collaborator Somesh Jha, November 2012

Best Paper, 25th Annual Computer Security Applications Conference, with advisees Machigar Ongtang, Stephen McLaughlin, and William Enck, December 2009

Faculty Marshal, College of Engineering, selected by student marshals for contributions to undergraduate education, leads procession into graduation ceremony, May 2009

Penn State Engineering Society Outstanding Research Award, highly selective award by the Penn State Engineering Society given to faculty in the College of Engineering who have made significant contributions to knowledge in their field, March 2009

Google Security and Product Safety Acknowledgement, in recognition of efforts in improving the security of Google Android cellular phone operating system, 2008

Commendation for Exceptional Leadership and Achievement, in recognition of efforts as PI of the EVEREST study, from Ohio Secretary of State Jennifer Brunner, August 2008

IEEE Technical Committee on Security and Privacy Outstanding Community Service Award, in recognition for technical program management of the IEEE Security and Privacy symposia, August 2008

National Science Foundation CAREER Award, Faculty early career development grant, August 2007

Penn State Computer Science and Engineering Outstanding Teaching Award, Given to best teacher in the department as selected by students, March 2007

ACM Certificate of Meritorious Service, Certificate acknowledging exemplary service as associate editor of ACM Transactions on Internet Technologies, April 2007

Best Student Paper, 22nd Annual Computer Security Applications Conference, as advisor, with Boniface Hicks and Kiyan Ahmadizadeh, December 2006

Best Paper, Innovations and Commercial Applications of Distributed Sensor Networks Symposia, Awarded for best paper in conference, October 2005.

Hartz Family Career Development Professor, Endowed Professorship, Pennsylvania State University, Fall 2004-2007

Bang for the Buck Award, DARPA Dynamic Coalitions Program, Award for most feature-rich/useful software system, April 2002

National Aeronautics and Space Administration, Kennedy Space Center Fellowship, Research Fellowship, September 1997 - August 2000

Electrical Engineering and Computer Science Summer Fellowship Award, University of Michigan, June 1997

Dean's Citation for Perfect Academic Record, Ball State University, June 1991

Keynote Addresses

1. Security's Role in Achieving Sustainability, 29th ACM Conference on Computer and Communications Security (CCS), Los Angeles, CA, November, 2022.
2. Security, Game Theory, and Their Role in Achieving Sustainability, Conference on Decision and Game Theory for Security, Pittsburgh, PA, October, 2022.
3. Prognosticating the Future of IoT Security, 2022 IEEE SafeThings Workshop, San Francisco, CA, May 2022.
4. The Challenges of Machine Learning in Adversarial Settings. Triangle Area Privacy and Security Day, Durham, NC, October, 2019.
5. The Challenges of Machine Learning in Adversarial Settings. 2019 Subversion and Assurance of AI Workshop, US National Reconnaissance Office, Washington, DC, March, 2019.
6. Attacks, Defenses, and Impacts of Machine Learning in Adversarial Settings. 2017 Conference on Security and Privacy in Communication Networks (SecureComm), Niagara Falls, Canada, October, 2017.
7. Tracing the Arc of Smartphone Application Security. 2017 ACM on International Workshop on Security And Privacy Analytics. Scottsdale, AZ, March, 2017.
8. Tracing the Arc of Smartphone Application Security. 12th International Conference on Information Systems Security, Jaipur, India, December, 2016.
9. The 25th International Conference on Computer Communication and Networks (ICCCN 2016), August, 2016, Waikoloa, Hawaii.
10. Learning from Ourselves: Where are we and where can we go in mobile systems security?. Mobile Security Technologies (MOST) 2016 Workshop, IEEE Computer Society Security and Privacy Workshops, San Jose, CA, May, 2016.
11. Eight Years of Mobile Smartphone Security. Center for Secure and Dependable Systems (CSDS) Cybersecurity Symposium, Coeur d'Alene, April, 2016.
12. The Importance of Measurement and Decision Making to a Science of Security, 2015 IEEE Conference on Communications and Network Security (CNS), September 2015, Florence, Italy.
13. The Importance of Measurement and Decision Making to a Science of Security. 3rd International Symposium on Resilient Cyber Systems, Philadelphia, PA, August, 2015.
14. The Importance of Measurement and Decision Making to a Science of Security, 2015 Symposium And Bootcamp on the Science of Security (Hotsos), April 2015, University of Illinois at Urbana-Champaign

-
15. Security and Science of Agility, ACM Workshop on Moving Target Defense (MTD 2014), November 2014, Scottsdale, AZ
 16. A Secondary Internet Revolution: How the Smart Device has Changed the Information Security Landscape, IEEE New Technology Industry Seminar (NTIS '13), Everett WA, August, 2013
 17. Permission-based Application Governance; A Step Forward or Backward?, 26th Annual WG 11.3 Conference on Data and Applications Security and Privacy (DBSec'12), Paris, France, July 2012.
 18. Scalable Integrity-Guaranteed AJAX, The 14th Asia-Pacific Web Conference (APWeb), Kunming, China, April 2012.
 19. Security Challenges and Solutions in Mobile Smartphone Applications, IEEE Computer Security Foundations (CSF 2011), Abbaye des Vaux de Cernay, France, June 2012.
 20. Password Exhaustion: Predicting the End of Password Usefulness. 2nd International Conference on Information Systems Security, Kolkata India, December, 2006.
 21. Physical and Digital Convergence: Where the Internet is the Enemy. Eighth International Conference on Information and Communications Security (ICICS '06), Raleigh, NC, December, 2006.

Distinguished Lectures

1. The Challenges of Machine Learning in Adversarial Settings: A Systems Perspective, Michigan State University, Lansing, MI, November, 2023.
2. The Challenges of Machine Learning in Adversarial Settings: A Systems Perspective. Computer Science Department, Temple University, Philadelphia, PA, February, 2022.
3. The Challenges of Machine Learning in Adversarial Settings: A Systems Perspective. Computer Science Department, University of Wisconsin-Madison, Madison, WI, February, 2020.
4. Shutterstock Distinguished Lecture: The Challenges of Machine Learning in Adversarial Settings. Computer Science Department, Stonybrook University, Stonybrook, NY, December, 2019.
5. Distinguished Blockchain Lecture: The Challenges of Machine Learning in Adversarial Settings. Cylab Security and Privacy Institute, Carnegie Mellon University, Pittsburgh, PA, December, 2019.
6. Distinguished Speaker Series: The Challenges of Machine Learning in Adversarial Settings. Department of Computer Science, University at Buffalo, Buffalo, NY, November, 2018.
7. Samuel D. Conte Distinguished Lecture Series: The Challenges of Machine Learning in Adversarial Settings. Department of Computer Science, Purdue University, West Lafayette, Indiana, November, 2018.
8. The Challenges of Machine Learning in Adversarial Settings. Department of Software and Information Systems, University of North Carolina at Charlotte, Charlotte, NC, February, 2018.
9. Tracing the Arc of Smartphone Application Security. Celebrating 50 Years of Computer Science @ NC State, North Carolina State University, Raleigh, NC, October, 2017.
10. Tracing the Arc of Smartphone Application Security. Computer Science Department and the Electrical and Computer Engineering Department Seminar Series, Colorado State University, Fort Collins, CO, October, 2017.
11. Tracing the Arc of Smartphone Application Security. Rochester Institute of Technology, College of Computing and Information Sciences, Rochester, NY, September, 2017.
12. Tracing the Arc of Smartphone Application Security, University of Texas-Dallas Department of Computer Science, Dallas, TX, May 2017.
13. Tracing the Arc of Smartphone Application Security. Rochester Institute of Technology, College of Computing and Information Sciences, Rochester, NY, May 2017.

-
14. Tracing the Arc of Smartphone Application Security. University of California-Irvine, Computer Science Department, Irvine CA, March, 2017.
 15. Tracing the Arc of Smartphone Application Security. The Ohio State University, Department of Computer Science and Engineering, Columbus, OH, March, 2017.
 16. Tracing the Arc of Smartphone Application Security. Virginia Technical University, Department of Computer Science, Blacksburg, VA, March, 2017.
 17. Six Years of Mobile Smartphone Security, CISPA Distinguished Lecture Series, Max Planck Institute/Saarland University, Saarbrücken Germany, July, 2015.
 18. Six Years of Mobile Smartphone Security. Technische Universität Darmstadt, Darmstadt Germany, July, 2015.
 19. Security Challenges and Solutions in Mobile Smartphone Applications. Computer and Information Science Department, University of Oregon, Eugene, OR, April, 2011.
 20. Security Challenges and Solutions in Mobile Smartphone Applications. Department of Software Information Systems College of Computing and Informatics, UNC Charlotte, Charlotte, NC, December, 2010.

RESEARCH SUPPORT

co-PI, *MURI: Cohesive and Robust Human-Bot Cybersecurity Teams*, Army Research Office, \$6,000,000 (PSU award \$739,527), 07/01/2021-06/30/2026, Collaborators: Many.

co-PI, *SaTC: CORE: Small: Adversarial Network Reconnaissance in Software Defined Networking*, NSF (CNS), \$500,000 (PSU award \$500,000), 1/1/2020-12/31/2022, Collaborators: He (Penn State).

PI, *CNS Core: Medium: Automated IoT Safety and Security Analysis and Synthesis*, NSF (CNS), \$272,033 (PSU award \$272,033), 6/25/2019-6/24/2022, Collaborators: Tan (Penn State).

PI, *Mapping Black-Box Attack Metrics and Parameter Spaces in Machine Learning*, US Army Aviation and Missile Research, Development and Engineering Center, \$436,677 (PSU award \$436,677), 6/25/2019-6/24/2022, Collaborators: (single PI).

PI, *SaTC CORE: Frontier: Collaborative: End-to-End Trustworthiness of Machine-Learning Systems*, NSF (CNS), \$9,649,366 (PSU award \$2,044,550), 8/15/2018-3/31/2023, Collaborators: Boneh (Stanford), Chaudhuri (UCSD), Evans (Virginia), Jha (Wisconsin), Liang (Stanford), Song (Berkeley).

PI, *2017 SaTC PI Meeting*, NSF (CNS), \$99,999 (PSU award \$50,230), 8/15/2016-3/31/2017, Collaborators: Antonakakis (GaTech), Mason (UIUC).

PI, *TWC: Medium: Collaborative: Scaling and Prioritizing Market-Sized Application Analysis*, NSF (CNS), \$1,147,213 (PSU award \$547,213), 7/01/2016-6/30/2020, Collaborators: Jha (Wisconsin).

PI, *Student Travel Support for Symposium on Security and Privacy 2014*, Army Research Office, \$10,000 (PSU award \$10,000), 5/1/14-5/1/15.

PI, *Models for Enabling Continuous Reconfigurability of Secure Missions (MACRO) Cyber-Security Collaborative Research Alliance (CRA)*, Army Research Laboratory, \$24.1 million (\$48.2 million with renewal at 12/17), 9/20/2013-9/19/2023 (renewed at 5 years), Collaborators: PSU, Carnegie Mellon, Indiana, UC Davis, UC Riverside, ARL, CERDEC.

PI, *Google Faculty Research Award, Plotting a Map of Android Inter-App Communication*, Google, \$50,000, 3/1/2012-2/28/2013, Collaborators: PSU (McDaniel), TU Darmstadt (Bodden), University of Luxembourg (Traon), .

PI, *Battelle BGP Security Study (Phase 2)*, Battelle, \$102,815, 10/1/2012-9/30/2013, Collaborators: PSU (McDaniel), Oregon (Butler).

PI, *TWC: Medium: Collaborative: Extending Smart-Phone Application Analysis*, NSF (CNS), \$1,386,518 (plus 16k REU supplement) (PSU award \$534,748), 8/1/2012-7/31/2016, Collaborators: PSU (McDaniel), Wisconsin (Jha).

PI, *Battelle BGP Security Study (Phase 1)*, Battelle, \$94,400, 2/15/2012-9/30/2012, Collaborators: PSU (McDaniel).

co-PI, *TC: Medium: Collaborative Research: Building Trustworthy Applications for Mobile Devices*, NSF (CNS), \$1,386,518 (PSU award \$350,000), 8/1/2011-7/31/2014, Collaborators: PSU (McDaniel), Wisconsin (Banerjee, Jha, Swift).

PI, *Closing the Loop on Security Testing and Security Requirements*, Security and Software Engineering Research Center, \$31,000, 8/1/2011-7/31/2012.

co-PI, *Managing Security and Vulnerability Risks in the Smart Grid*, Institute for CyberScience and The Penn State Institutes of Energy and the Environment, \$31,000, 08/1/09-12/16/09, Collaborators: PSU (Blumsack, McDaniel).

PI, *Smart Grid Cyber Security Research*, Lockheed Martin, \$250,000, 1/1/10-12/16/10.

PI, *NSF HECURA: Collaborative Research: Secure Provenance in High-End Computing Systems*, NSF (CCF), \$1,000,000 (PSU award \$307,073), 08/1/09-8/31/13, Collaborators: PSU (McDaniel), UIUC (Winslett), Stonybrook (Sion, Zadok).

PI, *TC: Medium: Collaborative Research: Security Services in Open Telecommunications Networks*, NSF (CNS), \$1,386,518 (PSU award \$594,941), 08/01/09-08/01/12, Collaborators: PSU (McDaniel, La Porta), UPenn (Blaze), Columbia (Schulzrinne).

PI, *Characterizing and Mitigating Wireless Systems Vulnerabilities*, Defense University Research Instrumentation Program (DURIP), Army Research Office (ARO), \$150,000, 05/22/09-02/28/11, Collaborators: PSU (La Porta, McDaniel).

co-PI, *Integrity Management for ICT Development*, Bell Labs Network Reliability and Security Office, Alcatel-Lucent, \$100,000, 11/30/08-11/30/09, Collaborators: PSU (La Porta, McDaniel).

PI, *Utility Grid Automation and Risk Management*, Lockheed Martin, \$400,000, 11/30/08-12/16/09.

PI, *EVEREST: Evaluation and Validation of Election-Related Equipment, Standards, and Testing*, The State of Ohio, \$716,336 (PSU award \$332,066), 10/01/07-01/07/08, Collaborators: PSU (McDaniel), UPenn (Blaze), UCSB (Kemmerer, Vigna), Berkeley (Hall, Quilter).

Co-PI, *Protecting Services for Emerging Wireless Telecommunications Infrastructure*, NSF (CNS), \$658,032, 09/01/07-08/31/11, Collaborators: PSU (La Porta, Jaeger, McDaniel).

Co-PI, *Security for Internet/IMS Convergence*, Cisco, \$100,000, 9/1/07-8/31/08, Collaborators: PSU (La Porta, McDaniel).

Co-PI, *System-Wide Information Flow Enforcement*, BAA 06-11-IFKA, "National Intelligence Community Enterprise Cyber Assurance Program", \$496,000, 2/1/07-8/1/08, Collaborators: PSU (Jaeger, McDaniel).

PI, *CAREER: Realizing Practical High Assurance through Security-Typed Information Flow Systems*, NSF (CNS), \$400,000, 1/2/07-1/1/12.

Co-PI, *CT-IS: Shamon: Systems Approaches for Constructing Distributed Trust*, NSF (CNS), \$400,000, 9/1/06-8/31/10, Collaborators: PSU (Jaeger, McDaniel).

Co-PI, *Center of Excellence*, Ben Franklin Technology Partners, \$75,000, 01/01/07-07/01/07, Collaborators: PSU (Cao, Jaeger, La Porta, McDaniel, Smith).

Co-PI, *Exploiting Asymmetry in Performance and Security Requirements for I/O in High-end Computing*, NSF (CFF), \$699,690, 9/1/06-8/31/10, Collaborators: PSU (McDaniel, Sivasubramaniam).

PI, *Automated Configuration with the PRESTO Network Management Platform*, AT&T, \$100,000, 6/1/06-5/31/07.

PI, *Testbed for Network-Scale Countermeasure Evaluation*, Cisco, \$45,938, 9/1/05-8/31/06.

PI, *Collaborative Research: CT-T: Flexible, Decentralized Information-flow Control for Dynamic Environments*, NSF (CFF), \$1,057,427 (PSU award \$234,585), 8/1/05-7/31/08, Collaborators: PSU (McDaniel), UPenn (Zdancewic), Maryland (Hicks), GMU (Winsborough).

PI, *Extending Developer Tools for Security-typed Languages*, Software Engineering Research Center, Sponsor: Motorola, \$23,200, 7/1/05-6/30/06.

PI, *Student Travel Support for ACM SIGCOMM 2005 Conference*, NSF, \$19,620, 4/1/05-3/31/06.

Co-PI, *NSF CyberTrust: Collaborative Research: Testing and Benchmarking Methodologies for Future Network Security Mechanisms (EMIST)*, NSF/DHS, \$5,344,459 (PSU award \$2,533,447), 8/1/04-8/31/06, Collaborators: PSU (Kesidis, Miller, Liu), Purdue (Fahmy, Rosenberg, Spafford, Shroff, Brodley), UCDavis (Wu, Levitt, Bishop, Rowe), ICSI/Berkeley (Paxson, Floyd, Weaver).

PROFESSIONAL ACTIVITIES

Editorial Positions, Panels, and Boards

IEEE Technical Committee on Security and Privacy

- *Chair*–January 2014–January 2016
- *Vice Chair*–January 2012–December 2014

ACM Transactions on Internet Technology (TOIT)

- *Editor in Chief*–September 2007–December 2012
- *Associate Editor*–April 2004–August 2007

IEEE Security and Privacy Magazine

- *Area Editor, Secure Systems*–January 2009–2015

IEEE Transactions on Computers (TC)

- *Associate Editor*–August 2008–2014

ACM Transactions on Information and System Security (TISSEC)

- *Associate Editor*–May 2007–May 2012

IEEE Transactions on Software Engineering (TSE)

- *Associate Editor*–January 2007–April 2012
- *Guest Editor, Special Issue on Topics in Security*–Fall 2006–April 2012

IEEE Transactions on Parallel and Distributed Systems (TPDS)

- *Guest Editor, Special Issue on Trust, Security and Privacy in Parallel and Distributed Systems*–Fall 2012

Elsevier Journal of Computer Networks

- *Guest Editor, Special Issue on Web Security*–Fall 2003–Spring 2005

Encyclopedia of Cryptography and Security

- *Editorial Board Member*–Fall 2002–Spring 2005

Journal of Defense Modeling and Simulation

- *Guest Editor, Special Issue on Cyber Risk and Vulnerability Estimation*–Winter 2018–

Other Professional Activities

IEEE Workshop on the Internet of Safe Things (SafeThings)

- *Steering Committee*–2022–present

Helmholtz Center for Information Security (CISPA), Scientific Advisory Board

- *Member*–2019–present

Ohio University College of Engineering, Board of Vistors

- *Member*–2018–present

Penn State CISO Advisory Board

- *Member*–2016–2022

Member, Technical Guideline Development Committee, U.S. Election Assistance Commission

- *Member*–2010–2011

Natural Sciences and Engineering Research Council of Canada, Internetworked Systems Security Network

- *Scientific Advisory Board*–2008–2013

Technology for Cyber Physical System Security Forum, Cyber Security Research and Development, (Senators Joseph I. Lieberman and Susan Collins, Chairs)

- *Speaker and Participant*–September 2008

ACM Student Organization Advisor

- *Penn State Computer Science and Engineering Department*–2006–2012

The Technology Collaborative

- *Penn State Representative (Pennsylvania economic development consortium)*–2007–2008

President's National Security Telecommunications Advisory Panel

- *Member, Next Generation Networks Task Force*–2005-2006

Abusable Technologies Awareness Center (ATAC)

- *Panelist*–October 2003-2010

AT&T IP Services Security Council

- *Member*–June 2003-August 2004

AT&T Internet Intellectual Property Review Team

- *Member*–September 2001-May 2002

ACM SIGCOMM Student Travel Grant Committee

- *Member*–August 2005

National Science Foundation, Grant Review Panel

- *Member*–2003-2004, 2006-2007, 2009-2021

Department of Energy SciDAC Review Panel

- *Member*–2001

Conference and Workshop Participation

IEEE Symposium on Security and Privacy

- *Technical Program Co-Chair*–2007, 2008
- *Program Committee*–2011, 2012, 2013, 2022, 2023, 2024

