

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent of: Benjamin Duval et al.
U.S. Patent No.: 12,395,359 Attorney Docket No.: 56407-0020IP1
Issue Date: August 19, 2025
App. Serial No.: 18/026,925
Filing Date: September 17, 2021
Title: METHOD OF IMPLEMENTING A PHYSICAL
UNCLONABLE FUNCTION

DECLARATION OF DR. PATRICK TRAYNOR

TABLE OF CONTENTS

I.	INTRODUCTION	10
II.	QUALIFICATIONS	10
III.	LEGAL PRINCIPLES.....	14
	A. Claim Interpretation.....	14
	B. Anticipation	15
	C. Obviousness.....	15
IV.	LEVEL OF ORDINARY SKILL IN THE ART	17
V.	MATERIALS CONSIDERED AND RELIED UPON	18
VI.	OVERVIEW OF THE '359 PATENT	22
	A. Brief Description	22
	B. Prosecution History	25
	C. Claim Construction.....	27
	1. Limitations [1.3], [1.4], [6.3], [6.4], [12.3], [12.4], [14.3], [14.4].....	27
VII.	OVERVIEW OF CONCLUSIONS FORMED	32
VIII.	OVERVIEW OF THE CITED REFERENCES.....	34
	A. Wu	34
	B. Muelich.....	37
	C. Tuyls	38
	D. Yang.....	38
	E. Chan.....	39
	F. Sundararajan	39
IX.	APPLICATION OF THE PRIOR ART TO THE CLAIMS.....	40
	A. GROUND 1A—Claims 6-9, 11, and 14 are obvious in view of Wu and Muelich.....	40
	1. Combination of Wu and Muelich.....	40
	2. Analysis	52
	B. GROUND 1B—Claim 10 is obvious in view of Wu, Muelich, and Tuyls	
	84	
	1. Combination of Wu, Muelich, and Tuyls.....	84

2. Analysis	88
C. GROUND 1C—Claims 1-3, 5, 12, and 13 are obvious in view of Wu, Muelich, and Yang	89
1. Combination of Wu, Muelich, and Yang	89
2. Analysis	93
D. GROUND 1D—Claim 4 is obvious in view of Wu, Muelich, Yang, and Tuyls	96
1. Combination of Wu, Muelich, Yang, Tuyls.....	96
2. Analysis	97
E. GROUND 2A—Claims 6-9, 11, and 14 are obvious in view of Wu, Muelich, and Chan.....	97
1. Combination of Wu, Muelich, and Chan	97
2. Analysis	99
F. GROUND 2B— Claim 10 is obvious in view of Wu, Muelich, Chan, Tuyls	102
1. Combination of Wu, Muelich, Chan, and Tuyls	102
2. Analysis	102
G. GROUND 2C—Claims 1-3, 5, 12, and 13 are obvious in view of Wu, Muelich, Chan, and Sundararajan	102
1. Combination of Wu, Muelich, Chan, and Sundararajan	102
2. Analysis	104
H. GROUND 2D—Claim 4 of the '359 patent is obvious in view of Wu, Muelich, Chan, Sundararajan, and Tuyls	107
1. Combination of Wu, Muelich, Chan, Sundararajan, and Tuyls.....	107
2. Analysis	107
I. GROUND 3—Claims 1-14 of the '359 patent would lack written description support if interpreted to avoid the prior art	107
1. Legal Principles	107
2. Review of the '359 patent	108
3. Analysis	109
J. GROUND 4— Claims 1-14 of the '359 patent would be indefinite if interpreted to avoid the prior art.....	111

1. Legal Principles.....	111
2. Review of the '359 patent	111
3. Analysis	112
X. ADDITIONAL REMARKS.....	114

LISTING OF CLAIMS

Claim 1 ¹	
[1.pre]	A method for repairing response bits of a physical unclonable function of an electronic device, said physical unclonable function outputting said response bits and additional output bits, said method comprising the following steps:
[1.1]	generating a PUF response from said response bits (S1),
[1.2]	detecting an error in the PUF response (S2),
[1.3]	determining erroneous response bits of the PUF response (S3) comprising replacing one by one, in a random order, each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is [] corrected,
[1.4]	determining a match (S4) between each determined erroneous response bit and a selected additional output bit such that replacing the erroneous response bits by the matching additional output bits corrects the error in the PUF response,
[1.5]	storing said determined match in a repair list (S5),
[1.6]	replacing (S6) in the PUF response said erroneous response bits with the matching additional output bits in the repair list.
Claim 2	
[2.1]	The method according to claim 1, wherein said electronic device is configured for access to a memory storing said repair list comprising at least one match between a response bit and an additional output bit,

¹ Claim 1 issued with a typo in [1.3] that is corrected in this Petition for clarity using bracketed text. *See* OKTA-1001, 7:15-16.

[2.2]	the step of generation of a PUF response (S1) comprises replacing in the PUF response said matched response bits with the matching additional output bits in the repair list.
Claim 3	
[3]	The method of claim 1, wherein said additional output bit are bits among most reliable output bits of said physical unclonable function.
Claim 4	
[4.1]	The method of claim 1, wherein said device is configured for access to a memory storing a reference hash value of said response bits,
[4.2]	detecting an error in the PUF response (S2) comprises: generating a hash value by applying a hash function on said bits of the PUF response, comparing said hash value with said stored reference hash value.
Claim 5	
[5]	The method of claim 3, wherein detecting an error in the PUF response (S2) comprises using an error correcting code.
Claim 6²	
[6.pre]	A method for repairing response bits of a physical unclonable function of an electronic device, said physical unclonable function outputting said response bits and additional output bits, said method comprising the following steps:
[6.1]	generating a PUF response from said response bits (S1),
[6.2]	detecting an error in the PUF response (S2),

² Claim 6 issued with typos in [6.3] and [6.6] that are corrected in this Petition for clarity using bracketed text. *See* OKTA-1001, 7:55-56, 7:65-66.

[6.3]	determining erroneous response bits of the PUF response (S3) comprising replacing one by one, in a [predetermined] order, each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is corrected,
[6.4]	determining a match (S4) between each determined erroneous bit and a selected additional output bit such that replacing the erroneous response bits by the matching additional output bits corrects the error in the PUF response,
[6.5]	storing said determined match in a repair list (S5),
[6.6]	replacing ([S6]) in the PUF response said erroneous response bits with the matching additional output bits in the repair list.
Claim 7	
[7.1]	The method according to claim 6, wherein said device being configured for having access to a memory storing a preventive toggling list of the least reliable response bits of said physical unclonable function,
[7.2]	determining erroneous response bits of the PUF response to be repaired (S3) comprises replacing first bits of the Preventive toggling list before replacing other bits of the PUF response.
Claim 8	
[8.1]	The method according to claim 6, wherein said electronic device is configured for access to a memory storing said repair list comprising at least one match between a response bit and an additional output bit,
[8.2]	the step of generation of a PUF response (S1) comprises replacing in the PUF response said matched response bits with the matching additional output bits in the repair list.
Claim 9	
[9]	The method of claim 6, wherein said additional output bit are bits among most reliable output bits of said physical unclonable function

Claim 10	
[10.1]	The method of claim 6, wherein said device is configured for access to a memory storing a reference hash value of said response bits,
[10.2]	detecting an error in the PUF response (S2) comprises: generating a hash value by applying a hash function on said bits of the PUF response, comparing said hash value with said stored reference hash value.
Claim 11	
[11]	The method of claim 9, wherein detecting an error in the PUF response (S2) comprises using an error correcting code.
Claim 12	
[12.pre]	An electronic device comprising active electronic primitives, PUF circuits generating a PUF response by running a physical unclonable function outputting response bits and additional output bits, a processor and at least one memory configured to perform the steps of:
[12.1]	generating a PUF response from said response bits (S1),
[12.2]	detecting an error in the PUF response (S2),
[12.3]	determining erroneous response bits of the PUF response (S3) comprising replacing one by one, in a random order, each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is corrected,
[12.4]	determining a match (S4) between each determined erroneous response bit and a selected additional output bit such that replacing the erroneous response bits by the matching additional output bits corrects the error in the PUF response,
[12.5]	storing said determined match in a repair list (S5),
[12.6]	replacing (S6) in the PUF response said erroneous response bits with the matching additional output bits in the repair list.

Claim 13	
[13]	The electronic device of claim 12 wherein detecting an error in the PUF response (S2) comprises using an error correcting code.
Claim 14	
[14.pre]	An electronic device comprising active electronic primitives, PUF circuits generating a PUF response by running a physical unclonable function outputting response bits and additional output bits, a processor and at least one memory configured to perform the steps of:
[14.1]	generating a PUF response from said response bits (S1),
[14.2]	detecting an error in the PUF response (S2),
[14.3]	determining erroneous response bits of the PUF response (S3) comprising replacing one by one, in a predetermined order, each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is corrected,
[14.4]	determining a match (S4) between each determined erroneous response bit and a selected additional output bit such that replacing the erroneous response bits by the matching additional output bits corrects the error in the PUF response,
[14.5]	storing said determined match in a repair list (S5),
[14.6]	replacing (S6) in the PUF response said erroneous response bits with the matching additional output bits in the repair list.

I, Patrick Gerard Traynor, declare as follows:

I. INTRODUCTION

1. I have been retained as a technical expert by counsel on behalf of Okta, Inc. (“Petitioner”). I understand that Petitioner is requesting that the Patent Trial and Appeal Board institute a Post Grant Review (“PGR”) proceeding with respect to U.S. Patent No. 12,395,359 (“the ’359 patent”) (OKTA-1001).