USENIX Security Symposium

- *Program Chair*–2005
- *Invited Talks Chair*–2006, 2009
- *Program Committee*–2001, 2003, 2004, 2007, 2014, 2018, 2019, 2020, 2021, 2022, 2023, 2024

ACM Conference on Computer and Communications Security (CCS)

- *Program Committee*–2006, 2008, 2009, 2010, 2012, 2018, 2019, 2020, 2021, 2022, 2023
- *Industry and Government Track Chair*–2004, 2007
- *Program Committee-Industry and Government Track*–2003, 2005, 2006
- *Test of Time Committee*–2019, 2020

IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)

- *Founding Program Co-Chair*–2023

IEEE European Symposium on Security and Privacy

- *Steering Committee*–2015-present
- *Program Committee*–2016, 2017

International Conference on Privacy, Security and Trust (PST)

- *Steering Committee*–2019-

ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)

- *Program Committee*–2017

Network and Distributed System Security Symposium (NDSS)

- *Program Committee*–2009, 2012, 2013, 2017

Annual Computer Security Applications Conference (ACSAC)

- *Program Committee*–2004, 2005, 2006, 2007, 2010, 2011, 2019
- *Test of Time Committee*–2019

Financial Cryptography

- *General Chair*–2006
- *Program Committee*–2007, 2008, 2012

Computer Security Foundations Symposium (CSF)

- *Program Committee*–2011, 2021

European Symposium on Research in Computer Security (ESORICS)

-
- *Program Committee*–2004, 2005, 2021
- ACM Symposium of SDx Research 2021 (SOSR)**
- *Program Committee*–2021
- International Symposium on Engineering Secure Software and Systems (ICISSP)**
- *Program Committee*–2015
- IEEE Conference on Communications and Network Security (CNS)**
- *Program Committee*–2015, 2017
- ACM Annual International Conference on Mobile Computing and Networking (MobiCom)**
- *Program Committee*–2010, 2011, 2012
 - *Program Committee, Distinguished Member*–2021
- ACM Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)**
- *Program Committee*–2012
- ACM Symposium on Access Control Models and Technologies (SACMAT)**
- *Program Committee*–2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2011
- ACM Conference on ASIA Computer and Communications Security (ASIA CCS)**
- *Program Committee*–2008
- ACM Conference on Electronic Commerce (ACM EC)**
- *Program Committee*–2005
- EAI International Conf. on Security and Privacy in Communication Networks (SecureComm)**
- *Program Committee*–2020
- International Conference on Applied Cryptography and Network Security (ACNS)**
- *Program Committee*–2006
- ACM Annual Digital Forensics Conference**
- *Program Committee*–2012
- IEEE Workshop on the Internet of Safe Things**
- *Program Committee*–2019, 2022
- ACM Workshop on Moving Target Defense (MTD)**
- *Program Committee*–2015, 2016
- IEEE ICNP Workshop on Secure Network Protocols (NPSec)**
- *Program Committee*–2005, 2006
- Conference on Decision and Game Theory for Security (GameSec)**
- *Program Committee*–2012, 2018
- ACM Symposium on Applied Computing (SAC)**
- *Program Committee, Information Security Research and Applications* –2010
- USENIX Annual Technical Conference**
- *Program Committee*–2002, 2003
- World Wide Web Conference (WWW)**
- *Security and Privacy Track Vice-Chair*–2005
 - *Security and Privacy Track Deputy Vice-Chair*–2004
 - *Program Committee*–2003, 2007, 2010, 2011
- Intl. Conference on Emerging Trends in Information and Communication Security (ETRICS)**
- *Program Committee*–2006
- International Conference On Distributed Computing Systems (ICDCS)**
- *Program Committee*–2011
- IEEE INFOCOM**
- *Program Committee*–2007
- IEEE GLOBECOM**
- *Program Committee*–2010
- Military Communications Conference (MILCOM)**

-
- *Program Committee*–2015, 2016, 2017, 2018, 2019, 2021, 2022
- The Five Nines Workshop on Designing and Managing High Availability Internet Services (INM)**
- *Program Committee*–2007
- International Conference on Information Systems Security (ICISS)**
- *Steering Committee*–2007
 - *Program Co-Chair*–2007
 - *Program Committee*–2005, 2006, 2009, 2011
- International Conference on Parallel Processing**
- *Program Committee-Network Security*–2003
- USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)**
- *Program Committee*–2010
- ACM Workshop on Networking, Systems, Applications on Mobile Handhelds (MobiHand)**
- *Program Committee*–2009
- ACM Workshop on Cloud Computing Security**
- *Program Committee*–2009, 2010
- ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM)**
- *Program Committee*–2011, 2012, 2013
- International Workshop on Security in Software Engineering**
- *Founding General Co-Chair*–2007
- USENIX Workshop On Offensive Technology (WOOT)**
- *Program Committee*–2007
- ACM Storage Security and Survivability Workshop**
- *Program Committee*–2006
- ACM SIGCOMM Workshop on Internet Network Management**
- *Program Committee*–2006, 2007
- Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSec)**
- *Program Committee*–2006, 2007, 2008
- Workshop on Workshop on Telecommunications Infrastructure Protection and Security (TIPS)**
- *General Chair*–2009
- USENIX Workshop on Hot Topics in Security (HotSec)**
- *Program Chair*–2011
 - *Program Committee*–2007, 2008, 2009, 2010, 2012
- ACM Workshop TPC on Security and Privacy in Smartphones and Mobile Devices**
- *Program Committee*–2011
- International Workshop on Security (IWSEC)**
- *Program Committee*–2006
- International Workshop on Systems and Network Security (SNS)**
- *Program Committee*–2005, 2006

COLLABORATORS (Last 48 Months)

Bolor-Erdene Zolbayar , Ahmed Abdou, Stefan Achleitner, Hidayet Aksu, Leonardo Babun, Paul Barford, Novella Bartolini, Yohan Beugin, Hoak, Blaine, Christopher Burchard, Quinn Burke, Z. Berkay Celik, Kamalika Chaudhuri, Adrien Cosson, Kyle Denney, Genannt Dohmann, Kyle Domico, Kyle D Domico, Matthew Durbin, Pauley, Eric, David Evans, Jean-Charles Noirot Ferrand, Felix Freiling, Anshul Gandhi, Rahul George, Kanad Ghose, Kartik Gopalan, Ryan Guide, Amy Hasan, Ting He, Blaine Hoak, Thorsten Holz, Syed Rafiul Hussain, Syed Rafiul Hussain, Alban Héon, Trent Jaeger, Somesh Jha, Rachel King, Engin Kirda, Farinaz Koushanfar, Srikanth V. Krishnamurthy, Daniel E. Krych, Domico, Kyle, Thomas La Porta, Dongyoon Lee, Kunyang Li, Azaree Lintereur, David Liu, David Yu Liu, Zhenhua Liu, Andrea Matwyshyn, Patrick McDaniel, Paul Barford Patrick McDaniel, Patrick Mcdaniel, Fidan Mehmeti, Shuai Mu, Namitha Nambiar, Michael Norris, Kyle Ostrowski, Nicolas Papernot, McDaniel, Patrick, Eric Pauley, Jonathan Petit, Jingyuan Qi, Burke, Quinn, Konrad Rieck,

Sheatsley, Ryan, Ahmad-Reza Sadeghi, Alejandro Andrade Salazar, Ryan Sheatsley, Tyler Shipp, Amit Kumar Sikder, Anand Sivasubramaniam, Wenjia Song, Indra Spiecker, Michael Swift, Gang Tan, Sanchal Thakkar, Selcuk Uluagac, Prasanna Venkatesh, Gunjan Verma, Bimal Viswanath, Michael J. Weisman, Ya Xiao, Tian Xie, Danfeng Yao, Beugin, Yohan, Mingi Yu, Mingli Yu, Erez Zadok, Danfeng Zhang, Shulin Zhao, Sencun Zhu, Shitong Zhu, Bolor-Erdene Zolbayarn

CASES (Expert Witness)

DivX, LLC v. Hulu, LLC and Netflix, Inc., Expert for Expert for the Defence, Central District of California, Case no. Case No. 2:19-cv-01602.

Inter Partes Review, Expert for Expert for Apple Inc., United States Patent and Trademark Office, Case no. Patent Nos. 9,947,000, 9,928,495 and 10,163,103.

Trusted Knight Corporation v. International Business Machines Corporation, Expert for Expert for IBM Corporation, Northern District of California, Case no. Case No. 3:19-cv-01206-EMC.

Inter Partes Review, Expert for Expert for IBM Corporation, United States Patent and Trademark Office, Case no. Patent No. 9,503,473.

Finjan Inc., vs. Rapid7 Inc and Rapid7 LLC, Expert for Expert for Rapid7, District of Delaware, Case no. C.A. No. 1:18-CV-01519-MN.

Finjan Inc., vs. Cisco Systems, Inc., Expert for Expert for CISCO, Northern District of California, San Jose Division, Case no. Case No. 5:17-cv-00072-BLF-SVK.

Inter Partes Review, Expert for Expert for BlackBerry Limited, United States Patent and Trademark Office, Case no. Patent Nos. 8,825,084 and 8,326,327.

BlackBerry Limited v. Snap. Inc., Expert for Expert for BlackBerry Limited, Central District of California, Case no. Case No. 2: 18-cv-02693.

Rimini Street, Inc. v. Oracle International Corporation, Expert for Expert for Oracle, United States District Court, District of Nevada, Case no. Case No. 2:14-cv-01699.

Inter Partes Review, Expert for Expert for Google Inc., United States Patent and Trademark Office, Case no. Patent No. 9,444,812.

Good Technology Corporation et al., v. AirWatch, LLC, Expert for Expert for Airwatch LLC, United States District Court for the Northern District of California, Case no. 5-12-cv-05827.

Inter Partes Review, Expert for Duo Security Inc., United States Patent and Trademark Office, Case no. IPR2017-01041.

Inter Partes Review, Expert for Duo Security Inc., United States Patent and Trademark Office, Case no. IPR2017-01064.

Frederick Whalen, et al., v. SEI/AARON'S, INC., Expert for defense, United States District Court for the Northern District of Georgia, Case no. 1:2014-cv-01209.

Certain Portable Electronic Communications Devices, Including Mobile Phones and Components Thereof, Expert for plaintiff, International Trade Commission.

Secure Access, LLC v. Bank of America Corp., et al., Expert for defense, United States District Court for the Eastern District of Texas, Tyler Division, Case no. 6-10-cv-00670.

Intellectual Ventures I LLC, v. Check Point et al., Expert for plaintiff, United States District Court for the District of Delaware, Case no. 1-10-cv-01067.

NetMonitor LLC, v. Compuware et al., Expert for plaintiff, United States District Court for the District of Delaware, Case no. 1-10-cv-01061.

Amdocs Ltd. v. Openet Telecom Ltd., Expert for defense, United States District Court for the Eastern District of Virginia, Alexandria Division, Case no. 1-10-cv-00910.

PSI Systems Inc. v. Stamps.com, Expert for defense, United States District Court for the Central District of California, Case no. 2-08-cv-05233.

Stamps.com v. Endicia, Expert for plaintiff, United States District Court for the Central District of California, Case no. 2-06-cv-07499.

Kara Technologies v. Stamps.com, Expert for defense, United States District Court for the Central District of California, Case no. 2-05-cv-01890.

Coinstar Inc. v. Coinxchange, Expert for defense, United States District Court for the Eastern District of Virginia, Richmond Division, Case no. 3-06-cv-00299.

VCode Holdings Inc. v. Stamps.com, Expert for defense, United States District Court for the Central District of California.

Inter Partes Review, Expert for Google Inc., United States Patent and Trademark Office, Case no. (Patent No. 8,489,868).

Inter Partes Review, Expert for Cisco Systems, Inc., , Case no. (Patent No. 7,536,598).

Vir2us v. Cisco Systems, INC and Sourcefire, LLC, Expert for defense, United States District Court for the Eastern District of Virginia, Case no. 51:16cv1095.

Zito Vault, LLC V. International Business Machines Corporation, and Softlayer Technologies, Inc., Expert for defense, , Case no. 3:16-CV-962-M.

INDUSTRIAL EXPERIENCE

Software Developer	1994-1995
Applied Innovation, Inc., Columbus, OH	
Project Manager	1993-1994
Primary Access Corporation, San Diego, CA	
Software Developer	1991-1993
Primary Access Corporation, San Diego, CA	
Software Developer	1989
Integrated Technologies, Inc., Muncie, IN	

PUBLICATIONS

Books and Book Chapters

Patrick Traynor and Patrick McDaniel and Thomas La Porta, *Security for Telecommunications Networks*, Springer, Advances in Information Security, July, 2008, ISBN: 978-0-387-72441-6.

Bolor-Erdene Zolbayar and Ryan Sheatsley and Patrick McDaniel, *Evading Machine Learning based Network Intrusion Detection Systems with GANs*, Game Theory and Machine Learning for Cyber Security, John Wiley & Sons, 2021. Eds. Charles A Kamhoua and Christopher D. Kiekintveld and Fei Fang and Quanyan Zhu. Hoboken, New Jersey.

Kevin Butler and William Enck and Patrick Traynor and Jennifer Plasterr and Patrick McDaniel, *Privacy Preserving Web-Based Email*, Algorithms, Architectures and Information Systems Security, Statistical Science and Interdisciplinary Research, World Scientific Computing, 349-371, November, 2008. Eds. Bhargab Bhattacharya, Susmita Sur-Kolay, Subhas Nandy and Aditya Bagchi.

Patrick McDaniel, *Authentication*, Handbook of Computer Networks, Volume II, Chapter 171, John Wiley and Sons, May, 2007. Eds. Hossein Bidgoli.

Patrick McDaniel, *Computer and Network Authentication*, Handbook of Information Security, John Wiley and Sons, September, 2004. Eds. Hossein Bidgoli.

Patrick McDaniel, *IPsec*, Encyclopedia of Information Security, Kluwer, 2003. Eds. Hossein Bidgoli.

Patrick McDaniel, *Policy*, Encyclopedia of Information Security, Kluwer, 2003. Eds. Hossein Bidgoli.

Patrick McDaniel, *Authentication*, The Internet Encyclopedia, John Wiley and Sons, 2002.

Journal Articles

Mingi Yu and Quinn Burke and Thomas La Porta and Patrick McDaniel, *Stealthy Misreporting Attacks Against Load Balancing*, IEEE/ACM Transactions on Networking (TON), 2024.

Anshul Gandhi and Dongyoon Lee and Zhenhua Liu and Shuai Mu and Erez Zadok and Kanad Ghose and Kartik Gopalan and David Yu Liu and Syed Rafiul Hussain and Patrick McDaniel, *Metrics for Sustainability in Data Centers*, SIGENERGY Energy Inform. Rev., New York, NY, USA, ACM, 3, 3, 40–46, October, 2023.

Quinn Burke and Patrick McDaniel and Thomas La Porta and Mingli Yu and Ting He, *Misreporting Attacks Against Load Balancers in Software-Defined Networking*, Mobile Networks and Applications, Springer, 2023.

Tian Xie and Sanchal Thakkar and Ting He and Patrick McDaniel and Quinn Burke, *Joint Caching and Routing in Cache Networks with Arbitrary Topology*, IEEE Transactions on Parallel and Distributed Systems, May, 2023.

Ya Xiao and Wenjia Song and Jingyuan Qi and Bimal Viswanath and Patrick McDaniel and Danfeng Yao, *Specializing Neural Networks for Cryptographic Code Completion Applications*, IEEE Transactions on Software Engineering, IEEE, 2023.

Matthew Durbin and Ryan Sheatsley and Patrick McDaniel and Azaree Lintereur, *Experimental tests of Gamma-ray Localization Aided with Machine-learning (GLAM) capabilities*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 1038, 2022.

Amit Kumar Sikder and Leonardo Babun and Z. Berkay Celik and Hidayet Aksu and Patrick McDaniel and Engin Kirda and Selcuk Uluagac, *Who's Controlling My Device? Multi-User Multi-Device-Aware Access Control System for Shared Smart Home Environment*, ACM Transactions on Internet of Things, Association for Comput-

ing Machinery, May, 2022.

Michael Norris and Z. Berkay Celik and Prasanna Venkatesh and Shulin Zhao and Patrick McDaniel and Anand Sivasubramaniam and Gang Tan, *IoTRepair: Flexible Fault Handling in Diverse IoT Deployments*, ACM Transactions on Internet of Things, ACM, 2022.

Tian Xie and Namitha Nambiar and Ting He and Patrick McDaniel, *Attack Resilience of Cache Replacement Policies: A Study Based on TTL Approximation*, IEEE/ACM Transactions on Networking, 2022.

Quinn Burke and Fidan Mehmeti and Rahul George and Kyle Ostrowski and Trent Jaeger and Thomas La Porta and Patrick Mcdaniel, *Enforcing Multilevel Security Policies in Unstable Networks*, IEEE Transactions on Network and Service Management, IEEE, June, 2022.

Ryan Sheatsley and Nicolas Papernot and Michael J. Weisman and Gunjan Verma and Patrick McDaniel, *Adversarial Examples for Network Intrusion Detection Systems*, Journal of Computer Security (JCS), IOS Press, January, 2022.

Quinn Burke and Patrick McDaniel and Thomas La Porta and Mingli Yu and Ting He, *Misreporting Attacks Against Load Balancers in Software-Defined Networking*, Mobile Networks and Applications, Springer, 2021.

Daniel E. Krych and Patrick McDaniel, *Exposing Android Social Applications: Linking Data Leakage to Privacy Policies*, Journal of Cyber Security Technology, Taylor & Francis, 5, 3-4, 2021.

Alejandro Andrade Salazar and Ryan Sheatsley and Jonathan Petit and Patrick McDaniel, *Physics-based Misbehavior Detection System for V2X Communications*, SAE International Journal of Connected and Automated Vehicles, SAE International, June, 2021.

Mingli Yu and Tian Xie and Ting He and Patrick McDaniel and Quinn Burke, *Flow Table Security in SDN: Adversarial Reconnaissance and Intelligent Attacks*, IEEE/ACM Transactions on Networking, 29, 6, 1063-6692, December, 2021.

Leonardo Babun and Kyle Denney and Z. Berkay Celik and Patrick McDaniel and Selcuk Uluagac, *A Survey on IoT Platforms: Communication, Security, and Privacy Perspectives*, Computer Networks, 192, 19, June, 2021.

Ryan Sheatsley and Matthew Durbin and Azaree Lintereur and Patrick McDaniel, *Improving Radioactive Material Localization by Leveraging Cyber-Security Model Optimizations*, IEEE Sensors, 21, 8, April, 2021.

Stefan Achleitner and Quinn Burke and Patrick McDaniel and Trent Jaeger and Thomas La Porta and Srikanth V. Krishnamurthy, *MLSNet: A Policy Complying Multilevel Security Framework for Software Defined Networking*, IEEE Transactions on Network and Service Management, 18, 1, March, 2021.

Dan Boneh and Andrew J. Grotto and Patrick McDaniel and Nicolas Papernot, *Preparing for the Age of Deepfakes and Disinformation*, Stanford HAI Policy Brief, 2020.

Dan Boneh and Andrew J. Grotto and Patrick McDaniel and Nicolas Papernot, *How Relevant Is the Turing Test in the Age of Sophisbots?*, IEEE Security & Privacy Magazine, 17, 64-71, Nov/Dec, 2019.

Z. Berkay Celik and Patrick McDaniel and Thomas Bowen, *Malware Modeling and Experimentation through Parameterized Behavior*, Journal of Defense Modeling and Simulation (JDMS), 15, 1, 31-48, 2017.

Z. Berkay Celik and Earlenice Fernandes and Eric Pauley and Gang Tan and Patrick McDaniel, *Program Analysis of Commodity IoT Applications for Security and Privacy: Opportunities and Challenges*, ACM Computing Surveys (CSUR), ACM, 42, 4, 2019.