2. I have been asked to provide my independent analysis of the ’359 patent in light of certain prior art publications, and I have done so based on my personal knowledge and experience.

3. I am not, and never have been, an employee of Petitioner. I received no compensation for this Declaration beyond my normal hourly compensation based on my time actually spent analyzing the ’359 patent, the prior art publications cited below, and various issues related thereto. I will not receive any added compensation based on the outcome of any PGR or other proceeding involving the ’359 patent.

II. QUALIFICATIONS

4. In formulating my opinions, I have relied on my knowledge, training, and experience in the relevant field. My education and experience are described more fully in my curriculum vitae (OKTA-1003). For ease of reference, I have highlighted certain information below.

5. I earned a B.S. in Computer Science from the University of Richmond

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

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

7. Prior to joining the University of Florida, I was an Associate Professor from March to August 2014 and an Assistant Professor of Computer Science from 2008 to March 2014 at the Georgia Institute of Technology. I have supervised many Ph.D., M.S., and undergraduate students during the course of my career.

8. My area of expertise is security, especially as it applies to mobile systems and networks, including cellular networks. As such, I regularly teach students taking my courses and participating in my research group to program and evaluate software and architectures for mobile and cellular systems. I have taught courses on the topics of network and systems security, cellular networks, and mobile systems at both Georgia Tech and the University of Florida. I also advised and

instructed the Information Assurance Officer Training Program for the United States Army Signal Corps in the Spring of 2010.

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

10. I have published over 100 articles in top conferences and journals in the areas of information security, mobile systems, and networking. Many of my results are highly cited, and I have received multiple "Best Paper" Awards. I have also written a book entitled "Security for Telecommunications Networks", which is used in wireless and cellular security courses at a number of top universities.

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

12. I have been the Program Chair for eight conferences and workshops, and have served as a member of the Program Committee for over 50 different conferences and workshops. I also previously served as an Associate Editor for

IEEE Security and Privacy Magazine and also as the Security Subcommittee Chair for the ACM US Technology Policy Committee (USACM).

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

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

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

16. I am a named inventor on more than ten US patents. These patents detail methods for determining the origin and path taken by phone calls as they traverse various networks, cryptographically authenticating phone calls, providing a secure means of indoor localization using mobile/wireless devices, detecting credit

card skimmers, identifying cloned credit cards, and blocking ransomware from encrypting data.

17. My curriculum vitae, included with this declaration as OKTA-1003, 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.

III. LEGAL PRINCIPLES

18. In forming my analysis and conclusions expressed in this Declaration, I have applied the legal principles described in the following paragraphs, which were provided to me by counsel for the Petitioner.

A. Claim Interpretation

19. I understand that, for purposes of my analysis in this Post Grant Review proceeding, the terms appearing in the patent claims should be interpreted according to their “ordinary and customary meaning.” In determining the ordinary and customary meaning, I understand that the words of a claim are first given their plain meaning that those words would have had to a person of ordinary skill in the art at the time of the alleged invention. I also understand that the structure of the claims, the specification, and file history may be used to better understanding the meaning of a claim. Moreover, I understand that even treatises and dictionaries may be used, albeit under limited circumstances, to determine the meaning attributed by a person

of ordinary skill in the art to a claim term.

B. Anticipation

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

C. Obviousness

21. I have been informed that a patent claim is invalid as “obvious” in light of one or more prior art references if it would have been obvious to a person of ordinary skill in the art at the time of the alleged invention (“POSITA”), taking into account (1) the scope and content of the prior art, (2) the differences between the prior art and the claims, (3) the level of ordinary skill in the art, and (4) any so called “secondary considerations” of non-obviousness (e.g., evidence of “long felt need” for the claimed invention, commercial success attributable to the claimed invention, unexpected results of the claimed invention, and “copying” of the claimed invention by others).

22. While I do not know the exact date that the alleged invention of the ’359 patent was made, I do know that the ’359 patent’s earliest claimed priority date is September 18, 2020. For purposes of my obviousness analysis in the Grounds

below, I have applied a date of September 18, 2020 as the date of the alleged invention (the “Critical Date”), although my opinions expressed below would hold true even if the date of the alleged invention were earlier, such as any time in the late 2010s.

23. I have been informed that a claim can be obvious in light of a single prior art reference or multiple prior art references. To be obvious in light of a single prior art reference or multiple prior art references, there must be a reason that would have prompted a POSITA to modify or supplement the single prior art reference, or to combine two or more references, in a manner that provides the elements of the claimed invention. This reason may come from a teaching, suggestion, or motivation to combine, or may come from the reference(s) themselves, the knowledge or “common sense” of a POSITA, or from the nature of the problem to be solved, and this reason may be explicit or implicit from the prior art as a whole. I have been informed that, under the law, the combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results. I also understand it is improper to rely on hindsight in making the obviousness determination.

24. I understand that an obviousness determination also requires that a POSITA would have had a reasonable expectation of success. This concept has been explained to me as relating to the relative likelihood or predictability—from the

perspective of a POSITA—of successfully modifying the prior art in a manner that would meet the claimed limitations of the patent being challenged (here the '359 patent). I understand that the expectation of success need only be “reasonable” and, thus, does not require the absolute certainty gleaned from physically creating the proposed prior art modification.

IV. LEVEL OF ORDINARY SKILL IN THE ART

25. In providing the opinions set forth in this declaration, I was asked to consider the patent claims and the prior art through the eyes of a person of ordinary skill in the art as of the effective filing date of the '359 patent (“POSITA”), which I have concluded is September 18, 2020 for the reasons that I address in detail above (§III.C). I understand that the factors considered in determining the ordinary level of skill in a field of art include the level of education and experience of persons working in the field; the types of problems encountered in the field; the teachings of the prior art; and the sophistication of the technology. I understand that a POSITA is not a specific real individual, but rather is a hypothetical individual having the qualities reflected by the factors above. I understand that a POSITA also would have knowledge from the teachings of the prior art, including the art cited below.

26. Taking these factors into consideration, on or before September 18, 2020, a POSITA relating to the technology of the '359 patent would have had (1) a bachelor's degree in computer engineering, electrical engineering, or a related field,

and (2) two years of industry experience in designing cryptographic or security systems in electronic devices. Additional education could substitute for professional experience, and *vice versa*. *Id.* Physically unclonable functions (PUFs) were well known around the time of the Critical Date, and the '359 patent's claimed techniques include only routine security and cryptographic techniques that a POSITA would have been exposed to during their undergraduate education. Above, §II. For instance, I believe the concepts described by the '359 patent would have first been introduced to a POSITA in an undergraduate networking course that includes discussion of authentication, authorization, or accounting techniques (also referred to as "AAA"). I have taught such courses previously.

27. By September 18, 2020, my education and research experience was at least that of a POSITA. I am qualified to provide opinions concerning what a POSITA would have known and understood at that time, and my analysis and conclusions herein are from the perspective of a POSITA as of that date.

V. MATERIALS CONSIDERED AND RELIED UPON

28. In forming my opinions and reaching the conclusions described in this declaration, I have relied on the documents and materials cited herein as well as those identified in this declaration, including the '359 patent, the prosecution history of the '359 patent, and prior art references cited herein. These materials comprise patents, related documents, and printed publications. Each of these materials is a

type of document that experts in my field would have reasonably relied upon when forming their opinions. I have also relied on my education, training, research, knowledge, and personal and professional experience in the relevant technologies and systems that were already in use prior to, and within the timeframe of the earliest effective filing date of the '359 patent (September 18, 2020).

29. Accordingly, as part of my independent analysis of the '359 Patent, I have analyzed the following publications and materials:

Exhibit	Description
OKTA-1001	U.S. Patent No. 12,395,359 to Duval et al. (“the '359 Patent”)
OKTA-1002	Prosecution History of the '359 patent (“the Prosecution History”)
OKTA-1004	U.S. Patent Application No. 2015/0188717 (“Wu”)
OKTA-1005	S. Muelich, & M. Bossert, “A New Error Correction Scheme for Physical Unclonable Functions,” SCC 2017; 11th International ITG Conference on Systems, Communications and Coding. VDE, 2017 (“Muelich”)
OKTA-1006	U.S. Patent Application No. 2021/0051010 (“Yang”)
OKTA-1007	U.S. Patent Application No. 2012/0072737 (“Schrijen”)
OKTA-1008	A. Aghaie & A. Moradi, “TI-PUF: Toward Side-Channel Resistant Physical Unclonable Functions,” published April 2020, IEEE Transactions On Information Forensics And Security (“Aghaie”)
OKTA-1009	U.S. Patent Application No. 2019/0147967 (“Tuyls”)