Z. Berkay Celik and Patrick McDaniel and Gang Tan and Leonardo Babun and Selcuk Uluagac, *Verifying IoT Safety and Security in Physical Spaces*, IEEE Security & Privacy Magazine, IEEE, 17, 5, 30-37, 2019.

Ahmed Atya and Zhiyun Qian and Srikanth V. Krishnamurthy and Thomas La Porta and Patrick McDaniel and Lisa Marvel, *Catch Me if You Can: Malicious Co-Residency on the Cloud*, IEEE/ACM Transactions on Networking, 27, 2, April, 2019.

Ian Goodfellow and Patrick McDaniel and Nicolas Papernot, *Making machine learning robust against adversarial inputs*, Communications of the ACM, ACM, 61, 7, 56-6, June/July, 2018.

Dave Tian and Kevin Butler and Joseph Choi and Patrick McDaniel and Padma Krishnaswamy, *Securing ARP/NDP From the Ground Up*, IEEE Transactions on Information Forensics and Security, 12, 9, 2131-2143, April, 2017.

Stefan Achleitner and Thomas La Porta and Patrick McDaniel and Shridatt Sugrim and Srikanth V. Krishnamurthy and Ritu Chada, *Deceiving Network Reconnaissance Using SDN-based Virtual Topologies*, IEEE Transactions on Network and Service Management, Special Issue on Advances in Management of Softwarized Networks, 14, 4, July, 2017.

Chaz Lever and Robert Walls and Yacin Nadjji and David Dagon and Patrick McDaniel and Manos Antonakakis, *Dawn of the Dead Domain: Measuring the Exploitation of Residual Trust in Domains*, IEEE Security & Privacy Magazine (Secure Systems issue column), April, 2017.

Patrick McDaniel and Nicolas Papernot and Z. Berkay Celik, *Machine Learning in Adversarial Settings*, IEEE Security & Privacy Magazine, 14, 3, May/June, 2016.

Patrick McDaniel and Ananthram Swami, *The Cyber Security Collaborative Research Alliance: Unifying Detection, Agility, and Risk in Mission-Oriented Cyber Decision Making*, CSIAC Journal, Army Research Laboratory (ARL) Cyber Science and Technology, 5, 1, January, 2017.

Damien Octeau and Daniel Luchaup and Somesh Jha and Patrick McDaniel, *Composite Constant Propagation and its Application to Android Program Analysis*, IEEE Transactions on Software Engineering, 42, 11, 999-1014, 2016.

Alexander Kott and Ananthram Swami and Patrick McDaniel, *Security Outlook: Six Cyber Game Changers for the Next 15 Years*, IEEE Computer, 47, 12, 104-106, 2014.

Zhenfu Cao and Keqiu Li and Xu Li and Patrick McDaniel and Radha Poovendran and Guojun Wang and Yang Xiang, *Guest Editors' Introduction: Special Issue on Trust, Security, and Privacy in Parallel and Distributed Systems*, IEEE Transactions on Parallel and Distributed Systems, 25, 2, 279-28, 2014.

Patrick McDaniel and Brian Rivera and Ananthram Swami, *Toward a Science of Secure Environments*, IEEE Security & Privacy Magazine, 12, 4, July/August, 2014.

William Enck and Peter Gilbert and Seungyeop Han and Vasant Tendulkar and Byung-Gon Chun and Landon Cox and Jaeyeon Jung and Patrick McDaniel and Anmol Sheth, *TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones*, ACM Transactions on Computer Systems (TOCS), 32, 2, June, 2014.

William Enck and Peter Gilbert and Byung-Gon Chun and Landon Cox and Jaeyeon Jung and Patrick McDaniel and Anmol Sheth, *TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones*, Communications of the ACM, 57, 3, March, 2014.

Machigar Ongtang and Stephen McLaughlin and William Enck and Patrick McDaniel, *Semantically Rich Application-Centric Security in Android*, Security and Communication Networks, 5, 6, 658-673, 2012.

Patrick McDaniel, *Bloatware Comes to the Smartphone*, IEEE Security & Privacy Magazine, 10, 4, July/August, 2011.

Thomas Moyer and Kevin Butler and Joshua Schiffman and Patrick McDaniel and Trent Jaeger, *Scalable Web Content Attestation*, IEEE Transactions on Computers, 61, 5, 686–699, April, 2011.

Patrick McDaniel, *Data Provenance and Security*, IEEE Security & Privacy Magazine, 9, 3, March/April, 2011.

Joshua Schiffman and Thomas Moyer and Trent Jaeger and Patrick McDaniel, *Network-based Root of Trust for Installation*, IEEE Security & Privacy Magazine, 40-48, Jan/Feb, 2011.

Kevin Butler and Stephen McLaughlin and Thomas Moyer and Patrick McDaniel, *New Security Architectures Based on Emerging Disk Functionality*, IEEE Security and Privacy Magazine, 8, 5, October, 2010.

Patrick McDaniel and William Enck, *Not So Great Expectations: Why Application Markets Haven't Failed Security*, IEEE Security & Privacy Magazine, 8, 5, 76–78, September/October, 2010.

Patrick Traynor and Chaitrali Amrutkar and Vikhyath Rao and Trent Jaeger and Patrick McDaniel and Thomas La Porta, *From Mobile Phones to Responsible Devices*, Journal of Security and Communication Networks (SCN), 4, 6, 719 – 726, June, 2011.

Patrick Traynor and Kevin Butler and William Enck and 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, March, 2010.

Matthew Pirretti and Patrick Traynor and Patrick McDaniel and Brent Waters, *Secure Attribute-Based Systems*, Journal of Computer Security (JCS), 18, 5, 799–837, 2010.

Boniface Hicks and Sandra Rueda and Luke St. Clair and Trent Jaeger and Patrick McDaniel, *A Logical Specification and Analysis for SELinux MLS Policy*, ACM Transactions on Information and System Security (TISSEC), 13, 26, 2010.

Kevin Butler and Toni Farley and Patrick McDaniel and J. Rexford, *A Survey of BGP Security Issues and Solutions*, Proceedings of the IEEE, 2010, 1, 100-122, January, 2010.

Kevin Butler and Sunam Ryu and 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, December, 2009.

Patrick McDaniel and Stephen McLaughlin, *Security and Privacy Challenges in the Smart Grid*, IEEE Security & Privacy Magazine (Secure Systems issue column), 7, 3, 75-77, May/June, 2009.

Patrick Traynor and William Enck and Patrick McDaniel and Thomas La Porta, *Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networks*, IEEE/ACM Transactions on Networking (TON), 17, 1, 40-53, February, 2009.

William Enck and Machigar Ongtang and Patrick McDaniel, *Understanding Android Security*, IEEE Security & Privacy Magazine, 7, 1, 50–57, January/February, 2009.

William Enck and Thomas Moyer and Patrick McDaniel and Shubho Sen and Panagiotis Sebos and Sylke Sporel and Albert Greenberg and Yu-Wei Sung and Sanjay Rao and William Aiello, *Configuration Management at Massive Scale: System Design and Experience*, IEEE Journal on Selected Areas in Communications (JSAC), 27, 3, 323-335, 2009.

Heesook Choi and William Enck and Jaesheung Shin and Patrick McDaniel and Thomas La Porta, *ASR: Anonymous and Secure Reporting of Traffic Forwarding Activity in Mobile Ad Hoc Networks*, Wireless Networks (WINET), ACM/Kluwer, 15, 4, 525-539, May, 2009.

Patrick Traynor and William Enck and Patrick McDaniel and Thomas La Porta, *Exploiting Open Functionality in SMS-Capable Cellular Networks*, Journal of Computer Security, 16, 6, 713-742, Febraury, 2009.

Patrick Traynor and Michael Chien and Scott Weaver and Boniface Hicks and Patrick McDaniel, *Non-Invasive Methods for Host Certification*, ACM Transactions on Information and System Security (TISSEC), 11, 3, 2008.

Patrick McDaniel and Bashar Nuseibeh, *Guest Editorial: Special Issue on Software Engineering for Secure Systems*, IEEE Transactions on Software Engineering, 34, 1, 3-4, 2008.

Wesam Lootah and William Enck and Patrick McDaniel, *TARP: Ticket-based Address Resolution Protocol*, Computer Networks, Elsevier, 51, 15, 4322-4337, October, 2007.

Patrick McDaniel and Avi Rubin, *Guest Editorial: Special Issue on Web Security*, Computer Networks, Elsevier, 22, 2, 2005.

Patrick McDaniel and Atul Prakash, *Enforcing Provisioning and Authorization Policy in the Antigone System*, Journal of Computer Security, 14, 6, 483-511, November, 2006.

Patrick McDaniel and Atul Prakash, *Methods and Limitations of Security Policy Reconciliation*, ACM Transactions on Information and System Security (TISSEC), Association for Computing Machinery, 9, 3, 259-291, August, 2006.

Patrick McDaniel and William Aiello and Kevin Butler and JohnIoannidis, *Origin Authentication in Interdomain Routing*, Journal of Communication Networks, Elsevier, 50, 16, 2953-2980, November, 2006.

Matthew Pirretti and Sencun Zhu and VijaykrishnanNarayanan and Patrick McDaniel and Mahmut Kandemir and Richard Brooks, *The Sleep Deprivation Attack in Sensor Networks: Analysis and Methods of Defense*, International Journal of Distributed Sensor Networks, 2, 3, 267-287, June, 2005.

Simon Byers and Lorrie Cranor and Eric Cronin and DaveKormann and Patrick McDaniel, *Analysis of Security Vulnerabilities in the Movie Production and Distribution Process*, Telecommunications Policy, 28, 8, 619-644, August, 2004.

Conference Publications

Yohan Beugin and Patrick McDaniel, *Interest-disclosing Mechanisms for Advertising are Privacy-Exposing (not Preserving)*, Proceedings on Privacy Enhancing Technologies (PETS), July, 2024.

Blaine Hoak and Patrick McDaniel, *Explorations in Texture Learning*, 12th International Conference on Learning Representations, Tiny Papers Track (ICLR), May, 2024.

Tian Xie and Sanchal Thakkar and Ting He and Novella Bartolini and Patrick McDaniel, *Host-based Flow Table Size Inference in Multi-hop SDN*, Proceedings of the IEEE GLOBECOM, IEEE, December, 2023. Kuala Lumpur, Malaysia.

Eric Pauley and Paul Barford and Patrick McDaniel, *The CVE Wayback Machine: Measuring Coordinated Disclosure from Exploits Against Two Years of Zero-Days*, Proceedings of the ACM 2023 Internet Measurement Conference (IMC), October, 2023. Montreal, Canada. (**runner up-best paper award**)

Mingli Yu and Quinn Burke and Thomas La Porta and Patrick McDaniel, *mMLSnet: Multilevel Security Network With Mobility*, Proceedings of the Military Communications Conference (MILCOM), IEEE, October, 2023. Boston, MA.

Ryan Guide and Eric Pauley and Yohan Beugin and Ryan Sheatsley and Patrick McDaniel, *Characterizing the Modification Space of Signature IDS Rules*, MILCOM 2023 - 2023 IEEE Military Communications Conference

(MILCOM), IEEE, October, 2023. Boston, MA.

Eric Pauley and Paul Barford Patrick McDaniel, *DScope: A Cloud-Native Internet Telescope*, 32nd USENIX Security Symposium (USENIX Security 23), USENIX Association, August, 2023. Anaheim, CA.

Ryan Sheatsley and Blaine Hoak and Eric Pauley and Patrick McDaniel, *The Space of Adversarial Strategies*, 32nd USENIX Security Symposium (USENIX Security 23), USENIX Association, August, 2023.

Eric Pauley and Gang Tan and Danfeng Zhang and Patrick McDaniel, *Performant Binary Fuzzing without Source Code using Static Instrumentation*, Conference on Communications and Network Security (CNS), IEEE, October, 2022.

Patrick McDaniel, *Keynote Address: Sustainability is a Security Problem (extended abstract)*, Proceedings of the ACM Conference on Computer and Communications Security (CCS), ACM, November, 2022.

Kyle Domico and Ryan Sheatsley and Yohan Beugin and Quinn Burke and Patrick McDaniel, *A Machine Learning and Computer Vision Approach to Geomagnetic Storm Forecasting*, Machine Learning in Heliophysics (ML-Helio), AGU, November, 2022.

Tian Xie and Sanchal Thakkar and Ting He and Patrick McDaniel and Quinn Burke, *Joint Caching and Routing in Cache Networks with Arbitrary Topology*, Proceedings of the International Conference on Distributed Computing Systems (ICDCS), July, 2022.

Yohan Beugin and Quinn Burke and Blaine Hoak and Ryan Sheatsley and Eric Pauley and Gang Tan and Syed Rafiul Hussain and Patrick McDaniel, *Building a Privacy-Preserving Smart Camera System*, Proceedings on Privacy Enhancing Technologies (PETS), July, 2022.

Eric Pauley and Ryan Sheatsley and Blaine Hoak and Quinn Burke and Yohan Beugin and Patrick McDaniel, *Measuring and Mitigating the Risk of IP Reuse on Public Clouds*, 2022 IEEE Symposium on Security and Privacy (IEEE S&P), IEEE, May, 2022. San Francisco, CA.

Ahmed Abdou and Ryan Sheatsley and Yohan Beugin and Tyler Shipp and Patrick McDaniel, *HoneyModels: Machine Learning Honey Pots*, Proceedings of the Military Communications Conference (MILCOM), IEEE, November, 2021.

Ryan Sheatsley and Blaine Hoak and Eric Pauley and Yohan Beugin and Michael J. Weisman and Patrick McDaniel, *On the Robustness of Domain Constraints*, Proceedings of the ACM Conference on Computer and Communications Security (CCS), ACM, November, 2021.

Tian Xie and Ting He and Patrick McDaniel and Namitha Nambiar, *Attack Resilience of Cache Replacement Policies*, IEEE International Conference on Computer Communications (INFOCOM), IEEE, May, 2021.

Adrien Cosson and Amit Kumar Sikder and Leonardo Babun and Z. Berkay Celik and Patrick McDaniel and Selcuk Uluagac, *Sentinel: A Robust Intrusion Detection System for IoT Networks Using Kernel-Level System Information*, In 6th ACM/IEEE Conference on Internet of Things Design and Implementation (IoTDI), April, 2021.

Matthew Durbin and Ryan Sheatsley and Patrick McDaniel and Azaree Lintereur, *A Multi-Step Machine Learning Approach to Directional Gamma Ray Detection*, 2020 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC), October, 2020.

Sayed M. Saghaian and Thomas La Porta and Simone Silvestri and Patrick McDaniel, *Improving Robustness of a Popular Probabilistic Clustering Algorithm Against Insider Attacks*, International Conference on Security and Privacy in Communication Networks (SecureComm 2020), EAI, October, 2020.

Quinn Burke and Patrick McDaniel and Thomas La Porta and Mingli Yu and Ting He, *Misreporting Attacks in Software-Defined Networking*, International Conference on Security and Privacy in Communication Networks

(SecureComm 2020), EAI, October, 2020.

Amit Kumar Sikder and Leonardo Babun and Z. Berkay Celik and Abbas Acar and Hidayet Aksu and Patrick McDaniel and Engin Kirda and Selcuk Uluagac, *KRATOS: Multi-User Multi-Device-Aware Access Control System for the Smart Home*, 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '20), ACM, July, 2020.

Mingli Yu and Ting He and Patrick McDaniel and Quinn Burke, *Flow Table Security in SDN: Adversarial Reconnaissance and Intelligent Attacks*, IEEE INFOCOM, IEEE Conference on Computer Communications, June, 2020. Beijing, China.

Michael Norris and Z. Berkay Celik and Prasanna Venkatesh and Shulin Zhao and Patrick McDaniel and Anand Sivasubramaniam and Gang Tan, *IoTRepair: Systematically addressing device faults in commodity IoT*, In 5th ACM/IEEE Conference on Internet of Things Design and Implementation (IoTDI), April, 2020.

Raquel Alvarez and Jake Levenson and Ryan Sheatsley and Patrick McDaniel, *Application Transiency: Towards a Fair Trade of Personal Information for Application Services*, Proceedings of the EAI Conference on Security and Privacy in Communication Networks (SecureComm), October, 2019. Orlando, FL.

Giuseppe Petracca and Yuqiong Sun and Ahmad-Atamli Reineh and Jens Grossklags and Patrick McDaniel and Trent Jaeger, *EnTrust: Regulating Sensor Access by Cooperating Programs via Delegation Graphs*, Proceedings of the 28th USENIX Security Symposium, August, 2019. Santa Clara, CA.

Matthew Durbin and Ryan Sheatsley and Christopher Balbier and Tristan Grieve and Patrick McDaniel and Azaree Lintereur, *Development of Machine Learning Algorithms for Directional Gamma Ray Detection*, Proceedings of the Institute of Nuclear Materials Management Annual Meeting (INMM), July, 2019. Palm Desert, CA. (**J. D. Williams student paper award, Nuclear Security and Physical Protection division**)

Z. Berkay Celik and Abbas Acar and Hidayet Aksu and Abbas Acar and Ryan Sheatsley and Selcuk Uluagac and Patrick McDaniel, *Curie: Policy-based Secure Data Exchange*, ACM Conference on Data and Applications Security (CODASPY), March, 2019. Dallas, TX.

Z. Berkay Celik and Gang Tan and Patrick McDaniel, *IoTGuard: Dynamic Enforcement of Security and Safety Policy in Commodity IoT*, Network and Distributed System Security Symposium (NDSS), February, 2019. San Diego, CA.

Dang Tu Nguyen and Chengyu Song and Zhiyun Qian and Srikanth V. Krishnamurthy and Edward J. M. Colbert and Patrick McDaniel, *IoTSan: Fortifying the Safety of IoT Systems*, Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '18), December, 2018.

Z. Berkay Celik and Leonardo Babun and Amit Kumar Sikder and Hidayet Aksu and Gang Tan and Patrick McDaniel and Selcuk Uluagac, *Sensitive Information Tracking in Commodity IoT*, Proceedings of the 27th USENIX Security Symposium, August, 2018. Baltimore, MD.

Rauf Izmailov and Shridatt Sugrim and Ritu Chadha and Patrick McDaniel and Ananthram Swami, *Enablers Of Adversarial Attacks in Machine Learning*, Proceedings of the Military Communications Conference (MILCOM), IEEE, October, 2018.

Sayed M. Saghaian and Thomas La Porta and Trent Jaeger and Z. Berkay Celik and Patrick McDaniel, *Mission-oriented Security Model, Incorporating Security Risk, Cost and Payout*, Proceedings of EAI SECURECOMM 2018, August, 2018. (**best paper award**)

Z. Berkay Celik and Patrick McDaniel and Gang Tan, *Soteria: Automated IoT Safety and Security Analysis*, USENIX Annual Technical Conference (USENIX ATC), July, 2018. Boston, MA.

Z. Berkay Celik and Patrick McDaniel and Rauf Izmailov and Nicolas Papernot and Ryan Sheatsley and Raquel Alvarez and Ananthram Swami, *Detection under Privileged Information*, ACM Asia Conference on Computer and Communications Security (ASIACCS), June, 2018.

Florian Tramer and Alexey Kurakin and Nicolas Papernot and Ian Goodfellow and Dan Boneh and Patrick McDaniel, *Ensemble Adversarial Training: Attacks and Defenses*, International Conference on Learning Representations (ICLR), 2018. Vancouver, Canada.

Nicolas Papernot and Patrick McDaniel and Arunesh Sinha and Michael Wellman, *SoK: Security and Privacy in Machine Learning*, Security and Privacy (EuroS&P), 2018 IEEE European Symposium on on Security and Privacy (EuroS&P), IEEE, April, 2018. London, UK.

Chun-Ming Lai and Xiaoyun Wang and Yunfeng Hong and Yu-Cheng Lin and Felix Wu and Patrick McDaniel and Hasan Cam, *Attacking Strategies and Temporal Analysis Involving Facebook Discussion Groups*, 13th IEEE International Conference on Network and Service Management, November, 2017. Tokyo, Japan.

Kathrin Grosse and Nicolas Papernot and Praveen Manoharan and Michael Backes and Patrick McDaniel, *Adversarial Examples for Malware Detection*, 22nd European Symposium on Research in Computer Security (ESORICS '17), September, 2017. Oslo, Norway.

Z. Berkay Celik and David Lopez-Paz and Patrick McDaniel, *Patient-Driven Privacy Control through Generalized Distillation*, Proceedings of the Privacy-Aware Computing (PAC), IEEE, 2017.

Abbas Acar and Z. Berkay Celik and Hidayet Aksu and Selcuk Uluagac and Patrick McDaniel, *Achieving Secure and Differentially Private Computations in Multiparty Settings*, Proceedings of the Privacy-Aware Computing (PAC), IEEE, 2017.

Vaibhav Rastogi and Drew Davidson and Lorenzo De Carli and Somesh Jha and Patrick McDaniel, *Cimplifier: Automatically Debloating Containers*, 11Th Joint Meeting of the European Software Engineering Conference and the Acm Sigsoft Symposium on the Foundations of Software Engineering, September, 2017. Paderborn, Germany.

Yunfeng Hong and Yongjian Hu and Chun-Ming Lai and Felix Wu and Iulian Neamtiu and Yu Paul and Patrick McDaniel and Hasan Cam and Gail-Joon Ahn, *Defining and Detecting Environment Discrimination in Android Apps*, The 13th EAI International Conference on Security and Privacy in Communication Networks (SecureComm 17), October, 2017.

Nicolas Papernot and Patrick McDaniel and Ian Goodfellow and Somesh Jha and Z. Berkay Celik and Ananthram Swami, *Practical Black-Box Attacks against Machine Learning*, ACM Asia Conference on Computer and Communications Security (ASIACCS) 2017, April, 2017.

Stefan Achleitner and Thomas La Porta and Trent Jaeger and Patrick McDaniel, *Adversarial Network Forensics in Software Defined Networking*, ACM Symposium on SDN Research (SOSR), ACM, April, 2017. (**best student paper award**)

Stefan Achleitner and Thomas La Porta and Patrick McDaniel and Srikanth V. Krishnamurthy and Alexander Poylisher and Constantin Serban, *Stealth Migration: Hiding Virtual Machines on the Network*, IEEE International Conference on Computer Communications (INFOCOM), IEEE, 2017.

Ahmed Atya and Zhiyun Qian and Srikanth V. Krishnamurthy and Thomas La Porta and Patrick McDaniel and Lisa Marvel, *Malicious Co-Residency on the Cloud: Attacks and Defense*, IEEE International Conference on Computer Communications (INFOCOM), IEEE, 2017.

Nathaniel Lageman and Eric Kilmer and Robert Walls and Patrick McDaniel, *BinDNN: Resilient Function Matching Using Deep Learning*, 2016 International Conference on Security and Privacy in Communication Networks (SECURECOMM), October, 2016.

Z. Berkay Celik and Nan Hu and Yun Li and Nicolas Papernot and Patrick McDaniel and Jeff Rowe and Robert Walls and Karl Levitt and Novella Bartolini and Thomas La Porta and Ritu Chadha, *Mapping Sample Scenarios to Operational Models*, Proceedings of the Military Communications Conference (MILCOM), IEEE, 2016.

Nicolas Papernot and Patrick McDaniel and Ananthram Swami and Richard Harang, *Crafting Adversarial Input Sequences for Recurrent Neural Networks*, Proceedings of the Military Communications Conference (MILCOM), IEEE, 2016.

Michael Backes and Sven Bugiel and Erik Derr and Patrick McDaniel and Damien Oteau and Sebastian Weisgerber, *On Demystifying the Android Application Framework: Re-Visiting Android Permission Specification Analysis*, Proceedings of the 25th USENIX Security Symposium, August, 2016.

Devin Pohly and Patrick McDaniel, *Modeling Privacy and Tradeoffs in Multichannel Secret Sharing Protocols*, 46th IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), June, 2016.

Chaz Lever and Robert Walls and Yacin Nadji and David Dagon and Patrick McDaniel and Manos Antonakakis, *Domain-Z: 28 Registrations Later*, Proceedings of the 37th IEEE Symposium on Security and Privacy, May, 2016. San Francisco, CA.

Yasemin Acar and Michael Backes and Sven Bugiel and Sascha Fahl and Patrick McDaniel and Matthew Smith, *SoK: Lessons Learned From Android Security Research For Appified Software Platforms*, Proceedings of the 37th IEEE Symposium on Security and Privacy, May, 2016. San Francisco, CA.

Nicolas Papernot and Patrick McDaniel and Xi Wu and Somesh Jha and Ananthram Swami, *Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks*, Proceedings of the 37th IEEE Symposium on Security and Privacy, May, 2016. San Francisco, CA.

Charles Huber and Scott Brown and Patrick McDaniel and Lisa Marvel, *Cyber Fighter Associate: A Decision Support System for Cyber Agility*, Proceedings of the 50th Annual Conference on Information Sciences and Systems (CISS), March, 2016. Princeton, NJ.

Nicolas Papernot and Patrick McDaniel and Somesh Jha and Matthew Fredrikson and Z. Berkay Celik and Ananthram Swami, *The Limitations of Deep Learning in Adversarial Settings*, Proceedings of the 1st IEEE European Symposium on Security and Privacy, IEEE, 2016. Saarbrücken, Germany.

Damien Oteau and Somesh Jha and Matthew Dering and Patrick McDaniel and Alexandre Bartel and Li Li and Jacques Klein and Yves Le Traon, *Combining Static Analysis with Probabilistic Models to Enable Market-Scale Android Inter-Component Analysis*, Proceedings of the 43rd ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL), January, 2016. St. Petersburg, Florida, USA.

Devin Pohly and Patrick McDaniel, *MICSS: A Realistic Multichannel Secrecy Protocol*, IEEE Global Communications Conference (GLOBECOM), December, 2015. San Diego, CA.

Robert Walls and Eric Kilmer and Nathaniel Lageman and Patrick McDaniel, *Measuring the Impact and Perception of Acceptable Advertisements*, Proceedings of the ACM 2015 Internet Measurement Conference (IMC), October, 2015. Tokyo, Japan.

Nicolas Papernot and Patrick McDaniel and Robert Walls, *Enforcing Agile Access Control Policies in Relational Databases using Views*, Proceedings of the Military Communications Conference (MILCOM), October, 2015. Tampa, FL.

Alessandro Oltramari and Lorrie Cranor and Robert Walls and Patrick McDaniel, *Computational Ontology of Network Operations*, Proceedings of the Military Communications Conference (MILCOM), October, 2015. Tampa, FL.

Z. Berkay Celik and Robert Walls and Patrick McDaniel and Ananthram Swami, *Malware Traffic Detection using Tamper Resistant Features*, Proceedings of the Military Communications Conference (MILCOM), October, 2015. Tampa, FL.

Devin Pohly and Charles Sestito and Patrick McDaniel, *Adaptive Protocol Switching Using Dynamically Insertable Bumps in the Stack*, Proceedings of the Military Communications Conference (MILCOM), October, 2015. Tampa, FL.

Azeem Aqil and Ahmed Atya and Trent Jaeger and Srikanth V. Krishnamurthy and Karl Levitt and Patrick McDaniel and Jeff Rowe and Ananthram Swami, *Detection of Stealthy TCP-based DoS Attacks*, Proceedings of the Military Communications Conference (MILCOM), October, 2015. Tampa, FL.

Daniel E. Krych and Stephen Lange-Maney and Patrick McDaniel and William Glodek, *Investigating Weaknesses in Android Certificate Security*, SPIE 9478, Modeling and Simulation for Defense Systems and Applications X, May, 2015.

Damien Oceau and Daniel Luchaup and Matthew Dering and Somesh Jha and Patrick McDaniel, *Composite Constant Propagation: Application to Android Inter-Component Communication Analysis*, Proceedings of the 37th International Conference on Software Engineering (ICSE), May, 2015. Florence, Italy.

Li Li and Alexandre Bartel and Tegawende Bissyande and Jacques Klein and Yves Le Traon and Steven Arzt and Siegfried Rasthofer and Eric Bodden and Damien Oceau and Patrick McDaniel, *IccTA: Detecting Inter-Component Privacy Leaks in Android Apps*, Proceedings of the 37th International Conference on Software Engineering (ICSE), May, 2015. Florence, Italy.

Jing Tian and Kevin Butler and Patrick McDaniel and Padma Krishnaswamy, *Securing ARP From the Ground Up*, CODASPY '15: Proceedings of the 5th ACM Conference on Data Application and Security and Privacy, March, 2015. San Antonio, TX, USA.

Alessandro Oltramari and Lorrie Cranor and Robert Walls and Patrick McDaniel, *Building an Ontology of Cyber Security*, Proc. Intl. Conference on Semantic Technologies for Intelligence, Defense, and Security (STIDS), November, 2014.

Matthew Dering and Patrick McDaniel, *Android Market Reconstruction and Analysis*, Proceedings of the Military Communications Conference (MILCOM), October, 2014. Baltimore, MD.

Wenhui Hu and Damien Oceau and Patrick McDaniel and Peng Liu, *Duet: Library Integrity Verification for Android Applications*, Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec), July, 2014. Oxford, United Kingdom.

Steven Arzt and Siegfried Rasthofer and Christian Fritz and Eric Bodden and Alexandre Bartel and Jacques Klein and Yves Le Traon and Damien Oceau and Patrick McDaniel, *FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps*, Proceedings of the 35th Conference on Programming Language Design and Implementation (PLDI), June, 2014. Edinburgh, UK.

Phillip Koshy and Diana Koshy and Patrick McDaniel, *An Analysis of Anonymity in Bitcoin Using P2P Network Traffic*, Proceedings of Financial Cryptography 2014, International Financial Cryptography Association (IFCA), February, 2014. Christ Church, Barbado.

Stephen McLaughlin and Devin Pohly and Patrick McDaniel and Saman Zonouz, *A Trusted Safety Verifier for Process Controller Code*, Proc. ISOC Network and Distributed Systems Security Symposium (NDSS), February, 2014. San Diego, CA.

Damien Oceau and Patrick McDaniel and Somesh Jha and Alexandre Bartel and Eric Bodden and Jacques Klein and Yves Le Traon, *Effective Inter-Component Communication Mapping in Android with Epiccc: An Essential Step Towards Holistic Security Analysis*, Proceedings of the 22th USENIX Security Symposium, August, 2013.

Washington, DC.

Devin J. Pohly and Stephen McLaughlin and Patrick McDaniel and Kevin Butler, *Hi-Fi: Collecting High-Fidelity Whole-System Provenance*, Proceedings of the 28th Annual Computer Security Applications Conference (ACSAC), December, 2012. Orlando, Florida.

Stephen McLaughlin and Patrick McDaniel, *SABOT: Specification-based Payload Generation for Programmable Logic Controllers*, 19th ACM Conference on Computer and Communications Security (CCS), October, 2012.

Weining Yang and Ninghui Li and Yuan Qi and Wahbeh Qardaji and Stephen McLaughlin and Patrick McDaniel, *Minimizing Private Data Disclosures in the Smart Grid*, 19th ACM Conference on Computer and Communications Security (CCS), October, 2012.

Eun Kyoung Kim and Patrick McDaniel and Thomas La Porta, *A Detection Mechanism for SMS Flooding Attacks in Cellular Networks*, Proceedings of the 8th International Conference on Security and Privacy in Communication Networks (SECURECOMM 2012), September, 2012. Padua, Italy.

Damien Oceau and Somesh Jha and Patrick McDaniel, *Retargeting Android Applications to Java Bytecode*, 20th International Symposium on the Foundations of Software Engineering (FSE), November, 2012. Research Triangle Park, NC. (**best artifact award**)

Thomas Moyer and Trent Jaeger and Patrick McDaniel, *Scalable Integrity-Guaranteed AJAX*, Proceedings of the 14th Asia-Pacific Web Conference (APWeb), April, 2012. Kunming, China. (*Invited Paper*)

Patrick McDaniel and Stephen McLaughlin, *Structured Security Testing in the Smartgrid*, Proceedings of 5th International Symposium on Communications, Control, and Signal Processing, May, 2012. Rome, Italy. (*Invited Paper*)

Stephen McLaughlin and Patrick McDaniel and William Aiello, *Protecting Consumer Privacy from Electric Load Monitoring*, The 18th ACM Conference on Computer and Communications Security (CCS), October, 2011. Chicago, IL.

William Enck and Damien Oceau and Patrick McDaniel and Swarat Chaudhuri, *A Study of Android Application Security*, Proceedings of the 20th USENIX Security Symposium, August, 2011. San Francisco, CA.

Kevin Butler and Stephen McLaughlin and Patrick McDaniel, *Kells: A Protection Framework for Portable Data*, Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC), December, 2010. Austin, TX.

Stephen McLaughlin and Dmitry Podkuiko and Adam Delozier and Sergei Miadzvezhanka and Patrick McDaniel, *Multi-vendor Penetration Testing in the Advanced Metering Infrastructure*, Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC), December, 2010. Austin, TX.

Machigar Ongtang and Kevin Butler and Patrick McDaniel, *Porscha: Policy Oriented Secure Content Handling in Android*, Proceedings of the 26th Annual Computer Security Applications Conference (ACSAC), December, 2010. Austin, TX.

Patrick Traynor and Joshua Schiffman and Thomas La Porta and Patrick McDaniel and Abhrajit Ghosh and Farooq Anjum, *Constructing Secure Localization Systems with Adjustable Granularity*, IEEE Global Communications Conference (GLOBECOM), December, 2010. Miami, FL.

William Enck and Peter Gilbert and Byung-Gon Chun and Landon Cox and Jaeyeon Jung and Patrick McDaniel and Anmol Sheth, *TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones*, Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI), October, 2010. Vancouver, BC.

Toby Ehrenkranz and Jun Li and Patrick McDaniel, *Realizing A Source Authentic Internet*, Proceedings of the 6th International ICST Conference on Security and Privacy in Communications Networks (Securecomm), September, 2010. Singapore.

Boniface Hicks and Sandra Rueda and David King and Thomas Moyer and Joshua Schiffman and Yogesh Sreenivasan and Patrick McDaniel and Trent Jaeger, *An Architecture for Enforcing End-to-End Access Control over Web Applications*, Proceedings of the Fifteenth ACM Symposium on Access Control Models and Technologies (SACMAT 2010), 163-172, June, 2010. Pittsburgh, PA.

Kevin Butler and Stephen McLaughlin and Patrick McDaniel, *Disk-Enabled Authenticated Encryption*, Proceedings of the 26th IEEE Symposium on Massive Storage Systems and Technologies (MSST), May, 2010.

Machigar Ongtang and Stephen McLaughlin and William Enck and Patrick McDaniel, *Semantically Rich Application-Centric Security in Android*, Proceedings of the 25th Annual Computer Security Applications Conference (ACSAC), 340-349, December, 2009. Honolulu, Hawaii. **(best paper)**

Thomas Moyer and Kevin Butler and Joshua Schiffman and Patrick McDaniel and Trent Jaeger, *Scalable Asynchronous Web Content Attestation*, Proceedings of the 25th Annual Computer Security Applications Conference (ACSAC), 95-104, December, 2009. Honolulu, Hawaii.

Joshua Schiffman and Thomas Moyer and Christopher Shal and Trent Jaeger and Patrick McDaniel, *Justifying Integrity Using a Virtual Machine Verifier*, Proceedings of the 25th Annual Computer Security Applications Conference (ACSAC), 83-92, December, 2009. Honolulu, Hawaii.

William Enck and Machigar Ongtang and Patrick McDaniel, *On Lightweight Mobile Phone App Certification*, Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS), 235-245, November, 2009.

Patrick Traynor and Michael Lin and Machigar Ongtang and Vikhyath Rao and Trent Jaeger and Thomas La Porta and Patrick McDaniel, *On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core*, Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS), 223-234, November, 2009.

William Enck and Patrick McDaniel and Trent Jaeger, *PinUP: Pinning User Files to Known Applications*, Proceedings of the 24th Annual Computer Security Applications Conference (ACSAC), December, 2008.

William Enck and Kevin Butler and Thomas Richardson and Patrick McDaniel and Adam Smith, *Defending Against Attacks on Main Memory Persistence*, Proceedings of the 24th Annual Computer Security Applications Conference (ACSAC), December, 2008.

Kevin Butler and Stephen McLaughlin and Patrick McDaniel, *Rootkit-Resistant Disks*, Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS), November, 2008. Alexandria, VA.

Patrick Traynor and Kevin Butler and William Enck and Patrick McDaniel, *Realizing Massive-Scale Conditional Access Systems Through Attribute-Based Cryptosystems*, ISOC Network & Distributed System Security Symposium (NDSS), February, 2008. San Diego, CA.

Luke St. Clair and Joshua Schiffman and Trent Jaeger and Patrick McDaniel, *Establishing and Sustaining System Integrity via Root of Trust Installation*, 23rd Annual Computer Security Applications Conference (ACSAC), 19-29, December, 2007. Miami, FL.

Boniface Hicks and Tim Misiak and Patrick McDaniel, *Channels: Runtime System Infrastructure for Security-typed Languages*, 23rd Annual Computer Security Applications Conference (ACSAC), 443-452, December, 2007. Miami, FL.

Dhananjay Bapat and Kevin Butler and Patrick McDaniel, *Towards Automated Privilege Separation*, Proceedings of 2nd International Conference on Information Systems Security (short paper), December, 2007. Delhi, India.

Lisa Johansen and Kevin Butler and Michael Rowell and Patrick McDaniel, *Email Communities of Interest*, Fourth Conference on Email and Anti-Spam (CEAS 2007, August, 2007. Mountain View, California.

Patrick Traynor and Patrick McDaniel and Thomas La Porta, *On Attack Causality in Internet-Connected Cellular Networks*, Proceedings of the 16th USENIX Security Symposium, 1–16, August, 2007. Boston, MA.

Boniface Hicks and Sandra Rueda and Trent Jaeger and Patrick McDaniel, *From Trusted to Secure: Building and Executing Applications that Enforce System Security*, Proceedings of the USENIX Annual Technical Conference, June, 2007. Santa Clara, CA.

William Enck and Patrick McDaniel and Shubho Sen and Panagiotis Sebos and Sylke Spoerel and Albert Greenberg and Sanjay Rao and William Aiello, *Configuration Management at Massive Scale: System Design and Experience*, Proceedings of the USENIX Annual Technical Conference, June, 2007. Santa Clara, CA.

Boniface Hicks and Sandra Rueda and Luke St. Clair and Trent Jaeger and Patrick McDaniel, *A Logical Specification and Analysis for SELinux MLS*, 12th ACM Symposium on Access Control Models and Technologies (SACMAT), ACM, June, 2007. Sophia Antipolis, France.

Anusha Sriraman and Kevin Butler and Patrick McDaniel and Padma Raghavan, *Analysis of IPv4 Address Space Delegation Structure*, 12th IEEE Symposium on Computers and Communications (ISCC), July, 2007. Aveiro, Portugal.

Heesook Choi and Thomas La Porta and Patrick McDaniel, *Privacy Preserving Communication in MANETs*, Proceedings of Fourth Annual IEEE Communications Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON 07), June, 2007. San Diego, CA.

Sophie Qiu and Patrick McDaniel and Fabian Monroe, *Toward Valley-Free Inter-domain Routing*, Proceedings of 2007 IEEE International Conference on Communications (ICC 2007), June, 2007. Glasgow, Scotland.

Sunam Ryu and Kevin Butler and Patrick Traynor and Patrick McDaniel, *Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems*, Proceedings of the 3rd IEEE International Symposium on Security in Networks and Distributed Systems (SSNDS-07), June, 2007. Niagara Falls, Canada.

Hosam Rowaihy and William Enck and Patrick McDaniel and Thomas La Porta, *Limiting Sybil Attacks in Structured Peer-to-Peer Networks*, Proceedings of IEEE INFOCOM 2007 MiniSymposium, May, 2007. Anchorage, AK.