Exhibit	Description
OKTA-1010	G. Sundararajan, C. Winstead, & E. Boutillon, “Noisy Gradient Descent Bit-Flip Decoding for LDPC Codes,” published 2014, IEEE Transactions on Communications (“Sundararajan”)
OKTA-1011	A. Chan & F. Kschischang, “A Simple Taboo-Based Soft-Decision Decoding Algorithm for Expander Codes,” published 1998, IEEE Communications Letters (“Chan”)
OKTA-1015	“A New Error Correction Scheme for Physical Unclonable Functions,” IEEE Xplore, Institute of Electrical and Electronics Engineers, https://ieeexplore.ieee.org/abstract/document/7938055 (accessed 8 Apr. 2026)
OKTA-1016	“A simple taboo-based soft-decision decoding algorithm for expander codes,” IEEE Xplore, Institute of Electrical and Electronics Engineers, https://ieeexplore.ieee.org/document/703905 (accessed 8 Apr. 2026)
OKTA-1017	“Noisy Gradient Descent Bit-Flip Decoding for LDPC Codes,” IEEE Xplore, Institute of Electrical and Electronics Engineers, https://ieeexplore.ieee.org/document/6894606 (accessed 8 Apr. 2026)
OKTA-1018	E. Stefanov, E. Shi, & D. Song, “Towards Practical Oblivious RAM,” <i>available at</i> https://arxiv.org/abs/1106.3652 (accessed 28 Apr. 2026) (“Stefanov”)
OKTA-1019	M. Yu & S. Devadas, “Secure and Robust Error Correction for Physical Unclonable Functions,” IEEE Computer Society, published 2010 (“Yu”)
OKTA-1020	U.S. Patent Application No. 2020/0059252 A1 (“Fackenthal”)

Exhibit	Description
OKTA-1021	U.S. Patent Application No. 2015/0381206 A1 (“Fainzilber”)
OKTA-1022	U.S. Patent Application No. 2017/0180140 A1 (“Mai”)
OKTA-1023	U.S. Patent Application No. 2013/0194886 A1 (“Schrijen-886”)
OKTA-1024	Liu et al., “Multi-Threshold Bit Flipping Algorithm for Decoding Structured LDPC Codes,” published Feb. 2015, IEEE Communications Letters (“Liu”)
OKTA-1025	U.S. Patent No. 10,749,694 (“Adams”)
OKTA-1026	U.S. Patent No. 10,574,467 (“Afghah”)
OKTA-1027	U.S. Patent No. 9,628,272 (“Rostami”)
OKTA-1028	Zhang et al., “Tabu-List Noisy Gradient Descent Bit Flipping Decoding of LDPC Codes,” published 2019, 11th International Conference on Wireless Communications and Signal Processing (WCSP) (“Zhang”)
OKTA-1029	Cui & Lin, “An Improved Gradient Descent Bit-Flipping Decoder for LDPC Codes,” published Aug. 2019, IEEE Transactions on Circuits and Systems (“Cui”)
OKTA-1031	T. Wadayama, K. Nakamura, M. Yagita, Y. Funahashi, S. Usmai, & I. Takumi, “Gradient Descent Bit Flipping Algorithms for Decoding LDPC Codes,” <i>available at</i> : https://arxiv.org/pdf/0711.0261 (“Wadayama”)
OKTA-1032	Pavithra G. & Naveen Kumar M., “BER Performance Comparison of Bit Flipping Algorithms used for Decoding of LDPC Codes,” <i>available at</i> : https://www.ijert.org/research/ber-performance-comparison-of-bit-flipping-algorithms-used-for-decoding-of-ldpc-codes-IJERTV4IS051311.pdf (“Pavithra”)

Exhibit	Description
OKTA-1033	U.S. Patent Application No. 2019/0305972 A1 (“Yao”)

30. My citations to non-patent literature throughout this Declaration reference the absolute page number added to the exhibit (as opposed to the original pagination of the document). For patent literature, I’ve used the column/line numbers or paragraph numbers. Unless I’ve indicated otherwise, all emphasis (bold/italics/underline) in any quoted text are ones I have added.

31. Although I cite and quote selected portions of various references in this Declaration, the reader should understand that these citations are representative examples. A POSITA would have viewed each reference in its entirety and in combination with other references. Accordingly, I intend the references identified in this Declaration to be viewed as incorporated in their entireties.

VI. OVERVIEW OF THE ’359 PATENT

A. Brief Description

32. The ’359 patent is directed to “an electronic device and a method thereof for repairing response bits of a physical unclonable function of an electronic device.” OKTA-1001, Abstract, 3:33-4:29. The ’359 patent explains that a “Physically Unclonable Function (PUF) is a function embodied in a physical device whose output depends on the microstructures of the device components generating it.” *Id.*, 1:14-58. PUF characteristics (also referred to as “primitives”) “depend on

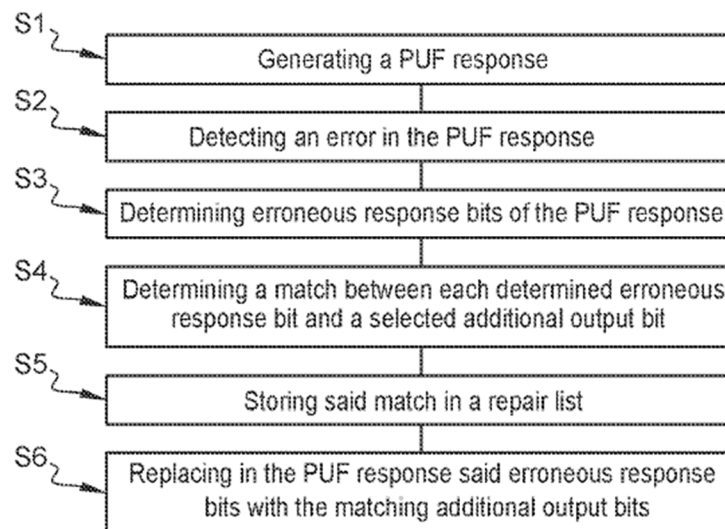
random process variations during manufacture that are uncontrolled and cannot be monitored.” *Id.* This makes a PUF “practically impossible to predict or duplicate.” *Id.*

33. Due to the “natural variations in the behavior of electronic components” the output of a PUF (the PUF “response”) “may not always be the same for a given challenge.” OKTA-1001, 1:14-58. In order to achieve a steady challenge-response relationship, the ’359 patent describes a need for “a method enabling to [*sic*] repair the response bits of a PUF when primitives forming the PUF response suffer from damage making the output bits unsteady or permanently modifying [*sic*] their value.” *Id.* The ’359 patent’s proposed solution involves replacing unreliable response bits with unused output bits of a comparable value (“additional output bits”). *Id.*, 3:33-4:29. “The main idea...is that when a PUF response bit is not stable anymore or was permanently damaged, it may be permanently replaced in the PUF response by another output bit providing the same value than the one the damaged response bit used to provide.” *Id.*, 4:11-16.

34. The process of replacing unreliable PUF response bits with more reliable additional output bits is accomplished by “replac[ing] one by one each bit of the PUF response by an arbitrary value and determin[ing] if the error in the PUF response is corrected.” OKTA-1001, 5:1-8. The ’359 patent explains that “[r]eplacing each bit may be performed in a random order in order to prevent an

attacker from deducing the index of erroneous bits from the time spent for identifying them.” *Id.*, 5:9-12. This process “determine[s] a match between each determined erroneous response bit and a selected additional output bit such that replacing the erroneous response bits by the matching additional output bits corrects the error in the PUF response.” *Id.*, 5:37-44. Once a match between an unreliable response bit and an additional output bit is made, “the determined match is stored” in a “repair list.” *Id.*, 5:52-64. An example process according to the ’359 patent is depicted below in FIG. 2.

Fig. 2



OKTA-1001, FIG. 2

35. The process of introducing randomization in electronic security systems was well established in the art prior to the Critical Date (as discussed more

in the grounds below). As an initial matter, concepts like “oblivious RAM” (oblivious Random Access Memory, or O-RAM) used randomization to obfuscate the addresses in a memory follow reads, thus frustrating an attacker’s ability to determine the content of the memory from observing memory accesses. OKTA-1018, 1-11. Thus, the application of randomization described in the ’359 patent just applies well known security techniques in the PUF space. *Id.*

B. Prosecution History

36. The ’359 patent issued on August 19, 2025 from U.S. Patent Application No. 18/026,925 (“the ’925 application”), which was filed on March 17, 2023 following an earlier international filing on September 17, 2021. OKTA-1002, 3-4. The prosecution of the ’925 application included a single Office Action (“the Action”) that presented a §103 rejection based on US 2019/0065734 (“Yao”) and US 2017/0310489 (“Van Der Sluis”). OKTA-1002, 81-94. Additionally, the Examiner indicated in the Action that certain dependent claims were allowable (then claims 7-9, each of which depended on then claims 3, 5, and 6). *Id.* The Examiner indicated that the reason for allowance was that the prior art of record allegedly lacked disclosure or suggestion of these claims. OKTA-1002, 90-92.

Allowable Subject Matter

6. Claims 7-9 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

OKTA-1002, 90.

Reasons for Allowance

7. The following is an examiner's statement of reasons for placing claims 7-9 under allowable subject matters:

The limitation of dependent claims 7-9 are allowed and the corresponding dependent claims including respective and any intervening claims are not disclosed by the any of the prior art of record. For example, the limitations in claim 9 recites "...wherein said device being configured for having access to a memory storing a Preventive toggling list of the least reliable response bits of said physical unclonable function, determining erroneous response bits of the PUF response to be repaired (S3) comprises replacing first bits of the Preventive toggling list before replacing other bits of the PUF response.." which are not are not taught or fairly suggested by the prior art of record.

OKTA-1002, 91

37. In response, the Applicant amended the claims to incorporate the allowable subject matter (claim 1 is reproduced below as an example, incorporating then claims 6 and 7). OKTA-1002, 55-63, 71-79. The Examiner then allowed the '925 application, noting Yao and Van Der Sluis as the closest art of record. OKTA-1002, 23-34.