Boniface Hicks and Kiyan Ahmadizadeh and Patrick McDaniel, *Understanding Practical Application Development in Security-Typed Languages*, 22st Annual Computer Security Applications Conference (ACSAC), 153–164, December, 2006. Miami, Fl. (**best student paper**)

Luke St. Clair and Lisa Johansen and William Enck and Matthew Pirretti and Patrick Traynor and Patrick McDaniel and Trent Jaeger, *Password Exhaustion: Predicting the End of Password Usefulness*, Proceedings of 2nd International Conference on Information Systems Security (ICISS), 37–55, December, 2006. Kolkata, India.

Kevin Butler and William Enck and Jennifer Plasterr and Patrick Traynor and Patrick McDaniel, *Privacy-Preserving Web-Based Email*, Proceedings of 2nd International Conference on Information Systems Security (ICISS), 116–131, December, 2006. Kolkata, India.

Matthew Pirretti and Patrick Traynor and Patrick McDaniel and Brent Waters, *Secure Attribute-Based Systems*, Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS), 99–112, November, 2006. Alexandria, VA.

Kevin Butler and William Aiello and Patrick McDaniel, *Optimizing BGP Security by Exploiting Path Stability*, Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS), 298-310, November, 2006. Alexandria, VA.

Patrick Traynor and William Enck and Patrick McDaniel and Thomas La Porta, *Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networks*, Proceedings of the Twelfth Annual International Conference on Mobile Computing and Networking (MobiCom), 182-193, September, 2006. Los Angeles, CA.

Patrick Traynor and Michael Chien and Scott Weaver and Boniface Hicks and Patrick McDaniel, *Non-Invasive Methods for Host Certification*, Proceedings of the Second IEEE Communications Society/CreateNet International Conference on Security and Privacy in Communication Networks (SecureComm), August, 2006. Baltimore, MD.

Sophie Qiu and Patrick McDaniel and Fabian Monrose and Avi Rubin, *Characterizing Address Use Structure and Stability of Origin Advertisement in Interdomain Routing*, 11th IEEE Symposium on Computers and Communications, 489-496, June, 2006. Pula-Cagliari, Sardinia, Italy.

Patrick McDaniel and Shubho Sen and Oliver Spatscheck and Jacobus Van der Merwe and William Aiello and Charles Kalmanek, *Enterprise Security: A Community of Interest Based Approach*, Proceedings of Network and Distributed Systems Security 2006 (NDSS), February, 2006. San Diego, CA.

Kevin Butler and Patrick McDaniel, *Understanding Mutable Internet Pathogens, or How I Learned to Stop Worrying and Love Parasitic Behavior*, Proceedings of 1st International Conference on Information Systems Security (ICISS), Springer-Verlag Lecture Notes in Computer Science, volume 3803, 36-48, December, 2005. Kolkata, India. (*Invited Paper*)

Wesam Lootah and William Enck and Patrick McDaniel, *TARP: Ticket-Based Address Resolution Protocol*, 21st Annual Computer Security Applications Conference (ACSAC), 95-103, December, 2005. Tuscon, AZ.

William Enck and Patrick Traynor and Patrick McDaniel and Thomas La Porta, *Exploiting Open Functionality in SMS-Capable Cellular Networks*, Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS), 393-404, May, 2005. Alexandria, VA.

Matthew Pirretti and Sencun Zhu and Vijaykrishnan Narayanan and Patrick McDaniel and Mahmut Kandemir and Richard Brooks, *The Sleep Deprivation Attack in Sensor Networks: Analysis and Methods of Defense*, Proceedings of the Innovations and Commercial Applications of Distributed Sensor Networks Symposium, October, 2005. Bethesda, Maryland. (**best paper**)

Louis Kruger and Somesh Jha and Patrick McDaniel, *Privacy Preserving Clustering*, 10th European Symposium on Research in Computer Security (ESORICS '05), September, 2005. Milan, Italy.

Heesook Choi and William Enck and Jaesheung Shin and Patrick McDaniel and Thomas La Porta, *Secure Reporting of Traffic Forwarding Activity in Mobile AdHoc Networks*, MobiQuitous 2005, July, 2005. San Diego, CA.

Simon Byers and Lorrie Cranor and Eric Cronin and Dave Kormann and Patrick McDaniel, *Exposing Digital Content Piracy: Approaches, Issues and Experiences*, Thirty-Eighth Conference on Signals, Systems, and Computers, 45-50, Nov, 2004. Monterey, CA. (*Invited paper*)

William Aiello and John Ioannidis and Patrick McDaniel, *Origin Authentication in Interdomain Routing*, Proceedings of 10th ACM Conference on Computer and Communications Security (CCS), ACM, 165-178, October, 2003. Washington, DC.

Eric Cronin and Sugih Jamin and Tal Malkin and Patrick McDaniel, *On the Performance, Feasibility, and Use of Forward Secure Signatures*, Proceedings of 10th ACM Conference on Computer and Communications Security (CCS), ACM, 131-144, October, 2003. Washington, DC.

Patrick McDaniel, *On Context in Authorization Policy*, 8th ACM Symposium on Access Control Models and Technologies (SACMAT), ACM, 80-8, June, 2003. Como, Italy.

Geoff Goodell and William Aiello and Tim Griffin and John Ioannidis and Patrick McDaniel and Avi Rubin, *Working Around BGP: An Incremental Approach to Improving Security and Accuracy of Interdomain Routing*, Proceedings of Network and Distributed Systems Security 2003 (NDSS), Internet Society, 75-8, February, 2003. San Diego, CA.

Patrick McDaniel and Atul Prakash, *Methods and Limitations of Security Policy Reconciliation*, 2002 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, 73-87, May, 2002. Oakland, CA.

Patrick McDaniel and Atul Prakash and Jim Irrer and Sharad Mittal and Thai-Chuin Thuang, *Flexibly Constructing Secure Groups in Antigone 2.0*, Proceedings of DARPA Information Survivability Conference and Exposition II, IEEE Computer Society Press, 55-67, June, 2001. Los Angeles, CA.

Hugh Harney and Andrea Colegrove and Patrick McDaniel, *Principles of Policy in Secure Groups*, Proceedings of Network and Distributed Systems Security 2001 (NDSS), Internet Society, 125-13, February, 2001. San Diego, CA.

Patrick McDaniel and Sugih Jamin, *Windowed Certificate Revocation*, Proceedings of IEEE INFOCOM 2000, IEEE, 1406-1414, March, 2000. Tel Aviv, Israel.

Patrick McDaniel and Avi Rubin, *A Response to ‘Can We Eliminate Certificate Revocation Lists?’*, Proceedings of Financial Cryptography 2000, International Financial Cryptography Association (IFCA), February, 2000. Anguilla, British West Indies.

Andrew Adamson and Charles J. Antonelli and Kevin Coffman and Patrick McDaniel and Jim Rees, *Secure Distributed Virtual Conferencing*, Proceedings of Communications and Multimedia Security (CMS ’99), 176-190, September, 1999. Katholieke Universiteit, Leuven, Belgium.

Patrick McDaniel and Atul Prakash and Peter Honeyman, *Antigone: A Flexible Framework for Secure Group Communication*, Proceedings of the 8th USENIX Security Symposium, 99-114, August, 1999. Washington, DC.

Workshop Papers

Yohan Beugin and Patrick McDaniel, *A Public and Reproducible Assessment of the Topics API on Real Data*, SecWeb 2024 Workshop, IEEE Security and Privacy Workshops (SPW), May, 2024.

Kunyang Li and Kyle D Domico and Jean-Charles Noirot Ferrand and Patrick McDaniel, *The Efficacy of Transformer-Based Adversarial Attacks in Security Domains*, Workshop on Artificial Intelligence for Cyber (MIL-COM 2023), October, 2023. Boston, MA.

Eric Pauley and Patrick McDaniel, *Understanding the Ethical Frameworks of Internet Measurement Studies*, The 2nd International Workshop on Ethics in Computer Security (EthiCS 2023), February, 2023. San Diego, CA. **(best paper award)**

Anshul Gandhi and Kanad Ghose and Kartik Gopalan and Syed Rafiul Hussain and Dongyoon Lee and David Liu and Zhenhua Liu and Patrick McDaniel and Shuai Mu and Erez Zadok, *Metrics for Sustainability in Data Centers*, Proceedings of the 1st Workshop on Sustainable Computer Systems Design and Implementation (Hot-Carbon’22), USENIX, July, 2022. San Diego, CA.

Leonardo Babun and Z. Berkay Celik and Patrick McDaniel and Selcuk Uluagac, *Real-time Analysis of Privacy-(un)aware IoT Applications*, Privacy Enhancing Technologies Symposium (PETS), 2021.

Sushrut Shringarputale and Patrick McDaniel and Kevin Butler and Thomas La Porta, *Co-residency Attacks on Containers are Real*, The ACM Cloud Computing Security Workshop (CCSW 2020), 2020.

Z. Berkay Celik and Patrick McDaniel, *Extending Detection with Privileged Information via Generalized Distillation*, IEEE Security & Privacy Workshop on Deep Learning and Security (IEEE S&P DLS), 2018.

Alexander Alexeev and Diane Henshel and Karl Levitt and Patrick McDaniel and Brian Rivera and Steve Templeton and Michael J. Weisma, *Constructing a Science of Cyber-Resilience for Military Systems*, Information Systems and Technology (IST) Panel, IST-153/RWS-21, CEUR Workshop Proceeding, 30-42, October, 2017.

Z. Berkay Celik and Patrick McDaniel and Rauf Izmailov, *Feature Cultivation in Privileged Information-augmented Detection*, Proceedings of the International Workshop on Security And Privacy Analytics (IWSPA 2017), 2017. (Invited Paper)

Stefan Achleitner and Thomas La Porta and Patrick McDaniel and Shridatt Sugrim and Srikanth V. Krishnamurthy and Ritu Chadha, *Cyber Deception: Virtual Networks to Defend Insider Reconnaissance*, Proceedings of the 8th ACM CCS International Workshop on Managing Insider Security Threats, ACM, 2016.

Patrick McDaniel and Robert Walls, *Estimating Attack Intent and Mission Impact from Detection Signals*, Proceedings of the NATO IST-128 Workshop: Assessing Mission Impact of Cyberattacks, Army Research Laboratory, July, 2015.

Patrick McDaniel and Trent Jaeger and Thomas La Porta and Nicolas Papernot and Robert Walls and Alexander Kott and Lisa Marvel and Ananthram Swami and Prasant Mohapatra and Srikanth V. Krishnamurthy and Iulian Neamtiu, *Security and Science of Agility*, First ACM Workshop on Moving Target Defense (MTD 2014), November, 2014. Scottsdale, AZ.

Joshua Schiffman and Thomas Moyer and Hayawardh Vijayakumar and Trent Jaeger and Patrick McDaniel, *Seeding Clouds with Trust Anchors*, Proceedings of CCSW 2010: The ACM Cloud Computing Security Workshop, October, 2010. Chicago, IL.

Stephen McLaughlin and Dmitry Podkuiko and Adam Delozier and Sergei Miadzvezhanka and Patrick McDaniel, *Embedded Firmware Diversity for Smart Electric Meters*, Proceedings of the 5th Workshop on Hot Topics in Security (HotSec '10), August, 2010. Washington, DC.

Patrick McDaniel and Kevin Butler and Stephen McLaughlin and Radu Sion and Erez Zadok and Marianne Winslett, *Towards a Secure and Efficient System for End-to-End Provenance*, the 2nd USENIX Workshop on the Theory and Practice of Provenance, February, 2010. San Jose, CA.

Thomas La Porta and Patrick McDaniel and Karl Rauscher and Jun Shu, *The Impact of Supply Chain on Information and Communications Technology Security*, In the 1st Workshop on Workshop on Telecommunications Infrastructure Protection and Security, December, 2009. Honolulu, Hawaii.

Stephen McLaughlin and Dmitry Podkuiko and Patrick McDaniel, *Energy Theft in the Advanced Metering Infrastructure*, In the 4th International Workshop on Critical Information Infrastructure Security, September, 2009. Bonn, Germany.

Matthew Blaze and Patrick McDaniel, *Below the Salt: The Dangers of Unfulfilled Physical Media Assumptions*, In Proceedings of Seventeenth International Workshop on Security Protocols, April, 2009. Cambridge, England.

Kevin Butler and William Enck and Harri Hursti and StephenMcLaughlin and Patrick Traynor and Patrick McDaniel, *Systemic Issues in the Hart InterCivic and Premier VotingSystems: Reflections Following Project EVEREST*, In Proceedings of the 3rd USENIX/ACCURATE ElectronicVoting Technology Workshop (EVT '08), July, 2008.

Kevin Butler and Stephen McLaughlin and Patrick McDaniel, *Non-Volatile Memory and Disks: Avenues for Policy Architectures*, Proceedings of the 1st ACM Computer Security Architectures Workshop, November, 2007.

Alexandria, VA.

William Enck and Sandra Rueda and Yogesh Sreenivasan and JoshuaSchiffman and Luke St. Clair and Trent Jaeger and Patrick McDaniel, *Protecting Users from "Themselves"*, Proceedings of the 1st ACM Computer Security Architectures Workshop, November, 2007. Alexandria, VA.

Boniface Hicks and David King and Patrick McDaniel, *Jifclipse: Development Tools for Security-Typed Applications*, Proceedings of the 2nd ACM SIGPLAN Workshop on Programming Languages and Analysis for Security (PLAS '07), ACM Press, June, 2007. San Diego, CA.

Boniface Hicks and Sandra Rueda and Trent Jaeger andPatrick McDaniel, *Integration of SELinux and Security-typed Languages*, Proceedings of the 2007 Security-Enhanced Linux Workshop, March, 2007. Baltimore, MD.

Sophie Qiu and Fabian Monroe and Andreas Terzis and Patrick McDaniel, *Efficient Techniques for Detecting False Origin Advertisements in Inter-domain Routing*, Proceedings of The Second Workshop on Secure Network Protocols (NPSec), November, 2006. Santa Barbara.

Shiva Chaitanya and Kevin Butler and Patrick McDaniel andAnand Sivasubramaniam, *Design, Implementation and Evaluation of Security iniSCSI-based Network Storage Systems*, Proceedings of 2nd International Workshop on StorageSecurity and Survivability (StorageSS 2006), October, 2006. Alexandria, Virginia.

Trent Jaeger and Patrick McDaniel and Luke St. Clair and RamonCaceres and Reiner Sailer, *Shame on Trust in Distributed Systems*, Proceedings of the First Workshop on Hot Topics inSecurity (HotSec '06), July, 2006. Vancouver, B.C., Canada.

Kevin Butler and Patrick McDaniel and Sophie Qiu, *BGPRV: A Library for Fast and Efficient Routing Data Manipulation*, Proceedings of DETER Community Workshop, June, 2006. Arlington, VA.

Kevin Butler and Patrick McDaniel, *Testing Large Scale BGP Security in Replayable NetworkEnvironments*, Proceedings of DETER Community Workshop, June, 2006. Arlington, VA.

Boniface Hicks and David King and Patrick McDaniel andMichael Hicks, *Trusted Declassification: High-level Policy for aSecurity-Typed Language*, Proceedings of ACM SIGPLAN Workshop on ProgrammingLanguages and Analysis for Security, 65-74, June, 2006. Ottawa, Canada.

Ali Al-Lawati and Dongwon Lee and Patrick McDaniel, *Blocking in Private Information Matching*, Proceedings of Second International ACM SIGMOD Workshopon Information Quality in Information Systems, June, 2005. Baltimore, MD.

William Aiello and Charles Kalmanek and Patrick McDaniel andShubho Sen and Oliver Spatscheck and Jacobus Van der Merwe, *Analysis of Communities Of Interest in Data Networks*, Passive and Active Measurement Workshop 2005, March, 2005. Boston, M.

Simon Byers and Lorrie Cranor and David Kormann andPatrick McDaniel, *Searching for Privacy: Design and Implementation of aP3P-Enabled Search Engine*, Proceedings of 2004 Workshop on Privacy EnhancingTechnologies (PETS), May, 2004. Toronto, Canad.

Hao Wang and Somesh Jha and Patrick McDaniel and MironLivn, *Security Policy Reconciliation in Distributed ComputingEnvironments*, Proceedings of 5th International Workshop on Policiesfor Distributed Systems and Networks (Policy 2004), IEEE Computer Society Press, 137-146, June, 2004. Yorktown Heights, N.

Simon Byers and Lorrie Cranor and Eric Cronin and DaveKormann and Patrick McDaniel, *Analysis of Security Vulnerabilities in the Movie Productionand Distribution Process*, Proceedings of 2003 ACM Workshop on Digital RightsManagement, ACM, October, 2003. Washington, DC, also appeared in Telecommunications PolicyResearch Conference – September 2003.

Patents

U.S. Patent No. 7,873,350, Patrick McDaniel and Martin Strauss, *End-to-end secure wireless communication for requesting a more secure channel*, January, 2011.

U.S. Patent No. 7,975,044, Oliver Spatscheck and Subhabrata Sen and Jacobus Van der Merwe and Patrick McDaniel, *Automated disambiguation of fixed-serverport-based applications from ephemeral applications*, July, 2011.

U.S. Patent No. 8,175,580, Patrick McDaniel and Martin Strauss, *End-to-end secure wireless communication for requesting a more secure channel*, May, 2012.

U.S. Patent No. 8,453,227, William Aiello and Charles Kalmanek and William Leighton III and Patrick McDaniel and Subhabrata Sen and Oliver Spatscheck and Jacobus Van der Merwe, *Reverse firewall with self-provisioning*, May, 2013.

U.S. Patent No. 8,732,293, Patrick McDaniel and Subhabrata Sen and Oliver Spatscheck and Jacobus Van der Merwe, *System and method for tracking individuals on a data network using communities of interest*, May, 2014.

U.S. Patent No. 8,813,213, William Aiello and Charles Kalmanek and William Leighton III and Patrick McDaniel and Subhabrata Sen and Oliver Spatscheck and Jacobus Van der Merwe, *Reverse firewall with self-provisioning*, August, 2014.

Other Publications

Patrick McDaniel and Farinaz Koushanfar, *NSF Secure and Trustworthy Computing 2.0 Vision Statement*, Public Report, August, 2023.

Patrick McDaniel and Thorsten Holz and Indra Spiecker and Genannt Dohmann and Christopher Burchard and Ahmad-Reza Sadeghi and Konrad Rieck and Kamalika Chaudhuri and Somesh Jha and Andrea Matwyshyn and David Evans and Felix Freiling and Amy Hasan, *Cybersecurity and Machine Learning: Vision Document, Report on the joint NSF/DFG Cybersecurity and Machine Learning Research Workshop*, Public Report, National Science Foundation/Deutsche Forschungsgemeinschaft, December, 2021.

Patrick McDaniel and John Launchbury, *Artificial Intelligence and Cybersecurity: Opportunities and Challenges 2019 Technical Workshop Report*, Public Report, Networking and Information Technology Research and Development Subcommittee, Machine Learning and Artificial Intelligence Subcommittee, and the Special Cyber Operations Research and Engineering Subcommittee of the National Science and Technology Council, 2020.

Patrick McDaniel and Avi Rubin, *Conference Proceedings*, 2008 IEEE Symposium on Security and Privacy, IEEE, May, 2008. Oakland, CA.

Patrick McDaniel and Kevin Butler and William Enck and Harri Hursti and Stephen McLaughlin and Patrick Traynor and Matthew Blaze and Adam Aviv and Pavol Cerny and Sandy Clark and Eric Cronin and Gaurav Shah and Micah Sherr and Giovanni Vigna and Richard Kemmerer and David Balzarotti and Greg Banks and Marco Cova and Viktoria Felmetzger and William Robertson and Fredrik Valeur and Joseph Lorenzo Hall and Laura Quilter, *EVEREST: Evaluation and Validation of Election-Related Equipment, Standards and Testing*, Ohio Secretary of State, Public Report, 2007.

Patrick McDaniel and Shyam K. Gupta, *Conference Proceedings*, The Third International Conference Information Systems Security, Springer, December, 2007. Delhi, India.

Birgit Pfitzmann and Patrick McDaniel, *Conference Proceedings*, 2007 IEEE Symposium on Security and Privacy, IEEE, May, 2007. Oakland, CA.

Patrick McDaniel, *Conference Proceedings*, The 14th USENIX Security Symposium, USENIX, August, 2005.

Baltimore, MD.