Listing of amended Claims:

1. (CURRENTLY AMENDED) A method for repairing response bits of a physical unclonable function of an electronic device, said physical unclonable function outputting said response bits and additional output bits,
said method comprising the following steps:
generating a PUF response from said response bits (S1),
detecting an error in the PUF response (S2),
determining erroneous response bits of the PUF response (S3) comprising replacing one by one, in a random order, each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is corrected,
determining a match (S4) between each determined erroneous response bit and a selected additional output bit such that replacing the erroneous response bits by the matching additional output bits corrects the error in the PUF response,
storing said determined match in a repair list (S5),
replacing (S6) in the PUF response said erroneous response bits with the matching additional output bits in the repair list.

OKTA-1002, 56

C. Claim Construction

38. I have been informed by counsel and understand that all claim terms should be interpreted according to their ordinary and customary meaning in view of the specification. The discussion that follows addresses the ordinary and customary meaning of a subset of limitations addressed in Section III. I believe that no further claim terms need be expressly construed at this time, although my opinions may evolve in view of arguments from the Patent Owner.

1. Limitations [1.3], [1.4], [6.3], [6.4], [12.3], [12.4], [14.3], [14.4]

39. As discussed below in §IX.A.2, independent claims 1, 6, 12, and 14

recite:

determining erroneous response bits of the PUF response
(S3) comprising replacing one by one...each bit of the PUF
response by an arbitrary value and determining if the error in
the PUF response is corrected,

determining a match (S4) between each determined
erroneous bit and a selected additional output bit such that
replacing the erroneous response bits by the matching
additional output bits corrects the error in the PUF response

40. As described below, a POSITA would have understood (using the ordinary and customary meaning) that this process, based on how the process is described in the '359 patent specification, requires replacing individual bit values until such replacement corrects the PUF response error and, thus, reveals an erroneous bit. OKTA-1001, 6:29-63, FIG. 4. When an erroneous bit is discovered, it is matched with an additional output bit.

41. The above-recited limitations do not require determining multiple erroneous response bits in a single error-detection pass. In other words, the claimed process of “replacing one by one...each bit of the PUF response...and determining if the error in the PUF response is corrected” need not reveal more than one erroneous bit.

42. While the claim language references “erroneous response bits” (plural),

I understand that, for the purposes of claim interpretation, the plural form of a claim element does not necessarily imply that multiple elements are required by the claim. The surrounding claim language makes clear that bit values are replaced one at a time, i.e., “replacing one by one,” and this iterative process would reveal only a single erroneous bit. When this single erroneous bit is replaced, it would have corrected the PUF response. The specification confirms this interpretation by juxtaposing the claimed “replacing one by one” approach against an unclaimed alternative where “multiple bits may be replaced” to address a scenario where “multiple bits are erroneous in the PUF response.” OKTA-1001, 5:1-8.

43. The above-recited limitations also do not require replacing the value of each and every single bit of the PUF response in a single error-detection pass. Rather than exhaustive bit replacement, the claimed “replacing one by one” scheme is conducted in an iterative fashion until the erroneous bit is discovered by “determining if the error in the PUF response is corrected.” Once the erroneous bit is discovered, the claims do not require any further bit value replacements. If it were otherwise, the claimed configuration would be non-functional as it would either continually search for non-existent errors or needlessly replace additional correct bits. PUFs are characterized in part by their inability to be reproduced due to their reliance on unpredictable manufacturing variances – significant unneeded modifications to a PUF response would depart from the foundational basis behind

PUFs (i.e., the PUF response would no longer be based on the PUF primitives, but rather the modifications to them). As the prior art explains below, repairs to a PUF response are sometimes needed, but repairs would have only been limited to what was needed to make the PUF reliable.

44. This understanding is supported by the specification's narrative that the purpose of its disclosed techniques are to "minimize the computations required to find the erroneous bits of the PUF response" and to "reduc[e] the number of response bits to be replaced." OKTA-1001, 2:51-52, 3:1-3.³ Conducting needless bit replacement operations defies the '359 patent's express purpose. This makes sense because PUFs were suggested for use in mobile computing environments and other Internet of Things (IoT) resource constrained devices, where algorithmic efficiency was key. These processing efficiencies translated to battery savings and cost efficiency.

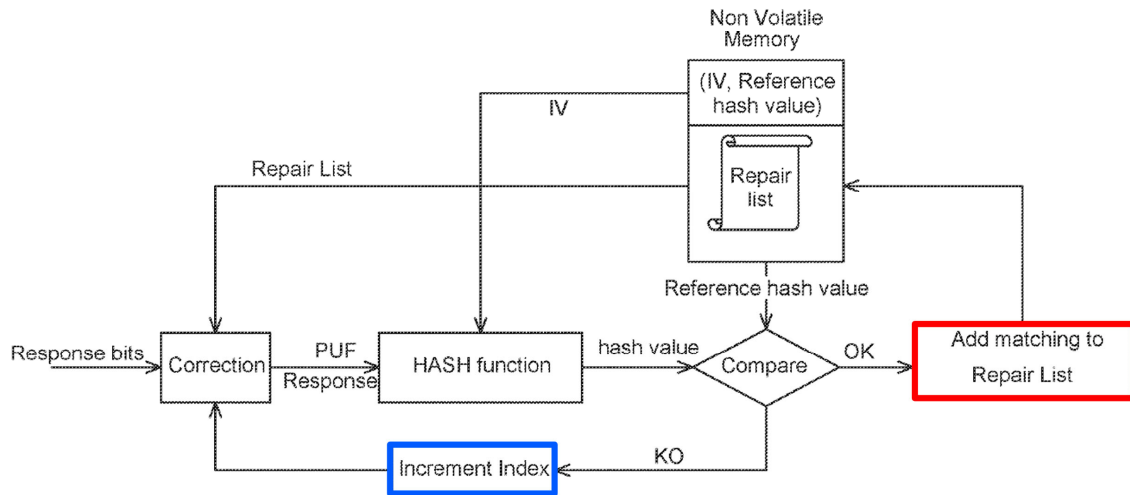
45. When describing the preferred embodiment of Figure 4, the '359 patent makes clear that the error-detection process concludes when the erroneous bit is identified and matched with a corresponding additional output bit. OKTA-1001, 6:29-36, FIG. 4. Said simply, a single erroneous bit is discovered and then replaced. *Id.* If it were otherwise, the specification would have expressly stated that the

³ See §II.A for a detailed overview of the Specification.

process of replacing bit values continues, even after identifying the erroneous bit, until every single bit of the PUF response has been replaced.

46. Figure 4 (below) also would have illustrated a sequential flow or operational link from the “**Add matching to Repair List**” step to the “**Increment Index**” step. An illustration like this would have indicated that the process continues iteratively, examining PUF response bits even after the erroneous bit is discovered and matched with a corrective additional output bit. But it does not exist. This makes sense, because doing so would be needlessly inefficient and waste processing power searching for subsequent errors that may not exist. The ’359 patent does not discuss such an arrangement because there would be little technical benefit from such an arrangement, which as described above, departs from the fundamental basis for PUFs (i.e., the response is based mostly on PUF primitives). Figure 4 shows the “**Add matching to Repair List**” step and the “**Increment Index**” step as distinct and separate branches from the “Compare” step. When the comparison indicates that the PUF response has been corrected, the repair list is updated but there is no need to continue incrementing the index because no further bit replacements occur.

Fig. 4



OKTA-1001, FIG. 4 (annotated)

47. The '359 patent's specification also does not describe an embodiment in which each and every bit of the PUF response is replaced. *See generally*, OKTA-1001. In my opinion, a POSITA would have read the claims in this way to require this undisclosed limitation. If the claims were read this way, the POSITA would have been unable to identify written description support for that interpretation (*below*, §IX.I) or understand the boundaries of claim scope with reasonable certainty (*infra*, §IX.J).

VII. OVERVIEW OF CONCLUSIONS FORMED

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

- (Ground 1A) Based upon my knowledge and experience, and my

review of the materials listed above, I believe that Claims 6-9, 11, and 14 of the '359 patent are obvious in view of Wu (OKTA-1004) and Muelich (OKTA-1005).

- (Ground 1B) Based upon my knowledge and experience, and my review of the materials listed above, I believe that Claim 10 of the '359 patent is obvious in view of Wu, Muelich, and Tuyls (OKTA-1009).
- (Ground 1C) Based upon my knowledge and experience, and my review of the materials listed above, I believe that Claims 1-3, 5, 12, and 13 of the '359 patent are obvious in view of Wu, Muelich, and Yang (OKTA-1006).
- (Ground 1D) Based upon my knowledge and experience, and my review of the materials listed above, I believe that Claim 4 of the '359 patent is obvious in view of Wu, Muelich, Yang, and Tuyls.
- (Ground 2A) Based upon my knowledge and experience, and my review of the materials listed above, I believe that Claims 6-9, 11, and 14 of the '359 patent are obvious in view of Wu, Muelich, and Chan (OKTA-1011).
- (Ground 2B) Based upon my knowledge and experience, and my review of the materials listed above, I believe that Claim 10 of the '359 patent is obvious in view of Wu, Muelich, Chan, Tuyls.