Patrick McDaniel, *Policy Evolution: Autonomic Environmental Security*, Software Engineering Research Center Showcase, USA, December, 2004.

Hugh Harney and Uri Meth and Andrea Colegrove and Angela Schuett and Patrick McDaniel and Gavin Kenny and Haitham Cruickshank and Sunil Iyengar, *GSAKMP (Draft)*, Internet Research Task Force, August, 2003.

Patrick McDaniel and Atul Prakash, *A Flexible Architecture for Security Policy Enforcement*, Proceedings of DARPA Information Survivability Conference and Exposition III, Research Summaries, 234-239, April, 2003.

Jim Irrer and Atul Prakash and Patrick McDaniel, *Antigone: Policy-Based Secure Group Communications Systems and AMirD: Antigone-Based Secure File Mirroring System*, Proceedings of DARPA Information Survivability Conference and Exposition III, Demo Summaries, 44-4, April, 2003.

Patrick McDaniel and Sugih Jamin, *Windowed Key Revocation in Public Key Infrastructures*, NASA Tech Briefs, 55, August, 2002.

Patrick McDaniel and Atul Prakash, *Antigone Secure Group Communication System*, NASA Tech Briefs, 2001.

Patrick McDaniel, *Policy Management in Secure Group Communication*, PhD Thesis, University of Michigan, Ann Arbor, MI, August, 2001.

Patrick McDaniel and Hugh Harney and Andrea Colegrove and Atul Prakash and Peter Dinsmore, *Multicast Security Policy Requirements and Building Blocks (Draft)*, Internet Research Task Force, Secure Multicast Research Group (SMuG), November, 2000.

Tom Hardjono and Hugh Harney and Patrick McDaniel and Andrea Colegrove and Peter Dinsmore, *Group Security Policy Token (Draft)*, Internet Research Task Force, Secure Multicast Research Group (SMuG), November, 2001.

Patrick McDaniel and Hugh Harney and Peter Dinsmore and Atul Prakash, *Multicast Security Policy (Draft)*, Internet Research Task Force, Secure Multicast Research Group (SMuG), June, 2000.

Patrick McDaniel, *8th USENIX Security Symposium Conference Summaries, Potpourri Session*, USENIX Login Magazine, 9-12, November, 1999.

Patrick McDaniel, *The Analysis of D_i , a Detailed Design Metric on Large Scale Software*, Masters Thesis, Ball State University, Muncie, IN, June, 1991.

Technical Reports

Quinn Burke and Ryan Sheatsley and Rachel King and Michael Swift and Patrick McDaniel, *Cloud Storage Integrity at Scale: A Case for Dynamic Hash Trees*, Technical Report 2405.03830, arXiv preprint, May, 2024.

Alban Héon and Ryan Sheatsley and Quinn Burke and Blaine Hoak and Eric Pauley and Yohan Beugin and Patrick McDaniel, *Systematic Evaluation of Geolocation Privacy Mechanisms*, Technical Report 2309.06263, arXiv preprint, September, 2023.

Yohan Beugin and Patrick McDaniel, *Interest-disclosing Mechanisms for Advertising are Privacy-Exposing (Not Preserving)*, Technical Report 2306.03825, arXiv preprint, May, 2023.

Quinn Burke and Yohan Beugin and Blaine Hoak and Rachel King and Eric Pauley and Ryan Sheatsley and Mingli Yu and Ting He and Thomas La Porta and Patrick McDaniel, *Securing Cloud File Systems using Shielded Execution*, Technical Report 2305.18639, arXiv preprint, May, 2023.

Yohan Beugin and Quinn Burke and Blaine Hoak and Ryan Sheatsley and Eric Pauley and Gang Tan and Syed Rafiul Hussain and Patrick McDaniel, *Privacy-Preserving Protocols for Smart Cameras and Other IoT Devices*, Technical Report 2208.09776, arXiv preprint, August, 2022.

Pauley, Eric and Domico, Kyle and Hoak, Blaine and Sheatsley, Ryan and Burke, Quinn and Beugin, Yohan and McDaniel, Patrick, *EIPSIM: Modeling Secure IP Address Allocation at Cloud Scale*, Technical Report arXiv:2210.14999, arXiv preprint, 2022.

Bolor-Erdene Zolbayarn and Ryan Sheatsley and Patrick McDaniel and Michael J. Weisman and Sencun Zhu and Shitong Zhu and Srikanth V. Krishnamurthy, *Generating Practical Adversarial Network Traffic Flows using NIDSGAN*, Technical Report arXiv:2203.06694, arXiv preprint, March, 2022.

Ryan Sheatsley and Blaine Hoak and Eric Pauley and Yohan Beugin and Michael J. Weisman and Patrick McDaniel, *On the Robustness of Domain Constraints*, Technical Report arXiv:2105.08619, arXiv preprint, 2021.

Ryan Sheatsley and Nicolas Papernot and Michael J. Weisman and Gunjan Verma and Patrick McDaniel, *Adversarial Examples in Constrained Domains*, Technical Report arXiv:2011.01183, arXiv preprint, 2020.

Stefan Achleitner and Quinn Burke and Patrick McDaniel and Trent Jaeger and Thomas La Porta and Srikanth V. Krishnamurthy, *MLSNet: A Policy Complying Multilevel Security Framework for Software Defined Networking*, Technical Report INSR-500-TR-0500-2019, Institute of Networking and Security Research, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, January, 2019.

Michael Norris and Z. Berkay Celik and Patrick McDaniel and Gang Tan and Prasanna Venkatesh and Shulin Zhao and Anand Sivasubramaniam, *IoTRepair: Systematically Addressing Device Faults in Commodity IoT (Extended Paper)*, Technical Report arXiv:2002.07641, arXiv preprint, 2020.

Amit Kumar Sikder and Leonardo Babun and Z. Berkay Celik and Abbas Acar and Hidayet Aksu and Patrick McDaniel and Engin Kirda and Selcuk Uluagac, *KRATOS: Multi-User Multi-Device-Aware Access Control System for the Smart Home*, Technical Report arXiv:1911.10186, arXiv preprint, 2020.

Leonardo Babun and Z. Berkay Celik and Patrick McDaniel and Selcuk Uluagac, *Real-time Analysis of Privacy-(un)aware IoT Applications*, Technical Report arXiv:1911.10461, arXiv preprint, 2019.

Dan Boneh and Andrew J. Grotto and Patrick McDaniel and Nicolas Papernot, *How Relevant is the Turing Test in the Age of Sophisbots?*, Technical Report arXiv:1909.00056, arXiv preprint, 2019.

Chun-Ming Lai and Xiaoyun Wang and Jon W. Chapman and Yu-Cheng Lin and Yu-Chang Ho and Felix Wu and Patrick McDaniel and Hasan Cam, *More or Less? Predict the Social Influence of Malicious URLs on Social Media*, Technical Report arXiv:1812.02978, arXiv preprint, 2018.

Dang Tu Nguyen and Chengyu Song and Zhiyun Qian and Srikanth V. Krishnamurthy and Edward J. M. Colbert and Patrick McDaniel, *IoTSan: Fortifying the Safety of IoT Systems*, Technical Report arXiv:1810.09551, arXiv preprint, 2018.

Z. Berkay Celik and Earlence Fernandes and Eric Pauley and Gang Tan and Patrick McDaniel, *Program Analysis of Commodity IoT Applications for Security and Privacy: Challenges and Opportunities*, Technical Report arXiv:1809.06962, arXiv preprint, 2018.

Giuseppe Petracca and Jens Grossklags and Patrick McDaniel and Trent Jaeger, *Regulating Access to System Sensors in Cooperating Programs*, Technical Report arXiv:1808.05579, arXiv preprint, 2018.

Z. Berkay Celik and Patrick McDaniel and Gang Tan, *Soteria: Automated IoT Safety and Security Analysis*, Technical Report arXiv:1805.08876, arXiv preprint, 2018.

Nicolas Papernot and Patrick McDaniel, *Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning*, Technical Report arXiv:1803.04765, arXiv preprint, 2018.

Z. Berkay Celik and Leonardo Babun and Amit Kumar Sikder and Hidayet Aksu and Gang Tan and Patrick McDaniel and Selcuk Uluagac, *Sensitive Information Tracking in Commodity IoT*, Technical Report arXiv:1802.08307, arXiv preprint, 2018.

Chun-Ming Lai and Xiaoyun Wang and Yunfeng Hong and Yu-Cheng Lin and Felix Wu and Patrick McDaniel and Hasan Cam, *Attacking Strategies and Temporal Analysis Involving Facebook Discussion Groups*, Technical Report arXiv:1802.04500, arXiv preprint, 2017.

Abbas Acar and Z. Berkay Celik and Hidayet Aksu and Selcuk Uluagac and Patrick McDaniel, *Achieving Secure and Differentially Private Computations in Multiparty Settings*, Technical Report arXiv:1707.01871, arXiv preprint, 2017.

Florian Tramer and Alexey Kurakin and Nicolas Papernot and Ian Goodfellow and Dan Boneh and Patrick McDaniel, *Ensemble Adversarial Training: Attacks and Defenses*, Technical Report arXiv:1705.07204, arXiv preprint, 2017.

Nicolas Papernot and Patrick McDaniel, *Extending Defensive Distillation*, Technical Report arXiv:1705.05264, arXiv preprint, 2017.

Florian Tramer and Nicolas Papernot and Ian Goodfellow and Dan Boneh and Patrick McDaniel, *The Space of Transferable Adversarial Examples*, Technical Report arXiv:1704.03453, arXiv preprint, 2017.

Z. Berkay Celik and Hidayet Aksu and Abbas Acar and Ryan Sheatsley and Selcuk Uluagac and Patrick McDaniel, *Curie: Policy-based Secure Data Exchange*, Technical Report arXiv:1702.08342, arXiv preprint, 2017.

Kathrin Grosse and Praveen Manoharan and Nicolas Papernot and Michael Backes and Patrick McDaniel, *On the (Statistical) Detection of Adversarial Examples*, Technical Report arXiv:1702.06280, arXiv preprint, 2017.

Z. Berkay Celik and David Lopez-Paz and Patrick McDaniel, *Patient-Driven Privacy Control through Generalized Distillation*, Technical Report arXiv:1611.08648, arXiv preprint, 2017.

Nicolas Papernot and Patrick McDaniel and Arunesh Sinha and Michael Wellman, *Towards the Science of Security and Privacy in Machine Learning*, Technical Report arXiv:1611.03814, arXiv preprint, 2016.

Nicolas Papernot and Ian Goodfellow and Ryan Sheatsley and Reuben Feinman and Patrick McDaniel, *cleverhans v1.0.0: an adversarial machine learning library*, Technical Report arXiv:1610.0076, arXiv preprint, 2016.

Nicolas Papernot and Patrick McDaniel, *On the Effectiveness of Defensive Distillation*, Technical Report arXiv:1607.05113, arXiv preprint, 2016.

Kathrin Grosse and Nicolas Papernot and Praveen Manoharan and Michael Backes and Patrick McDaniel, *Adversarial Perturbations Against Deep Neural Networks for Malware Classification*, Technical Report arXiv:1606.04435, arXiv preprint, 2016.

Nicolas Papernot and Patrick McDaniel and Ian Goodfellow, *Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples*, Technical Report arXiv:1605.07277, arXiv preprint, 2016.

Nicolas Papernot and Patrick McDaniel and Ananthram Swami and Richard Harang, *Crafting Adversarial Input Sequences for Recurrent Neural Networks*, Technical Report arXiv:1604.08275, arXiv preprint, 2016.

Z. Berkay Celik and Patrick McDaniel and Rauf Izmailov and Nicolas Papernot and Ryan Sheatsley and Raquel Alvarez and Ananthram Swami, *Detection under Privileged Information*, Technical Report arXiv:1603.09638,

arXiv preprint, 2016.

Vaibhav Rastogi and Drew Davidson and Lorenzo De Carli and Somesh Jha and Patrick McDaniel, *Towards Least Privilege Containers with Cimplifier*, Technical Report arXiv:1602.08410, arXiv preprint, 2016.

Nicolas Papernot and Patrick McDaniel and Ian Goodfellow and Somesh Jha and Z. Berkay Celik and Ananthram Swami, *Practical Black-Box Attacks against Machine Learning*, Technical Report arXiv:1602.02697, arXiv preprint, 2016.

Nicolas Papernot and Patrick McDaniel and Somesh Jha and Matthew Fredrikson and Z. Berkay Celik and Ananthram Swami, *The Limitations of Deep Learning in Adversarial Settings*, Technical Report arXiv:1511.07528, arXiv preprint, 2015.

Nicolas Papernot and Patrick McDaniel and Xi Wu and Somesh Jha and Ananthram Swami, *Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks*, Technical Report arXiv:1511.04508, arXiv preprint, 2015.

Alexander Kott and Ananthram Swami and Patrick McDaniel, *Six Potential Game-Changers in Cyber Security: Towards Priorities in Cyber Science and Engineering*, Technical Report arXiv:1511.00509, arXiv preprint, 2015.

Li Li and Alexandre Bartel and Jacques Klein and Yves Le Traon and Steven Arzt and Siegfried Rasthofer and Eric Bodden and Damien Ochteau and Patrick McDaniel, *I know what leaked in your pocket: uncovering privacy leaks on Android Apps with Static Taint Analysis*, Technical Report arXiv:1404.7431, arXiv preprint, 2014.

Devin Pohly and Patrick McDaniel, *Modeling Privacy and Tradeoffs in Multichannel Secret Sharing Protocols*, Technical Report NAS-TR-0188-2016, Institute of Networking and Security Research, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, January, 2016.

Nicolas Papernot and Patrick McDaniel and Somesh Jha and Matthew Fredrikson and Z. Berkay Celik and Ananthram Swami, *The Limitations of Deep Learning in Adversarial Settings*, Technical Report NAS-TR-0172-2014, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, October, 2016.

Wenhui Hu and Damien Ochteau and Patrick McDaniel and Peng Liu, *Duet: Library Integrity Verification for Android Applications*, Technical Report NAS-TR-0172-2014, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, February, 2016.

Devin Pohly and Stephen McLaughlin and Patrick McDaniel and Kevin Butler, *Hi-Fi: Collecting High-Fidelity Whole-System Provenance*, Technical Report NAS-TR-0160-2012, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, June, 2016.

Damien Ochteau and Somesh Jha and Patrick McDaniel, *Retargeting Android Applications to Java Bytecode*, Technical Report NAS-TR-0150-2011, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, September, 2011.

Stephen McLaughlin and Patrick McDaniel, *Protecting Consumer Privacy from Electric Load Monitoring*, Technical Report NAS-TR-0147-2011, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, March, 2011.

William Enck and Patrick McDaniel, *Federated Information Flow Control for Mobile Phones*, Technical Report NAS-TR-0136-2010, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, July, 2010.

Kevin Butler and Stephen McLaughlin and Patrick McDaniel, *Kells: A Protection Framework for Portable Data*, Technical Report NAS-TR-0134-2010, Network and Security Research Center, Department of Computer Science

and Engineering, Pennsylvania State University, University Park, PA, USA, June, 2010.

Stephen McLaughlin and Dmitry Podkuiko and Adam Delozier and Sergei Miadzvezhanka and Patrick McDaniel, *Multi-vendor Penetration Testing in the Advanced Metering Infrastructure*, Technical Report NAS-TR-0133-2010, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, June, 2010.

Machigar Ongtang and Kevin Butler and Patrick McDaniel, *Porscha: Policy Oriented Secure Content Handling in Android*, Technical Report NAS-TR-0132-2010, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, June, 2010.

Joshua Schiffman and Thomas Moyer and Hayawardh Vijayakumar and Trent Jaeger and Patrick McDaniel, *Seeding Clouds with Trust Anchors*, Technical Report NAS-TR-0127-2010, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, April, 2010.

William Enck and Machigar Ongtang and Patrick McDaniel, *TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones*, Technical Report NAS-TR-0120-2010, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, February, 2010.

Joshua Schiffman and Thomas Moyer and Christopher Shal and Trent Jaeger and Patrick McDaniel, *Justifying Integrity Using a Virtual Machine Verifier*, Technical Report NAS-TR-0119-2009, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, August, 2010.

Machigar Ongtang and Stephen McLaughlin and William Enck and Patrick McDaniel, *Semantically Rich Application-Centric Security in Android*, Technical Report NAS-TR-0116-2009, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, June, 2010.

Stephen McLaughlin and Dmitry Podkuiko and Patrick McDaniel, *Energy Theft in the Advanced Metering Infrastructure*, Technical Report NAS-TR-0115-2009, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, June, 2010.

Kevin Butler and Stephen McLaughlin and Thomas Moyer and Joshua Schiffman and Patrick McDaniel and Trent Jaeger, *Firma: Disk-Based Foundations for Trusted Operating Systems*, Technical Report NAS-TR-0114-2009, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, April, 2010.

William Enck and Machigar Ongtang and Patrick McDaniel, *On Lightweight Mobile Phone App Certification*, Technical Report NAS-TR-0113-2009, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, April, 2010.

Patrick Traynor and Michael Lin and Machigar Ongtang and Vikhyath Rao and Thomas La Porta and Patrick McDaniel, *On Cellular Botnets: Measuring the Impact of Malicious Devices on the Network Core*, Technical Report NAS-TR-0110-2009, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, March, 2010.

Boniface Hicks and Sandra Rueda and Yogesh Sreenivasan and Guruprasad Jakka and David King and Trent Jaeger and Patrick McDaniel, *An Architecture for Enforcing End-to-End Security Over Web Applications*, Technical Report NAS-TR-0104-2009, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, January, 2009.

Joshua Schiffman and Thomas Moyer and Christopher Shal and Trent Jaeger and Patrick McDaniel, *No Node Is an Island: Shamon Integrity Monitoring Approach*, Technical Report NAS-TR-0103-2009, Network and Security

Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, February, 2009.

Patrick Traynor and Michael Lin and Machigar Ongtang and Vikhyath Rao and Trent Jaeger and Thomas La Porta and Patrick McDaniel, *On Cellular Botnets: Measuring the Impact of Malicious Devices on the Network Core*, Technical Report NAS-TR-0099-2008, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, November, 2008.

Kevin Butler and Stephen McLaughlin and Thomas Moyer and Patrick McDaniel and Trent Jaeger, *SwitchBlade: Policy-Driven Disk Segmentation*, Technical Report NAS-TR-0098-2008, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, November, 2008.

Thomas Moyer and Kevin Butler and Joshua Schiffman and Patrick McDaniel and Trent Jaeger, *Scalable Asynchronous Web Content Attestation*, Technical Report NAS-TR-0095-2008, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, September, 2008.

William Enck and Machigar Ongtang and Patrick McDaniel, *Automated Cellphone Application Certification in Android (or) Mitigating Phone Software Misuse Before It Happens*, Technical Report NAS-TR-0094-2008, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, September, 2008.

Boniface Hicks and Sandra Rueda and Luke St. Clair and Trent Jaeger and Patrick McDaniel, *A Logical Specification and Analysis for SELinux MLS Policy*, Technical Report NAS-TR-0091-2008, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, July, 2008.

Kevin Butler and Stephen McLaughlin and Patrick McDaniel, *Rootkit-Resistant Disks*, Technical Report NAS-TR-0089-2008, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, April, 2008.

Patrick Traynor and Joshua Schiffman and Thomas La Porta and Patrick McDaniel and Abhrajit Ghosh and Farooq Anjum, *Constructing Secure Localization Systems with Adjustable Granularity*, Technical Report NAS-TR-0084-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, December, 2007.

Stephen McLaughlin and Kevin Butler and William Enck and Patrick McDaniel, *Genbd - A Generic Block Device*, Technical Report NAS-TR-0082-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, November, 2007.

Kevin Butler and Stephen McLaughlin and Patrick McDaniel, *Non-Volatile Memory and Disks: Avenues for Policy Architectures*, Technical Report NAS-TR-0074-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, June, 2007.

William Enck and Sandra Rueda and Joshua Schiffman and Yogesh Sreenivasan and Luke St. Clair and Trent Jaeger and Patrick McDaniel, *Protecting Users From "Themselves"*, Technical Report NAS-TR-0073-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, June, 2007.

Dhananjay Bapat and Kevin Butler and Patrick McDaniel, *Towards Automated Privilege Separation*, Technical Report NAS-TR-0071-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, May, 2007.

Patrick Traynor and Kevin Butler and William Enck and Patrick McDaniel, *Realizing Massive-Scale Conditional Access Systems Through Attribute-Based Cryptosystems*, Technical Report NAS-TR-0070-2007, Network

and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, May, 2007.

Luke St. Clair and Joshua Schiffman and Trent Jaeger and Patrick McDaniel, *Establishing and Sustaining System Integrity via Root of Trust Installation*, Technical Report NAS-TR-0067-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, April, 2007.

Boniface Hicks and David King and Patrick McDaniel, *Jifclipse: Development Tools for Security-Typed Language*, Technical Report NAS-TR-0065-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, April, 2007.

William Enck and Patrick McDaniel and Trent Jaeger, *Protecting User Files by Reducing Application Access*, Technical Report NAS-TR-0063-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, February, 2007.

Boniface Hicks and Sandra Rueda and Trent Jaeger and Patrick McDaniel, *A Logical Specification and Analysis for SELinux MLSPolicy*, Technical Report NAS-TR-0058-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, January, 2007.

Boniface Hicks and Sandra Rueda and Trent Jaeger and Patrick McDaniel, *From Trusted to Secure: Building and Executing Applications that Enforce System Security*, Technical Report NAS-TR-0061-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, January, 2007.

Lisa Johansen and Kevin Butler and William Enck and Patrick Traynor and Patrick McDaniel, *Grains of SANs: Building Storage Area Networks from Memory Spots*, Technical Report NAS-TR-0060-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, January, 2007.

Patrick Traynor and Vikhyath Rao and Trent Jaeger and Patrick McDaniel and Thomas La Porta, *From Mobile Phones to Responsible Devices*, Technical Report NAS-TR-0059-2007, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, January, 2007.

Anusha Sriraman and Kevin Butler and Patrick McDaniel and Padma Raghavan, *Analysis of the IPv4 Address Space Delegation Structure*, Technical Report NAS-TR-0057-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, December, 2006.

Luke St. Clair and Joshua Schiffman and Trent Jaeger and Patrick McDaniel, *Sum of the Parts: Composing Trust from Validation Primitives*, Technical Report NAS-TR-0056-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, November, 2006.

Sophie Qiu and Patrick McDaniel and Fabian Monrose, *Toward Valley-Free Inter-domain Routing*, Technical Report NAS-TR-0054-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, October, 2006.

Boniface Hicks and Sandra Rueda and Trent Jaeger and Patrick McDaniel, *Integrating SELinux with Security-typed Languages*, Technical Report NAS-TR-0052-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, October, 2006.

Patrick Traynor and William Enck and Patrick McDaniel and Thomas La Porta, *Mitigating Attacks on Open Functionality in SMS-Capable Networks*, Technical Report NAS-TR-0051-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, October, 2006.

USA, October, 2006.

Patrick Traynor and Kevin Butler and William Enck and Kevin Borders and Patrick McDaniel, *malnets: Large-Scale Malicious Networks via Compromised Wireless Access Points*, Technical Report NAS-TR-0048-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, September, 2006.

Boniface Hicks and Sandra Rueda and Trent Jaeger and Patrick McDaniel, *Breaking Down the Walls of Mutual Distrust: Security-typed Email Using Labeled IPsec*, Technical Report NAS-TR-0049-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, September, 2006.

Sunam Ryu and Kevin Butler and Patrick Traynor and Patrick McDaniel, *Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems*, Technical Report NAS-TR-0043-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, August, 2006.

Wesam Lootah and William Enck and Patrick McDaniel, *TARP: Ticket-based Address Resolution Protocol (extended version)*, Technical Report NAS-TR-0046-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, August, 2006.

Patrick McDaniel, *Understanding Equivalence in High-Level and Information Flow Policy*, Technical Report NAS-TR-0042-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, July, 2006.

Trent Jaeger and Patrick McDaniel and Luke St. Clair and Ramon Caceres and Reiner Sailer, *Shame on Trust in Distributed Systems*, Technical Report RC239664 (W0605-129), IBM Research Division, Yorktown Heights, NY, May, 2006.

Lisa Johansen and Kevin Butler and Michael Rowell and Patrick McDaniel, *Email Communities of Interest*, Technical Report NAS-TR-0040-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, May, 2006.

Boniface Hicks and Kiyam Ahmadizadeh and Patrick McDaniel, *From Languages to Systems: Understanding Practical Application Development in Security-typed Languages*, Technical Report NAS-TR-0035-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, April, 2006.

Boniface Hicks and David King and Patrick McDaniel and Michael Hicks, *Trusted Declassification: High-level policy for a security-typed language*, Technical Report NAS-TR-0033-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, March, 2006.

Matthew Pirretti and Patrick Traynor and Patrick McDaniel and Brent Waters, *Secure Attribute-Based Systems*, Technical Report NAS-TR-0028-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, February, 2006.

William Enck and Kevin Butler and Thomas Richardson and Patrick McDaniel, *Securing Non-Volatile Main Memory*, Technical Report NAS-TR-0029-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, February, 2006.

Luke St. Clair and Lisa Johansen and William Enck and Matthew Pirretti and Patrick Traynor and Patrick McDaniel and Trent Jaeger, *Password Exhaustion: Predicting the End of Password Usefulness*, Technical Report NAS-TR-0030-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, February, 2006.

Heesook Choi and Patrick McDaniel and Thomas La Porta, *Privacy Preserving Communication in MANETs*, Technical Report NAS-TR-0031-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, USA, December, 2005.

Matthew Pirretti and Vijaykrishnan Narayanan and PatrickMcDaniel and Bharat Madan, *SLAT: Secure Localization with Attack Tolerance*, Technical Report NAS-TR-0024-2005, Network and Security Research Center, Department of Computer Science and Engineering,Pennsylvania State University, University Park, PA, USA, August, 2005.

Patrick Traynor and Michael Chien and Scott Weaver andBoniface Hicks and Patrick McDaniel, *Non-Invasive Methods for Host Certification*, Technical Report NAS-TR-0025-2005, Network and Security Research Center, Department of Computer Science and Engineering,Pennsylvania State University, University Park, PA, USA, September, 2005.

Sophie Qiu and Patrick McDaniel and Fabian Monroe andAvi Rubin, *Characterizing Address Use Structure and Stabillity ofOrigin Advertizement in Interdomain Routing*, Technical Report NAS-TR-0018-2005, Network and Security Research Center, Department of Computer Science and Engineering,Pennsylvania State University, University Park, PA, USA, July, 2005.

Hosam Rowaihy and William Enck and Patrick McDaniel andThomas La Porta, *Limiting Sybil Attacks in Structured Peer-to-Peer Networks*, Technical Report NAS-TR-0017-2005, Network and Security Research Center, Department of Computer Science and Engineering,Pennsylvania State University, University Park, PA, USA, July, 2005.

Boniface Hicks and Patrick McDaniel and Ali Hurson, *Information Flow Control in Database Security: A CaseStudy for Secure Programming with Jif*, Technical Report NAS-TR-0011-2005, Network and Security Research Center, Department of Computer Science and Engineering,Pennsylvania State University, University Park, PA, USA, April, 2005.

Wesam Lootah and William Enck and Patrick McDaniel, *TARP: Ticket-Based Address Resolution Protocol*, Technical Report NAS-TR-0010-2005, Network and Security Research Center, Department of Computer Science and Engineering,Pennsylvania State University, University Park, PA, USA, June, 2005.

Patrick Traynor and Kevin Butler and William Enck andJennifer Plasterr and Scott Weaver and John van Bramer and PatrickMcDaniel, *Privacy-Preserving Web-Based Email*, Technical Report NAS-TR-0009-2005, Network and Security Research Center, Department of Computer Science and Engineering,Pennsylvania State University, University Park, PA, USA, June, 2005.

William Enck and Patrick Traynor and Patrick McDaniel andThomas La Porta, *Exploiting Open Functionality in SMS-Capable Cellular Networks*, Technical Report NAS-TR-0007-2005, Network and Security Center, Department of Computer Science, Pennsylvania State University, May, 2005.

Boniface Hicks and David King and Patrick McDaniel, *Declassification with Cryptographic Functions in aSecurity-Typed Language*, Technical Report NAS-TR-0004-2005, Network and Security Center, Department of Computer Science, Pennsylvania State University, January, 2005.

William Aiello and Kevin Butler and Patrick McDaniel, *Path Authentication in Interdomain Routing*, Technical Report TR NAS-TR-0002-2004, Network and Security Center, Department of Computer Science and Engineering, Penn State University, November, 2004.

Dan Pei and William Aiello and Anna Gilbert and PatrickMcDaniel, *Origin Disturbances in BGP*, Technical Report TD-62TJFF8, AT&T Labs - Research, Florham Park, NJ, July, 2004.

Kevin Butler and Toni Farley and Patrick McDaniel andJennifer Rexford, *A Survey of BGP Security Issues and Solutions*, Technical Report TD-5UGJ33, AT&T Labs - Research, Florham Park, NJ, February, 2004.

Patrick McDaniel and Atul Prakash, *Securing Distributed Applications Using a Policy-based Approach*, Technical Report TD-5UDKVD, AT&T Labs - Research, Florham Park, NJ, December, 2003.

William Aiello and John Ioannidis and Patrick McDaniel, *Origin Authentication in Interdomain Routing*, Technical Report TD-5QHG2G, AT&T Labs - Research, Florham Park, NJ, August, 2003.

Eric Cronin and Sugih Jamin and Tal Malkin and Patrick McDaniel, *On the Performance, Feasibility, and Use of Forward Secure Signatures*, Technical Report TD-5QHGBK, AT&T Labs - Research, Florham Park, NJ, August, 2003.

Simon Byers and Lorrie Cranor and Eric Cronin and Dave Kormann and Patrick McDaniel, *Analysis of Security Vulnerabilities in the Movie Production and Distribution Process*, Technical Report TD-5N6SJ4, AT&T Labs - Research, Florham Park, NJ, August, 2003.

Patrick McDaniel, *On Context in Authorization Policy*, Technical Report TD-5JCJCK, AT&T Labs - Research, Florham Park, NJ, January, 2003.

Patrick McDaniel and Atul Prakash, *An Architecture for Security Policy Enforcement*, Technical Report TD-5C6JFV, AT&T Labs - Research, Florham Park, NJ, July, 2002.

Patrick McDaniel and Atul Prakash, *Methods and Limitations of Security Policy Reconciliation*, Technical Report TD57-PAW, AT&T Labs - Research, Florham Park, NJ, February, 2002.

Patrick McDaniel and Atul Prakash, *Ismene: Provisioning and Policy Reconciliation in Secure Group Communication*, Technical Report CSE-TR-438-00, Electrical Engineering and Computer Science, University of Michigan, December, 2000.

Patrick McDaniel and Atul Prakash, *Lightweight Failure Detection in Secure Group Communication*, Technical Report CSE-TR-428-00, Electrical Engineering and Computer Science, University of Michigan, June, 2000.

Patrick McDaniel and Atul Prakash, *Antigone: Implementing Policy in Secure Group Communication*, Technical Report CSE-TR-426-00, Electrical Engineering and Computer Science, University of Michigan, May, 2000.

Patrick McDaniel and Sugih Jamin, *Windowed Certificate Revocation*, Technical Report CSE-TR-413-99, Electrical Engineering and Computer Science, University of Michigan, November, 1999.

Patrick McDaniel and Atul Prakash and Peter Honeyman, *Antigone: A Flexible Framework for Secure Group Communication*, Technical Report 99-2, Center for Information Technology Integration, September, 1999.

Patrick McDaniel and Avi Rubin, *A Response to "Can We Eliminate Certificate Revocation Lists?"*, Technical Report 99.8.1, AT&T Labs - Research, Florham Park, NJ, August, 1999.

Andrew Adamson and Charles J. Antonelli and Kevin Coffman and Patrick McDaniel and Jim Rees, *Secure Distributed Virtual Conferencing: Multicast or Bust*, Technical Report 99-1, Center for Information Technology Integration, January, 1999.

Patrick McDaniel and Sugih Jamin, *Windowed Key Revocation in Public Key Infrastructures*, Technical Report CSE-TR-376-98, Electrical Engineering and Computer Science, University of Michigan, , 1998.

Patrick McDaniel and Sugih Jamin, *A Scalable Key Distribution Hierarchy*, Technical Report CSE-TR-366-98, Electrical Engineering and Computer Science, University of Michigan, , 1998.

Patrick McDaniel and Peter Honeyman and Atul Prakash, *Lightweight Secure Group Communication*, Technical Report 98-2, Center for Information Technology Integration, University of Michigan, April, 1998.

Wayne Zage and Delores Zage and Patrick McDaniel and IrshadKhan, *Evaluating Design Metrics on Large-Scale Software*, Technical Report SERC-TR-106-P, Software Engineering Resource Center, PurdueUniversity, September, 1991.

PUBLIC SPEAKING

The Challenges of Machine Learning in Adversarial Settings: A Systems Perspective, University of California, Los Angeles, Los Angeles, CA, December, 2023.

The Security of AI (what organizations need to know), TEDxUWMadison, Madison, WI, November, 2023.

The Challenges of Machine Learning in Adversarial Settings: A Systems Perspective, Michigan State University, Lansing, MI, November, 2023.

The Security of AI (what organizations need to know), Greater Madison Chamber of Commerce, Madison, WI, September, 2023.

Security's Role in Achieving Sustainability, Center for Sustainability and the Global Environment, Madison, WI, February, 2023.

Security's Role in Achieving Sustainability, 29th ACM Conference on Computer and Communications Security (CCS), Los Angeles, CA, November, 2022.

Security, Game Theory, and Their Role in Achieving Sustainability, Conference on Decision and Game Theory for Security, Pittsburgh, PA, October, 2022.

NSF Funding: Why, What and How, School of Computer, Data & Information Sciences, UW-Madison, Madison, WI, October, 2022.

Prognosticating the Future of IoT Security, 2022 IEEE SafeThings Workshop, San Francisco, CA, May, 2022.

The Challenges of Machine Learning in Adversarial Settings: A Systems Perspective, Temple University, Philadelphia, PA, March, 2022.

The Challenges of Machine Learning in Adversarial Settings: A Systems Perspective, CACR Security Speaker Series, Indiana University, Online, August, 2021.

The Challenges of Machine Learning in Adversarial Settings: A Systems Perspective, Robustness of AI Systems to Adversarial Attacks (RAISA3), Online, August, 2020.

The Challenges of Machine Learning in Adversarial Settings: A Systems Perspective, Computer Science Department, University of Wisconsin-Madison, Madison, WI, February, 2020.

The Challenges of Machine Learning in Adversarial Settings, Computer Science Department, Stonybrook University, Stonybrook, NY, December, 2019.

The Challenges of Machine Learning in Adversarial Settings, Cylab Security and Privacy Institute, Carnegie Mellon University, Pittsburgh, PA, December, 2019.

The Challenges of Machine Learning in Adversarial Settings, S2ERC, Ball State University, Muncie, IN, November, 2019.

The Challenges of Machine Learning in Adversarial Settings, Triangle Area Privacy and Security Day, Durham, NC, October, 2019.

AI-Cybersecurity Workshop Briefing to the NITRD and MLAI Subcommittees, NITRD and MLAI Subcommittees Quarterly Meeting, Washington, DC, July, 2019.

Workshop on the Security and Privacy of Machine Learning, Workshop on the Security and Privacy of Machine Learning, International Conference on Machine Learning, Long Beach, CA, June, 2019.

The Challenges of Machine Learning in Adversarial Settings, 2019 Subversion and Assurance of AI Workshop, US National Reconnaissance Office, Washington, DC, March, 2019.

The Challenges of Machine Learning in Adversarial Settings, National Science Foundation, Alexandria, VA, March, 2019.

Convergence of AI and IoT, Intelligence Community Studies Board, Division on Engineering and Physical Sciences, The National Academy of Sciences/Engineering, Washington, DC, February, 2019.

Tracing the Arc of Smartphone Application Security, Duke University, Durham, NC, February, 2019.

The Challenges of Machine Learning in Adversarial Settings, Department of Computer Science, University at Buffalo, Buffalo, NY, November, 2018.

The Challenges of Machine Learning in Adversarial Settings, Department of Computer Science, Purdue University, West Lafayette, Indiana, November, 2018.

The Challenges of Machine Learning in Adversarial Settings, Computer Science Department, Indiana University of Pennsylvania, Indiana, PA, October, 2018.

The Challenges of Machine Learning in Adversarial Settings, Penn State University Alumni Association, University Park, PA, September, 2018.

The Challenges of Machine Learning in Adversarial Settings, Department of Software and Information Systems, University of North Carolina at Charlotte, Charlotte, NC, February, 2018.

Tracing the Arc of Smartphone Application Security, School of Electrical and Computer Engineering, Georgia Tech University, Atlanta, GA, December, 2017.

Tracing the Arc of Smartphone Application Security, Department of Electrical Engineering and Computer Science, Ohio University, Athens, OH, October, 2017.

Attacks, Defenses, and Impacts of Machine Learning in Adversarial Settings, 2017 Conference on Security and Privacy in Communication Networks (SecureComm), Niagara Falls, Canada, October, 2017.

Attacks, Defenses, and Impacts of Machine Learning in Adversarial Settings, Celebrating 50 Years of Computer Science @ NC State, North Carolina State University, Raleigh, NC, October, 2017.

Tracing the Arc of Smartphone Application Security, Computer Science Department and the Electrical and Computer Engineering Department Seminar Series, Colorado State University, Fort Collins, CO, October, 2017.

Tracing the Arc of Smartphone Application Security, Rochester Institute of Technology, College of Computing and Information Sciences, Rochester, NY, September, 2017.

Tracing the Arc of Smartphone Application Security, University of Texas-Dallas, Department of Computer Science, Dallas, TX, May, 2017.

Tracing the Arc of Smartphone Application Security, 2017 ACM on International Workshop on Security And Privacy Analytics, Scottsdale, AZ, March, 2017.

Tracing the Arc of Smartphone Application Security, The Ohio State University, Department of Computer Science and Engineering, Columbus, OH, March, 2017.

Tracing the Arc of Smartphone Application Security, University of California-Irvine, Computer Science Department, Irvine, CA, March, 2017.

Tracing the Arc of Smartphone Application Security, Virginia Technical University, Department of Computer Science, Blacksburg, VA, March, 2017.

Tracing the Arc of Smartphone Application Security, 12th International Conference on Information Systems Security , Jaipur, India, December, 2016.

Tracing the Arc of Smartphone Application Security, University of Michigan, Ann Arbor, Ann Arbor, MI, November, 2016.

Machine Intelligence in Adversarial Settings, Developing a Normative Framework for Cyberwarfare, United States Naval Academy, Annapolis, MD, September, 2016.

Eight Years of Mobile Smartphone Security, University of Pittsburgh, Pittsburgh, PA, September, 2016.

Eight Years of Mobile Smartphone Security, New Jersey Institute of Technology, Newark, NJ, September, 2016.

Setting a Cyber-Security Baseline for Physical Systems: Terminology, Technologies, and Goals, Pacific Northwest Clean Water Association, webinar, August, 2016.

The Limitations of Machine Learning in Adversarial Settings, 25th International Conference on Computer Communication and Networks (ICCCN 2016), Waikoloa, HI, August, 2016.

Learning from Ourselves: Where are we and where can we go in mobile systems security?, Mobile Security Technologies (MOST) 2016 Workshop, IEEE Computer Society Security and Privacy Workshops, San Jose, CA, May, 2016.

Eight Years of Mobile Smartphone Security, Center for Secure and Dependable Systems (CSDS) Cybersecurity Symposium, Coeur d'Alene, April, 2016.

Eight Years of Mobile Smartphone Security, University of Idaho, Moscow, ID, April, 2016.

Army Installation 2035: Cyber Challenges and Opportunities, US Department of Defense, Arlington, VA, April, 2016.