- (Ground 2C) Based upon my knowledge and experience, and my review of the materials listed above, I believe that Claims 1-3, 5, 12, and 13 of the '359 patent are obvious in view of Wu, Muelich, Chan, and Sundararajan (OKTA-1010).
- (Ground 2D) Based upon my knowledge and experience, and my review of the materials listed above, I believe that Claim 4 of the '359 patent is obvious in view of Wu, Muelich, Chan, Sundararajan, and Tuyls.
- (Ground 3) Based upon my knowledge and experience, and my review of the materials listed above, I believe that Claims 1-14 of the '359 patent would lack written description support if interpreted to avoid the prior art as discussed below.
- (Ground 4) Based upon my knowledge and experience, and my review of the materials listed above, I believe that Claims 1-14 of the '359 patent would be indefinite if interpreted to avoid the prior art as discussed below.

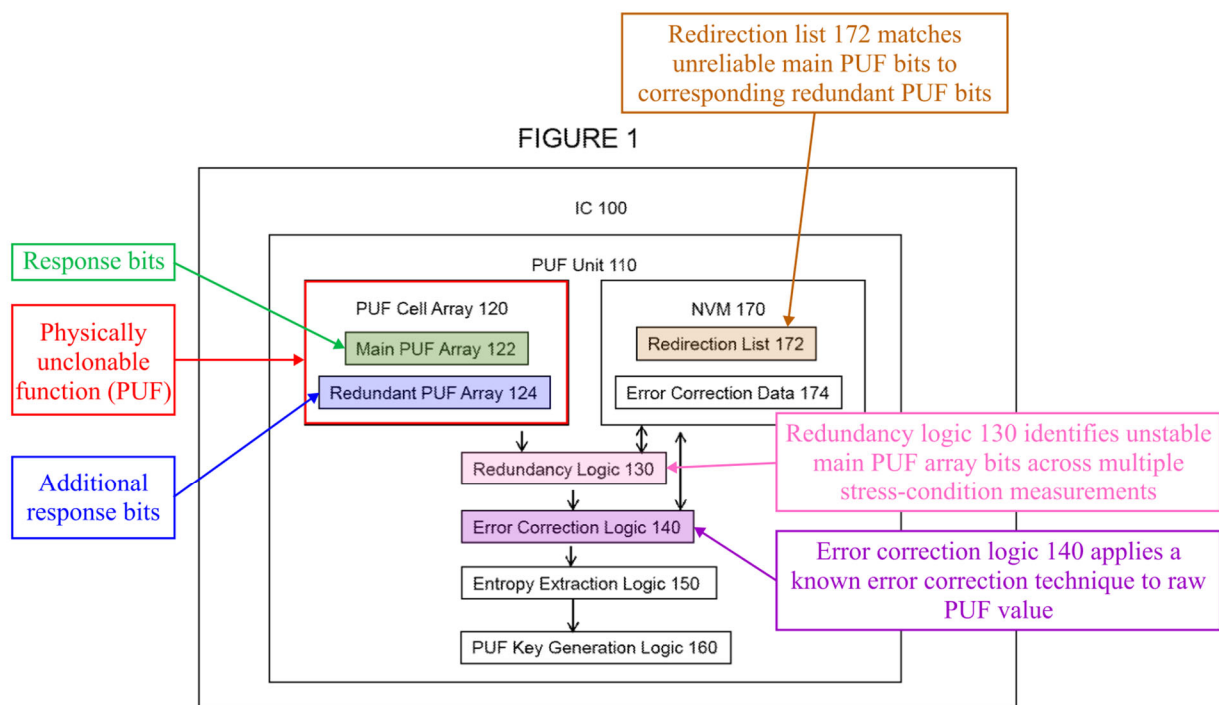
VIII. OVERVIEW OF THE CITED REFERENCES

A. Wu

49. Wu discloses techniques “providing for physically unclonable function redundant bits” in the context of electronic devices and integrated circuits. OKTA-

1004, [0010], [0015]-[0018]. Like the '359 patent, Wu teaches that a PUF typically includes redundant bits that can be used for “replacing bad bits in order to reduce the noise in and/or error rate of PUF key generation.” *Id.*, [0020]-[0025], FIG. 1.

50. For example, referencing Figure 1 (below), Wu describes a redirection list 172 on an integrated circuit 100, where each entry of the list corresponds to “one of the PUF cells in redundant PUF array 124” and indicates “a particular bit in main PUF array 122...that is to be replaced by the redundant PUF cell identified by that entry.” OKTA-1004, [0028]-[0030], FIG. 1.



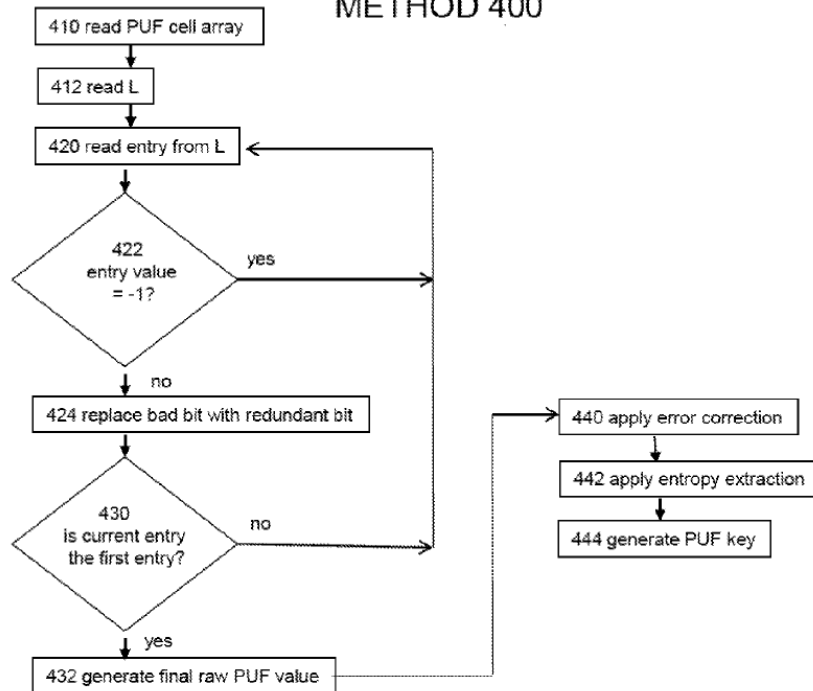
OKTA-1004, FIG. 1 (annotated)

51. Wu further describes how “redundancy logic 130” is configured to perform the testing, identification, and replacement of “bad” bits. OKTA-1004,

[0023]. Wu's testing and identification of bad bits can occur at different stages of the device's lifecycle. In one example, Wu teaches that the testing and bad-bit identification is performed "by the manufacturer or vendor," such that the device is provisioned with a populated redirection list before shipment. *See id.* (before sale or release"). In another example, Wu teaches that the testing and identification process can also occur "post-manufacturing (e.g., during customer use)" so as "to counter device aging effects." *Id.*, [0073].

52. Bad bit replacement by the redundancy logic 130 occurs "during the use of [integrated circuit] 100 (evaluation time), for example, by a customer," as shown in the exemplary workflow shown in the method 400 of Figure 4 (below). OKTA-1004, [0058].

FIGURE 4
METHOD 400



OKTA-1004, FIG. 4

53. As shown above, Wu’s exemplary workflow involves reading/evaluating the PUF cell array (step 410) and performing a list-driven replacement of identified “noisy” (bad) bits using redundant bits (step 424). OKTA-1004, FIG. 4, [0063]-[0064]. The “final raw PUF value” is then “processed by error correction logic 140” (step 440), which Wu says “may include any logic circuitry, or other hardware and/or firmware to provide for correcting errors in the generation of PUF keys.” *Id.*, [0064], [0025].

B. Muelich

54. Like the ’359 patent and Wu, Muelich acknowledges that error correction is an “indispensable component” to PUFs because “environmental

factors” inevitably cause variance in a PUF’s response over time. OKTA-1005, 1. Muelich describes improving error identification and correction in PUF-based key generators using iterative, bit-level correction during decoding. *Id.* Muelich’s “Bitflip Decoding” technique iteratively selects a bit position, flips that bit, and checks whether the decoding condition is satisfied. *Id.*, 4-5 (“We use a simple, iterative bitflip procedure[.]”).

C. Tuyls

55. Tuyls is directed to a “programming device arranged to obtain and store a random bit string in a memory device,” OKTA-1009, [0008]-[0021], FIG. 1. Tuyls discloses techniques for using a “hash over at least a part of the data in the multiple memory cells” for the purpose of identifying “faulty memory cells.” OKTA-1009, [0016], [0100]-[0114]. Specifically, Tuyls discloses an “authentication unit 162” that computes a “first hash” over a set of data. OKTA-1009, [0100]. Next, a “validation unit 164” compares the computed first hash with a “stored first hash” and determines if the two hash values match. OKTA-1009, [0101]-[0114]. If the hash values differ, it is determined that an error has occurred in the data. *Id.*

D. Yang

56. Yang discloses a memory including a “physically unclonable function (PUF) unit, a controller and a memory array” where the PUF is used to “provide a random bit pool.” OKTA-1006, [0004], [0010]-[0019], FIG. 1. Yang’s controller

“extract[s] a random bit sequence from the random bit pool in the PUF unit 105” for the purposes of generating a “true random number” (TRN). OKTA-1006, [0014]-[0015], FIG. 1. Yang discloses that these techniques allow “secure data storage, unique identity generation, true random number generation and secure key storage,” while “protecting data from unauthorized access and enhancing data security.” OKTA-1006, [0019].