The Limitations of Machine Learning in Adversarial Settings, Florida Institute on National Security Assured Autonomy Workshop, Fort Myers, FL, February, 2016.

SABOT: Specification-based Payload Generation for Programmable Logic Controllers, Messaging, Malware and Mobile Anti-Abuse Working Group (M3AAWG) , San Francisco, CA, February, 2016.

Seven Years of Mobile Smartphone Security, Computer and Information Sciences Department, Temple University, Philadelphia, PA, January, 2016.

Seven Years of Mobile Smartphone Security, Massachusetts Institute of Technology–Lincoln Labs, Lexington, MA, January, 2016.

Six Years of Mobile Smartphone Security, Information Trust Institute, University of Illinois at Urbana-Champaign, Urbana-Champaign, IL, September, 2015.

The Importance of Measurement and Decision Making to a Science of Security, 2015 IEEE Conference on Communications and Network Security, Florence, Italy, September, 2015.

The Importance of Measurement and Decision Making to a Science of Security, 3rd International Symposium on Resilient Cyber Systems, Philadelphia, PA, August, 2015.

Six Years of Mobile Smartphone Security, CISPA Distinguished Lecture Series, Max Planck Institute/Saarland University, Saarbrücken Germany, July, 2015.

Six Years of Mobile Smartphone Security, Technische Universität Darmstadt, Darmstadt Germany, July, 2015.

Estimating Attack Intent and Mission Impact From Detection Signals, Workshop on Cyber Attack Detection, Forensics and Attribution for Assessment of Mission Impact, NATO Science and Technology Organization, Information Systems Technology Panel, Istanbul, Turkey, June, 2015.

The Importance of Measurement and Decision Making to a Science of Security, 2015 Symposium And Bootcamp on the Science of Security (Hotsos), University of Illinois at Urbana-Champaign, April, 2015.

Security and Science of Agility, First ACM Workshop on Moving Target Defense (MTD 2014), Scottsdale, AZ, November, 2014.

Evaluating Mobile Smartphone Security: The First Five Years, Computer Science Colloquium Series, Harvard School of Engineering and Applied Sciences, Harvard University, Boston, MA, October, 2014.

A Secondary Internet Revolution: How the Smart Device has Changed the Information Security Landscape, IEEE New Technology Industry Seminar (NTIS '13), Everett, WA, August, 2013.

Geotargeting: Mobile Device Privacy and Security, National Academy of Sciences, Washington DC, February, 2013.

Authentication and Web Security, Security and Privacy in IT-EMTM 604 Guest Lecture, University of Pennsylvania, Philadelphia, PA, February, 2013.

The Realities of Voting: A Retrospective of Ten Years of Information Security and Electronic Voting Systems, 2012 Information Assurance Day, Computer Science Department, Indiana University of Pennsylvania, Indiana, PA, November, 2012.

Permission-based Application Governance; A Step Forward or Backward?, 26th Annual WG 11.3 Conference on Data and Applications Security and Privacy (DBSec'12), Paris, France, July, 2012.

Evaluating Mobile Smartphone Security: The First Four Years, Carnegie Mellon University, Pittsburgh, PA, April, 2012.

Scalable Integrity-Guaranteed AJAX, The 14th Asia-Pacific Web Conference (APWeb), Kunming, China, April, 2012.

Evaluating Mobile Smartphone Application Security, Singapore Management University, Singapore, September, 2011.

Evaluating Mobile Smartphone Application Security, Computer Security Foundations Symposium, Florham Park, NJ, July, 2011.

Security Challenges and Solutions in Mobile Smartphone Applications, Computer Security Foundations Symposium, Domaine de l'Abbaye des Vaux de Cernay, France, June, 2011.

Security Challenges and Solutions in Mobile Smartphone Applications, Computer and Information Science Department, University of Oregon, Eugene, OR, April, 2011.

Identifying (and Addressing) Security and Privacy Issues in Smart Electric Meters, Center for Non-Linear Studies, Los Alamos, NM, February, 2011.

Security Challenges and Solutions in Mobile Smartphone Applications, Department of Software Information Systems College of Computing and Informatics, UNC Charlotte, Charlotte, NC, December, 2010.

Security Challenges and Solutions in Mobile Smartphone Applications, Computer Science Department, Indiana University of Pennsylvania, Indiana, PA, December, 2010.

Security Challenges and Solutions in Mobile Smartphone Applications, Computer Science Department, Georgetown University, Washington D.C., November, 2010.

Security Challenges and Solutions in Mobile Smartphone Applications, Networking and Security Research Center, Computer Science and Engineering, Pennsylvania State University, University Park, PA, October, 2010.

Security Challenges and Solutions in Mobile Smartphone Applications, Security Day Seminar, Penn State University, University Park, PA, October, 2010.

The Changing Vulnerability Landscape, Association for Computing Machinery, Penn State Student Chapter, University Park, PA, September, 2010.

The Changing Vulnerability Landscape, ExxonMobil, Falls Church, VA, March, 2010.

The Impact of Supply Chain on Information and Communications Technology Security, The 1st Workshop on Telecommunications Infrastructure Protection and Security, Honolulu, HI, December, 2000.

Energy Theft in the Advanced Metering Infrastructure, Networking and Security Research Center, Computer Science and Engineering, Pennsylvania State University, State College, PA, October, 2009.

Secure Provenance in High-End Computing Systems, NSF HECURA FSIO PI Meeting, Arlington, VA, August, 2009.

Missing Glue: Architectural Support for Security Annotations, National Science Foundation Security Driven Architecture Workshop, Arlington, VA, July, 2009.

Scalable Integrity-Justified Content Provenance, NSERC ISSNet Workshop, Ottawa, Canada, June, 2009.

Utility Grid Automation and Risk Management, Clean Technology Conference and Expo, Houston, Texas, May, 2009.

Scalable Integrity-Justified Content Provenance, Center for Applied Cybersecurity Research, Indiana University, Bloomington, IN, April, 2009.

Scalable Integrity-Justified Content Provenance, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, April, 2009.

What is Security, Dickenson Law School, Penn State University, State College, PA, April, 2009.

Scalable Integrity-Justified Content Provenance, Department of Computer Science and Engineering, Notre Dame University, South Bend, IN, April, 2009.

Electronic Voting: The Good, the Bad, and the Reality, Software Engineering Research Center Showcase, Muncie, IN, November, 2008.

Ohio Voting Systems Integrity: The EVEREST Report, Networking and Security Research Center, Computer Science and Engineering, Pennsylvania State University, State College, PA, October, 2008.

Data Provenance: Challenges and Technology, Cyber Physical System Security Forum, Cyber Security Research and Development Review, Washington DC, October, 2008.

System-Wide Information Flow Enforcement, NICIAR PI Meeting, Washington DC, September, 2008.

Presto: Configuration Management at Massive Scale, NSF Workshop on Assurable and Usable Security Configuration, George Mason University, Fairfax, VA, August, 2008.

Asymmetry in Performance and Security Requirements for I/O in High-end Computing, NSF HECURA FSIO PI Meeting, Arlington, VA, August, 2008.

Authentication and Web Security, Security and Privacy in IT-EMTM 604 Guest Lecture, University of Pennsylvania, Philadelphia, PA, May, 2008.

SPAM and SPAM Mitigation, Computer Science Department, St. Vincent's University, Latrobe, PA, April, 2008.

Phones, The Press, Research and Grad School ... or how to make trouble and have fun doing it, Computer Science Department, St. Vincent's University, Latrobe, PA, April, 2008.

Applications and Services in Telecommunications Networks, NSF Wireless Security Workshop, Georgia Institute of Technology, Atlanta, GA, March, 2008.

Vulnerabilities and Opportunities in SMS-Capable Cellular Networks, Computer Science Department, Carleton University, Ottawa, Canada, March, 2008.

Ohio Voting Systems Integrity: The EVEREST Report, Case-Western Reserve University, Cleveland, OH, February, 2007.

Ohio Voting Systems Integrity: The EVEREST Report, Ohio State University, Columbus, OH, February, 2007.

Ohio Voting Systems Integrity: The EVEREST Report, Ohio University, Athens, OH, February, 2007.

Ohio Voting Systems Integrity: The EVEREST Report, Miami University, Ohio, Oxford, OH, February, 2007.

Ohio Voting Systems Integrity: The EVEREST Report, Bowling Green State University, Bowling Green, OH, February, 2007.

Vulnerabilities and Opportunities in SMS-Capable Cellular Networks, Computer Science Department, Indiana University of Pennsylvania, Indiana, PA, September, 2007.

Asymmetry in Performance and Security Requirements for I/O in High-End Computing, HECIWG FSIO 2007 Workshop, NSF, Arlington, VA, August, 2007.

Toward Valley-Free Interdomain Routing, IEEE International Conference on Communications (ICC) 2007, Glasgow, Scotland, June, 2007.

Extending Developer Tools for Security-Typed Languages, Software Engineering Research Center Fall Showcase, West Lafayette, IN, June, 2007.

Open Functionality in SMS/Cellular Networks, Computer and Information Science, University of Oregon, Eugene, OR, May, 2007.

Open Functionality in SMS/Cellular Networks, Computer Security Symposium, St. Cloud State University, St. Cloud, MN, May, 2007.

Authentication and Web Security, Security and Privacy in IT-EMTM 604 Guest Lecture, University of Pennsylvania, Philadelphia, PA, April, 2007.

Grains of SANs: Building Storage Area Networks from Memory Spots, CISCO Remote Faculty Seminar, University Park, PA, April, 2007.

Grains of SANs: Building Storage Area Networks from Memory Spots, 2007 IEEE Security and Privacy Crystal Ball Workshop, Hawthorne, NY, January, 2007.

Password Exhaustion: Predicting the End of Password Usefulness, 2nd International Conference on Information Systems Security , Kolkata, India, December, 2006.

Privacy Preserving Web-based Email, 2nd International Conference on Information Systems Security, Kolkata, India, December, 2006.

Physical and Digital Convergence: Where the Internet is the Enemy, Eighth International Conference on Information and Communications Security (ICICS '06), Raleigh, NC, December, 2006.

Extending Developer Tools for Security-Typed Languages, Software Engineering Research Center Fall Showcase, Muncie, IN, November, 2006.

Open Functionality in SMS/Cellular Networks, Johns Hopkins University, Computer Science Department, Baltimore, MD, September, 2006.

Open Functionality in SMS/Cellular Networks, George Mason University, Computer Science Department, Fairfax, VA, September, 2006.

Exploiting Open Functionality in SMS-Capable Cellular Networks, Motorola Security Symposium, Itasca, IL, September, 2006.

IsleB: Testing Large Scale BGP Security in Replayable Network Environments, NSF/DETER Community Workshop, Arlington, VA, June, 2006.

BGPRV: A Library for Fast and Efficient Routing Data Manipulation, NSF/DETER Community Workshop, Arlington, VA, June, 2006.

JifClipse: Extending Developer Tools for Security-Typed Languages, Software Engineering Research Center Spring Showcase, Schaumburg, IL, June, 2006.

Trends in Security: Critical Engineering in the Large, Schlumberger InnovateIT! 2006, Cambridge, MA, May, 2006.

Information Flow Revisited: Software Engineering to Provable Security, Network Center of Excellence, Motorola Labs, Schaumburg, IL, May, 2006.

Authentication and Web Security, Security and Privacy in IT-EMTM 604 Guest Lecture, University of Pennsylvania, Philadelphia, PA, April, 2006.

Exploiting Open Functionality in SMS-Capable Cellular Networks, InfraGard Pittsburgh Chapter General Meeting, Pittsburgh, PA, March, 2006.

Exploiting Open Functionality in SMS-Capable Cellular Networks, Computer Science Department, University of Virginia, Charlottesville, VA, January, 2006.

Software Engineering Tools for Security-Typed Languages: Using Eclipse to Make Secure Programming Practical, Software Engineering Research Center Showcase, Muncie, IN, November, 2005.

Exploiting Open Functionality in SMS-Capable Cellular Networks, AT&T IP Services Security Council, Middletown, NJ, October, 2005.

Exploiting Open Functionality in SMS-Capable Cellular Networks, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA, October, 2005.

Exploiting Open Functionality in SMS-Capable Cellular Networks, Computer Science Department, Yale University, New Haven, CT, October, 2005.

Exploiting Open Functionality in SMS-Capable Cellular Networks, Computer Science Department, SUNY-Stony Brook, Stony Brook, NY, October, 2005.

Exploiting Open Functionality in SMS-Capable Cellular Networks, Networking and Security Research Center, Computer Science and Engineering, Pennsylvania State University, State College, PA, October, 2005.

IsleB: Trace Driven Modeling of Internet-Scale BGP Attacks and Countermeasures, 2nd Annual DETER/EMIST Workshop, Newport Beach, CA, September, 2005.

Critical Infrastructure Security through Provably Secure Network Mediation, 2nd Japan/US Workshop on Critical Information Infrastructure Protection (CIIP), Tokyo, Japan, June, 2005.

Extending Developer Tools for Security-typed Languages, Software Engineering Research Center Showcase, West Lafayette, IN, June, 2005.

Origin Authentication in Interdomain Routing, 2005 IEEE Communications Quality and Reliability (CQR) International Workshop, St. Petersburg, FL, April, 2005.

Analysis of Security Vulnerabilities in the Movie Production and Distribution Process, Messiah College, Senior Seminar Series, Grantham, PA, April, 2005.

Origin Authentication in Interdomain Routing, Intel Research, Folsom CA, April, 2005.

Key Distribution Strategies For Low-Power Wireless Networks, Network Center of Excellence, Motorola Labs, Schaumburg, IL, April, 2005.

Policy Evolution: Autonomic Environmental Security, Software Engineering Research Center Showcase, Muncie, IN, December, 2004.

Information Assurance for Enterprise Networks, BAE Systems, Networking Seminar, Reston, VA, November, 2004.

Origin Authentication in Interdomain Routing, Computer Science Department, Purdue University, West Lafayette, IN, October, 2004.

Origin Authentication in Interdomain Routing, Electrical Engineering and Computer Science Department, University of Michigan, Ann Arbor, MI, October, 2004.

Origin Authentication in Interdomain Routing, Computer Science Department, University of Wisconsin, Madison, MD, September, 2004.

Analysis of Security Vulnerabilities in the Movie Production and Distribution Process, Computer Science and Engineering Student Organization, University Park, PA, September, 2004.

Useless Metaphors? Why Specifying Policy is So Hard, Workshop on Usable Privacy and Security Software, Center for Discrete Mathematics and Theoretical Computer Science (DIMACS), New Brunswick, New Jersey, July, 2004.

Analysis of Security Vulnerabilities in the Movie Production and Distribution Process, Information Systems Research Seminar, Stern School of Business, New York University, New York, NY, April, 2004.

Analysis of Security Vulnerabilities in the MovieProduction and Distribution Process, Ball State University, Muncie, IN, April, 2004.

Origin Authentication in Interdomain Routing, Computer Science Department, University of Illinois, Champaign, IL, March, 2004.

Origin Authentication in Interdomain Routing, Computer Science and Engineering Department, University of Minnesota, Minneapolis, MN, March, 2004.

Origin Authentication in Interdomain Routing, Computer Science Department, Johns Hopkins University, Baltimore, MD, March, 2004.

Origin Authentication in Interdomain Routing, Department of Computer Science, University of Massachusetts - Amherst, Amherst, MA, March, 2004.

Origin Authentication in Interdomain Routing, Computer Science and Engineering Department, PennState University, University Park, PA, March, 2004.

Origin Authentication in Interdomain Routing, School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, February, 2004.

Origin Authentication in Interdomain Routing, Computer Science Department, Northwestern University, Evanston, IL, February, 2004.

Origin Authentication in Interdomain Routing, Computer Science Department, SUNY-Stony Brook, Stony Brook, NY, February, 2004.

Attack Profiling and Simulation in Interdomain Routing, P2INGS Quarterly Meeting, Tempe, AZ, February, 2004.

Analysis of Security Vulnerabilities in the MovieProduction and Distribution Process, AT&T Finance Lunch, Morristown, NJ, January, 2004.

Origin Authentication in Interdomain Routing, AT&T IP Security Conference, Middletown, NJ, November, 2003.

Origin Authentication in Interdomain Routing, 10th ACM Conference on Computer and Communications Security (CCS), Washington, DC, October, 2003.

Origin Authentication in Interdomain Routing, Computer Science Department, Stevens Institute of Technology, Hoboken, NJ, October, 2003.

Origin Authentication in Interdomain Routing, Computer Science Department, Arizona State University, Mesa, AZ, September, 2003.

Analysis of Security Vulnerabilities in the MovieProduction and Distribution Process, 31st Technology Policy Research Conference (TPRC), Arlington, VA, September, 2003.

On Context in Authorization Policy, 8th ACM Symposium on Access Control Models and Technologies, Como, Italy, June, 2003.

The Antigone Project, DARPA Principal Investigator Meeting, San Antonio, TX, January, 2003.

Methods and Limitations of Security Policy Reconciliation, 2002 IEEE Symposium on Security and Privacy, Oakland, CA, May, 2002.

Policy Management in Distributed Systems, Cigital, Washington, DC, April, 2002.

Antigone: Policy Management in Secure Group Communication, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, April, 2001.

Antigone: Policy Management in Secure Group Communication, Computer Science Department, University of Wisconsin, Madison, MD, April, 2001.

Antigone: Policy Management in Secure Group Communication, Computer Science Department, University of Maryland, College Park, MD, April, 2001.

Antigone: Policy Management in Secure Group Communication, Computer Science Department, University of North Carolina, Chapel Hill, NC, April, 2001.

Antigone: Policy Management in Secure Group Communication, Computer Science Department, Johns Hopkins University, Baltimore, MD, March, 2001.

Antigone: Policy Management in Secure Group Communication, AT&T Shannon Laboratory, Florham Park, NJ, February, 2001.

Antigone: Policy Management in Secure Group Communication, Telcordia Applied Research Laboratory, Morristown, NJ, February, 2001.

Policy Problem Area 3 - Overview and Requirements, Internet Engineering Task Force MSEC BOF, San Diego, CA, December, 2000.

Multicast Security Policy Requirements and Building Blocks, Quarterly Secure Multicast Research Group Meeting (SMuG), San Diego, CA, December, 2000.

Antigone: Implementing Policy in Secure Multiparty Communication, Systems Design and Implementation (SDI) / Laboratory for Computer Systems (LCS) seminar series, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, November, 2000.

Secure Group Communication in Antigone 2.0, 11th Annual IPoCSE Research Symposium, Ann Arbor, MI, October, 2000.

Antigone Secure Group Communication, Bi-Annual DARPA Visit, Software System Research Laboratory, Ann Arbor, MI, September, 2000.

Problem Area 3: Policy, Quarterly Secure Multicast Research Group Meeting (SMuG), Pittsburgh, PA, July, 2000.

Windowed Certificate Revocation, IEEE INFOCOM 2000, Tel Aviv, Israel, March, 2000.

A Response to 'Can We Eliminate Certificate Revocation Lists?', Financial Cryptography 2000, Anguilla, British West Indies, February, 2000.

Multicast Security Policy Definition, Quarterly Secure Multicast Research Group Meeting (SMuG), Washington, DC, November, 1999.

Antigone: A Flexible Framework for Secure Group Communication, Quarterly Secure Multicast Research Group Meeting (SMuG), NAI Labs, Baltimore, MD, September, 1999.

Antigone: A Flexible Framework for Secure Group Communication, 10th Annual IPoCSE Research Symposium, Ann Arbor, MI, September, 1999.

Antigone: A Flexible Framework for Secure Group Communication, 8th USENIX Security Symposium, Washington, DC, August, 1999.

Antigone: A Flexible Framework for Secure Group Communication, IBM Watson Security Seminar, Westchester County, NY, July, 1999.

Windowed Revocation in Public Key Infrastructures, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, September, 1998.

Scalable Key Distribution Hierarchy, 9th Annual IPoCSE Research Symposium, Ann Arbor, MI, March, 1998.

JavaLauncher Applet Platform, NASA, Kennedy Space Center Security Seminar, Kennedy Space Center, FL, January, 1998.

Secure High Performance Group Communication, Directed StudyDefense, Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, September, 1997.