E. Chan

57. Chan discloses a bit-flipping algorithm with a “taboo” function that “avoids repeated flipping of the same bit” by assigning a “taboo number” to each bit when it is flipped, which only allows subsequent flipping of that bit “if the current iteration number is equal to or less than the taboo number of that variable.” OKTA-1011, 1-3. Chan’s techniques are useful for avoiding “local minima” and result in “improved performance and improved decoding speed” for bit flipping algorithms, “while still maintaining ... simplicity.” *Id.*

F. Sundararajan

58. Sundararajan describes a modified “Gradient Descent Bit Flipping (GDBF) algorithm” for decoding LDPC bit-flipping codes. OKTA-1010, 1. Sundararajan’s “Noisy GDBF” introduces a “random perturbation” (vector “z” of “identically distributed Gaussian random variables”) that randomizes bit-flipping decisions made by the algorithm. OKTA-1010, 1-4. Introducing this randomness

“offers a low-complexity solution to escape spurious local maxima” that a bit-flipping algorithm would otherwise erroneously converge upon. *Id.*

IX. APPLICATION OF THE PRIOR ART TO THE CLAIMS

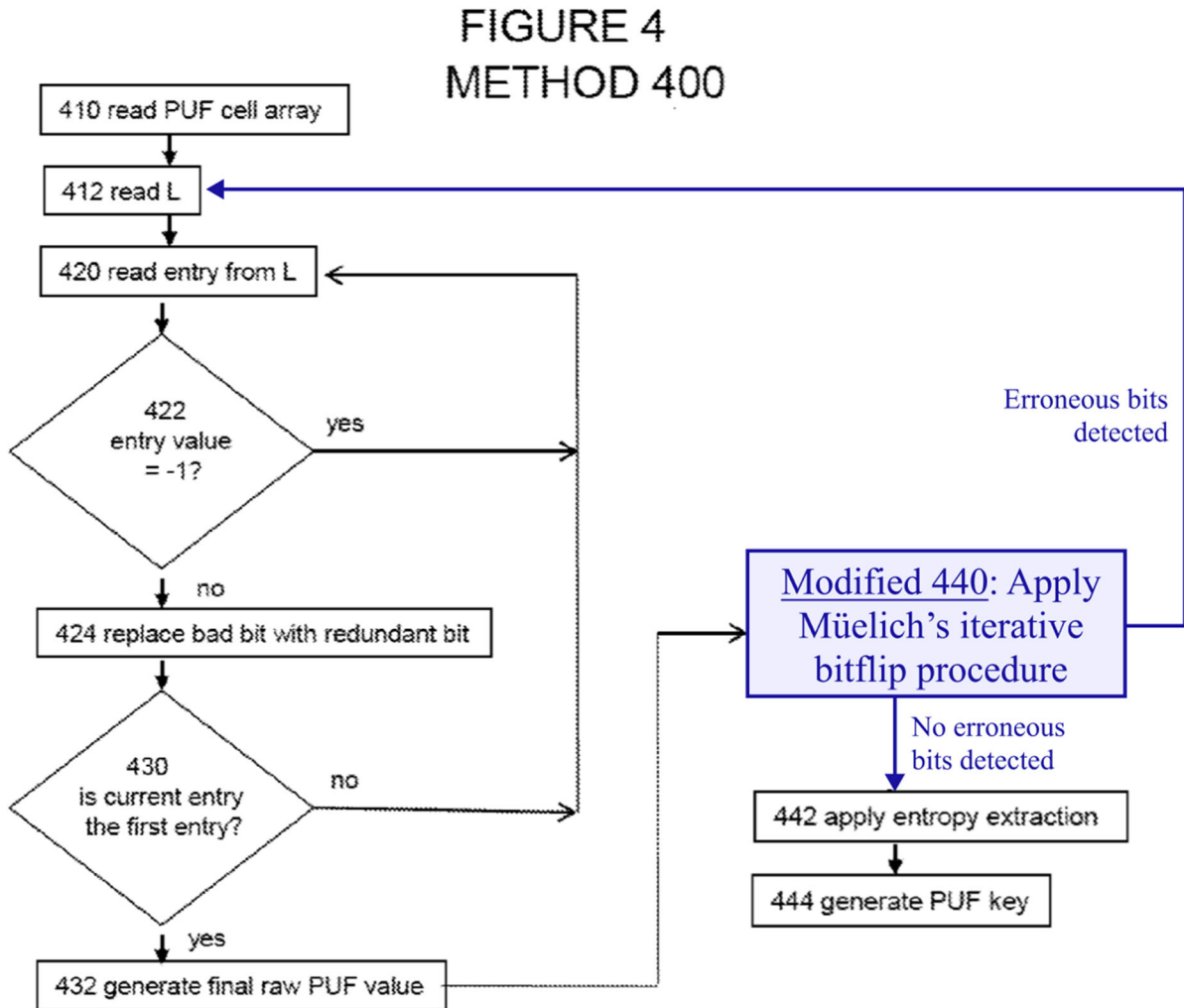
A. GROUND 1A—Claims 6-9, 11, and 14 are obvious in view of Wu and Muelich

1. Combination of Wu and Muelich

59. As discussed below, the Wu-Muelich Combination represents what would have been the predictable and expected result of incorporating Wu’s PUF key generation workflow, including Muelich’s teachings regarding its iterative bitflip procedure to identify erroneous bits in a PUF response (e.g., Wu’s “final raw PUF value”) for recordation in Wu’s redirection list L; and Wu’s suggestion regarding post-manufacturing updating of its redirection list L, which governs list-driven replacement at Wu’s step 424 in subsequent iterations of Wu’s method 400. By incorporating these features, the Wu-Muelich Combination would have improved Wu’s baseline error correction process by enabling Wu’s PUF key generation to perform per-bit evaluations through Muelich’s iterative bitflip procedure to identify additional erroneous bits, update Wu’s redirection list L based on those per-bit evaluations, and perform corrections (i.e., bit replacements) via list-driven substitution at step 424 on a subsequent iteration of method 400. *Id.*

60. In more detail, a POSITA would have implemented Wu’s error correction logic 140 to execute Muelich’s iterative bitflip procedure on the “final

raw PUF value” produced upstream in Wu’s method 400. OKTA-1004, FIG. 4, [0063]-[0064]; OKTA-1005, 4. In operation, the Wu-Müelich Combination would have proceeded from Wu’s evaluation-time readout of the PUF cell array (step 410) to Wu’s list-driven replacement stage (step 424) to yield the “final raw PUF value,” which is then processed by a modified step 440. OKTA-1004, [0063]-[0064]. This is shown in the modified version of Wu’s Figure 4 below:



Wu, FIG. 4 (modified and annotated)

61. As shown above, at the modified step 440, Muelich's iterative bitflip procedure evaluates the final raw PUF value on a per-bit basis to identify additional erroneous bits not previously addressed by the list-driven replacement at step 424. OKTA-1005, 4-5. If Muelich's procedure detects no additional erroneous bits, method 400 proceeds to Wu's downstream entropy extraction at step 442 and key generation at step 444. OKTA-1004, [0064], FIG. 4. If Muelich's procedure identifies one or more additional erroneous bits, however, the Wu-Muelich Combination records the identified bit positions in Wu's redirection list L (consistent with Wu's suggestion regarding implementation flexibility of method 400). OKTA-1004, [0072]. This enables bad-bit identification and replacement to occur post-manufacturing during customer use. *Id.*, [0073]. Upon such identification, the Wu-Muelich Combination would have reverted to step 412, where the updated redirection list L is read. OKTA-1004, [0059]-[0061], [0072]-[0073], FIG. 4. Steps 420, 422, and 424 then traverse the updated L. When the traversal reaches the newly-added entry, step 424 uses that entry to perform list-driven replacement of the bit at the erroneous position identified at modified step 440 with a redundant bit from redundant PUF array 124. OKTA-1004, [0061], FIG. 4.

62. The configuration described above relies on existing Wu infrastructure without introducing any new subsystem. Wu already teaches replacing identified bad bits by selecting redundant bits via redirection list L and performing list-driven

replacement at step 424, and Wu further teaches storing list L in non-volatile memory 172 for later use. OKTA-1004, [0028]-[0030], [0063]-[0064]. Accordingly, upon Muelich's procedure identifying one or more erroneous bit positions at the modified step 440, the Wu-Muelich Combination would have:

- (i) selected an available redundant-bit source from redundant PUF array 124, and
- (ii) updated the mapping in list L stored in NVM 172, making the mapping available for the subsequent reversion to step 412 and the list-driven replacement at step 424 discussed above.

OKTA-1004 [0025], [0063]-[0064], [0073].

63. From a POSITA's perspective, integrating Muelich's iterative bitflip procedure into Wu's open-ended "error correction logic" would have involved applying known solutions to known problems in the field of cryptographic circuit design to achieve predictable outcomes. The fundamental aspects of Wu's disclosure—including the redundant PUF array 124, redirection list L, and redundancy logic 130—would have remained intact, supplemented by the well-known teachings of Muelich to arrive at the claimed configuration with a reasonable expectation of success. *Id.* Likewise, the POSITA would have expected success when applying Wu's own supplemental teaching to continue identifying bad bits

post-manufacturing during customer use. *Id.* The POSITA would have recognized that Muelich's iterative bitflip procedure would readily identify erroneous bit positions in Wu's final raw PUF value, that those positions would be recorded in Wu's redirection list L, and that the bits at those positions would then be replaced by redundant bits on subsequent iterations of Wu's method 400. *Id.*

(a) Applying Muelich's teachings on iterative bitflip procedure in the context of Wu's error correction logic

64. A POSITA would have been motivated to incorporate Muelich's iterative bitflip procedure into Wu's "error correction logic" stage (step 440). The predictable result of this aspect of the Wu-Muelich Combination includes a PUF generation system that evaluates Wu's final raw PUF value on a per-bit basis using Muelich's iterative bitflip procedure to identify erroneous bit positions, thereby reducing the error rate of the PUF key output by Wu's key generation logic 160.

65. The following independent reasons, taken separately or collectively, support this aspect of the combination:

66. **First**, a POSITA would have been motivated to combine these references because Wu expressly invites the use of known error correction approaches, and Muelich provides an exemplary technique suited for this context. Wu teaches an "error correction logic" stage (step 440) but leaves the task of selecting an error correction mechanism open to the POSITA, stating that "[a]ny

known error correction technique may be used.” OKTA-1004, [0025].

67. Muelich teaches an example of a known error correction technique through an iterative bitflip procedure applied to PUF-based key generators that evaluates candidate bit positions on a per-bit basis and terminates when the syndrome weight criterion indicates the error is corrected. OKTA-1005, 1, 5. Given Wu’s explicit suggestion to employ “known” techniques, and Muelich’s disclosure of a bitflip procedure applied to the PUF key-reproduction context, it would have been clear for a POSITA to employ Muelich’s technique to implement Wu’s “error correction logic” to achieve the predictable result of correcting errors in the final raw PUF value.

68. **Second**, a POSITA would have been led to the Wu-Muelich Combination because it involves the application of a known technique (Muelich’s iterative bitflip procedure) to a known system (Wu’s PUF-based key generation workflow) to achieve predictable results (a system that reliably reproduces PUF responses with reduced error rate). Each element within the combination performs similar functions as it would have separately. Wu’s system generates and manages the PUF key lifecycle, and Muelich’s iterative bitflip procedure identifies erroneous bit positions in the PUF response, which are then recorded in Wu’s redirection list L and used to govern list-driven replacement at Wu’s step 424 in subsequent iterations of method 400. *Id.* PUF error correcting codes were well known solutions

to improving PUF reliability before the Critical Date, For example, Yu discloses an “index-based syndrome” code that uses an encoder that generates “indices” that point to “parity bits” that are representative of the PUF response bits. OKTA-1019, 1-8, Yu also describes conventional “XOR masking” where syndrome bits are formed using a “bitwise XOR operation” of parity bits formed by a conventional error correcting code. *Id.*, 8-10. Bit-flipping error correcting techniques, like that described in Muelich, were also well known. Fackenthal is one such example where a “bit flipping component” to identify and correct errors in a codeword by selectively flipping them. OKTA-1020, ¶¶[0018]-[0031].

69. **Third**, a POSITA would have been motivated to employ Muelich’s iterative bitflip procedure in the context of Wu because it uses a simple, efficient approach that is highly suitable for onboard processing.

70. Muelich’s procedure evaluates candidate bit flips on a per-bit basis and selects the single flip that most reduces the syndrome weight at each iteration, a straightforward greedy-descent mechanism implementable in minimal hardware. *Id.* Muelich describes this as a “simple, iterative bitflip procedure.” OKTA-1005, 5. A POSITA would have recognized that using such a streamlined approach would have been particularly appropriate in the context of Wu’s system, where the error correction is performed by the “internal operation” of an integrated circuit. OKTA-1004, [0058]. In implementations where circuit footprint and processing capability

are naturally constrained, a POSITA would have been motivated to select Muelich's "simple" routine to limit the computational complexity and hardware overhead required to implement Wu's "error correction logic." For example, Fainzilber demonstrates that an example of how bit flipping error correcting techniques were known to be efficient around the Critical Date because they can be "implemented with reduced latency and power consumption" when compared to other error correcting codes. OKTA-1021, ¶[0005], *see also* ¶¶[0006]-[0037].

71. **Fourth**, a POSITA would have recognized that applying Muelich's teaching would predictably improve Wu's system efficiency over the device's operational lifetime. As discussed above, each iteration of method 400 in which Muelich's iterative bitflip procedure identifies an erroneous bit position results in that position being recorded in Wu's redirection list L. OKTA-1004, [0029], [0059]-[0061], [0072]-[0073]. A POSITA would have recognized that, as L accumulates entries over successive iterations, a greater portion of the device's erroneous bit positions are handled by Wu's deterministic list-driven replacement at step 424 and a correspondingly smaller portion require iterative diagnosis at modified step 440. OKTA-1004, [0061], [0063], FIG. 4. The Wu-Muelich Combination thus predictably reduces the computational workload of the error-correction stage over the device's operational lifetime while maintaining reliable PUF key regeneration. Again, Fainzilber provides an example of a known configuration round the Critical

Date that involves reduced computation resources used by bit flipping error correcting codes. OKTA-1021, ¶[0005], *see also* ¶¶[0006]-[0037].

(b) Applying Muelich’s iterative bitflip procedure to implement Wu’s teaching of post-manufacturing identification and replacement of bad bits

72. Having implemented Wu’s error correction logic at modified step 440 using Muelich’s iterative bitflip procedure, a POSITA would have been further motivated to utilize the per-bit erroneous position information produced by that procedure to implement Wu’s express teaching regarding post-manufacturing identification and replacement of bad bits to counter device aging effects. *See* OKTA-1004, [0073].

73. Around the Critical Date, a POSITA would have known that PUFs degrade as a device ages, which would have motivated the POSITA to pursue Wu’s direction to implement anti-aging error correction. *Id.* Mai supports this understanding through recognition of a need for anti-aging means in PUFs because device aging results in “bits of the raw PUF response” becoming “inconsistent/unreliable.” OKTA-1022, ¶¶[0004]-[0006], *see also* ¶¶[0020]-[0024]. Schrijen-886 also recognizes that “generation of the response pattern in a semiconductor memory may be sensitive to degradation after repeated applications of the activation signal,” and subsequently proposes writing “anti-aging data” to

memory that documents the states of bits before they degrade. OKTA-1023, ¶¶[0153]-[0161]; *see* OKTA-1025 (“As the PUF circuits age, the PUF values generated by the PUF circuits may drift away from the enrollment values”); *see also* OKTA-1026 (“PUFs are subjected to a set of physical parameters...that cause the output of a PUF to change over time” and “this varying output can result in an increased and undesirable error rate”). In view of the recognized need in the art that PUFs degrade over time, the predictable result of using Muelich’s techniques in Wu’s error correction system would have been to record erroneous bit positions identified at modified step 440 in Wu’s redirection list L stored in NVM 172, such that subsequent iterations of method 400 substitute redundant bit values for those positions via Wu’s list-driven replacement at step 424. In this way, over time, the redirection list would have been updated to reflect bits that degrade and become unreliable (a problem recognized by Wu, and many others in the art). OKTA-1004, [0073]; OKTA-1022, ¶¶[0004]-[0006], *see also* ¶¶[0020]-[0024]; OKTA-1023, ¶¶[0153]-[0161].

74. The following independent reasons, taken separately or collectively, support this aspect of the combination:

75. **First**, a POSITA would have been prompted to pursue this combination based on the express teachings apparent from Wu itself. Wu explicitly identifies a desire to enable “post-manufacturing (e.g., during customer use) identification and

replacement of bad bits,” specifically to “counter device aging effects.” OKTA-1004, [0073]. As discussed above, this problem was well recognized in the art. OKTA-1022, ¶¶[0004]-[0006], *see also* ¶¶[0020]-[0024]; OKTA-1023, ¶¶[0153]-[0161]. However, Wu does not detail the specific technique for diagnosing these new bad bits in the field. Muelich provides such a technique to satisfy this design need. *Id.* Specifically, Muelich’s iterative bitflip procedure identifies the bit position responsible for an erroneous PUF response by sweeping each position $i = 1, \dots, n$, computing the syndrome weight that would result from flipping the bit at position i , and selecting the position whose flipping most reduces the syndrome weight (i.e., most corrects the error). OKTA-1005, 5. The match between Wu’s stated goal and Muelich’s technical solution would have been apparent to a POSITA, who would have sought to combine them to achieve the predictable result of a PUF system that self-updates Wu’s redirection list L as the device ages. As discussed above, a POSITA would have been well aware of the adverse impacts of device aging on a PUF. OKTA-1022, ¶¶[0004]-[0006], *see also* ¶¶[0020]-[0024]; OKTA-1023, ¶¶[0153]-[0161].

76. **Second**, a POSITA would have recognized that updating Wu’s redirection list L based on Muelich’s erroneous bit identifications would predictably improve the long-term reliability of the device. Wu discloses a “redirection list” specifically designed to map “bad bits” to record the address of each bit to be

replaced together with the redundant bit that replaces it, and stores the list in NVM 172 for persistent access across iterations of method 400. OKTA-1004, [0028]-[0030], [0059]. A POSITA would have recognized that when Muelich's iterative bitflip procedure identifies a bit position whose flipping drives the syndrome weight to zero, that bit position is effectively "bad." Rather than requiring Muelich's iterative procedure to re-diagnose the same erroneous bit position on every subsequent iteration of method 400, it would have been obvious to persist this diagnosis by updating Wu's redirection list to map the newly identified bad bit to a redundant bit from redundant PUF array 124. *Id.* This yields a device that incrementally becomes more reliable as L accumulates entries for erroneous bit positions identified over the device's operational lifetime. *Id.*

77. Finally, a POSITA would have also had a reasonable expectation of success in implementing the Wu-Muelich Combination. Wu expressly identifies step 440 as the evaluation-time stage for applying error correction logic to the final raw PUF value and expressly permits the use of "any known error correction technique," including with error-correction data stored in NVM for later use. OKTA-1004, [0025], [0058], [0063], [0064]. Muelich provides an iterative bitflip procedure applicable to PUF-based key reproduction. OKTA-1005, 1, 3, 4. Incorporating Muelich's procedure into Wu's step-440 error-correction logic, and additionally persisting identified erroneous bit positions into Wu's redirection list L

stored in NVM 172, reflects the predictable use of known techniques in the field of PUF error correction to improve an expressly-identified error-correction stage. A POSITA would have implemented this predictable integration of Wu and Muelich through routine engineering choices. As discussed above, bit flipping error correcting codes were known in the art, such that the combination of Wu and Muelich would have simply been the implementation of a known algorithm in a system that expressly invites a POSITA to implement such an error correcting algorithm. OKTA-1021, ¶[0005], *see also* ¶¶[0006]-[0037].

2. Analysis

(a) Claim 6

[6.pre] A method for repairing response bits of a physical unclonable function of an electronic device, said physical unclonable function outputting said response bits and additional output bits, said method comprising the following steps:

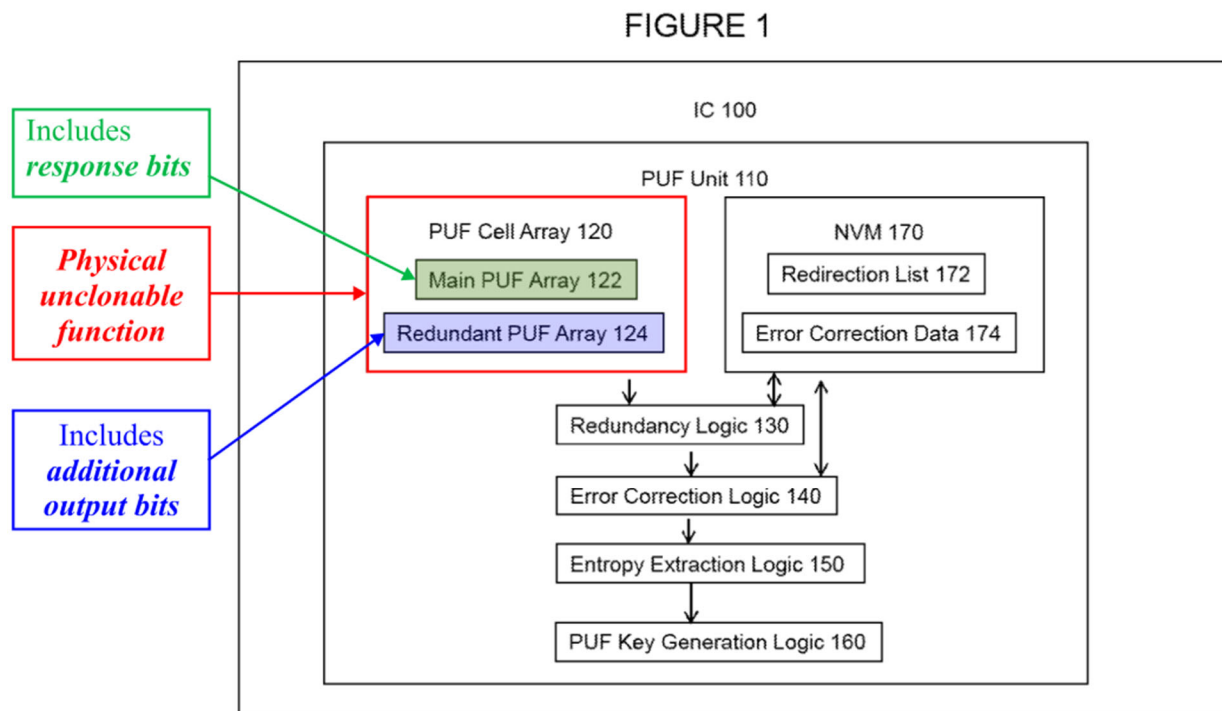
78. As discussed above in §IX.A.1, the Wu-Muelich Combination features Wu's evaluation-time workflow (e.g., method 400) augmented by Muelich's iterative bitflip procedure to facilitate error correction (***method of repairing response bits of a physical unclonable function of an electronic device***).

79. Wu describes “redundancy logic 130 [that] provides for replacing bad bits⁴ in order to reduce the noise and/or error rate of PUF key generation.” OKTA-1004, [0024] (emphasis added). Wu also describes an error correction logic 140 that

⁴ All emphasis has been added unless otherwise indicated.

provides for correcting errors in PUF key generation. *Id.*, [0025].

80. As shown in the annotated version of Wu's Figure 1 below, the Wu-Müelich Combination features an integrated circuit 100 (*electronic device*) having a PUF cell array 120 (*physical unclonable function*) that includes a main PUF array 122 *outputting response bits* and a redundant PUF array 124 *outputting additional output bits*. See OKTA-1004, [0010], [0013], [0015]-[0017], [0020]-[0022], [0027], [0046], [0059], [0063].



OKTA-1004, FIG. 1 (annotated)

81. More specifically, the main PUF array 122 includes “bits” (e.g., “bits b_0 through b_{15} ”) (*id.*, [0046]) that are used for “cryptographic and/or other keys to be generated or derived...” (*id.*, [0021]). The redundant PUF array 124 “may be

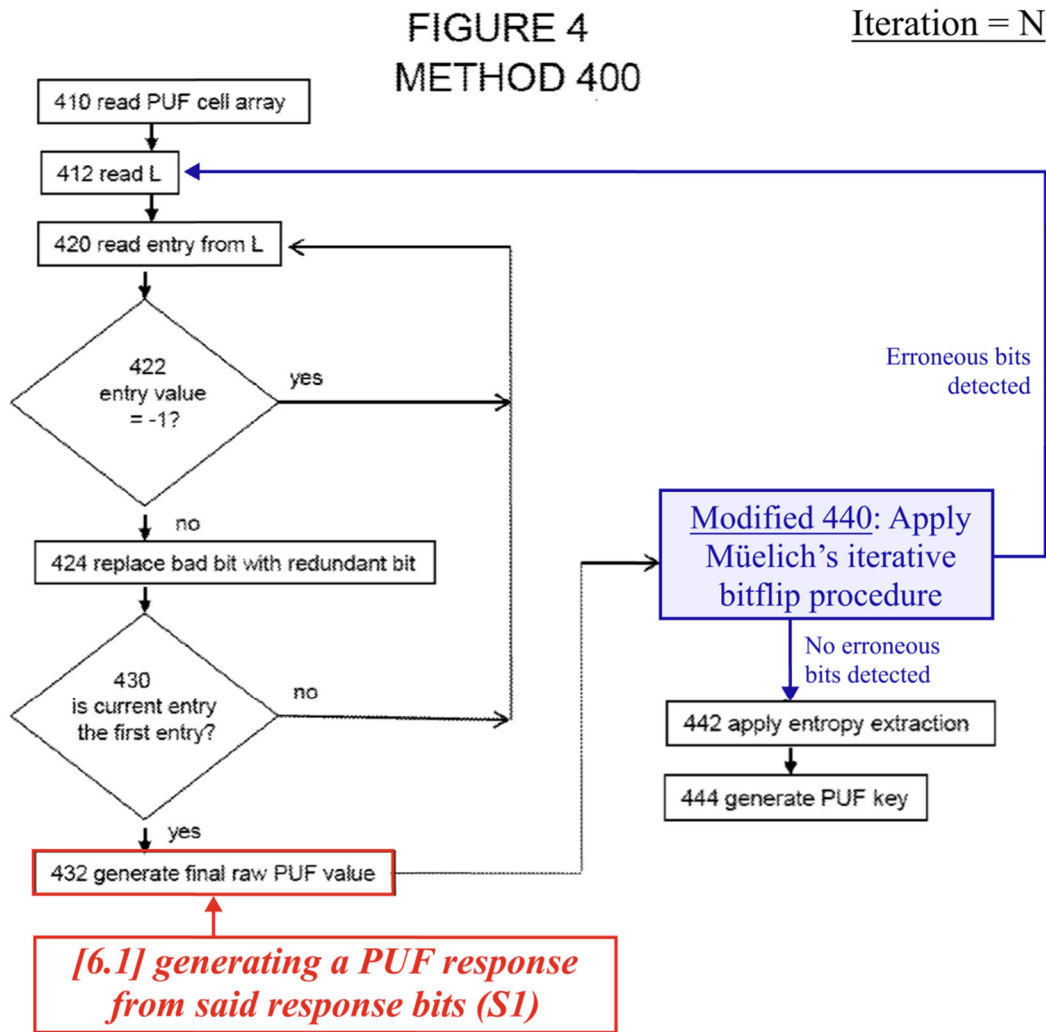
proportional to the size of the main PUF array 122,” for example, “the ratio of redundant PUF cells to main PUF cells may correspond to the raw error rate, plus extra redundant PUF cells to provide a desired margin based on the expected variability in the raw error rate.” OKTA-1004, [0022]; *see also id.*, [0046] (describing a four-bit redundant PUF array comprising bits b_{16} through b_{19}). In Wu’s nomenclature, “a PUF cell from redundant PUF array 12[4] may be referred to as a redundant bit.” *Id.* As discussed above in §IX.A.1, Wu contemplates **repairing response bits** from the main PUF array 122 by selectively replacing them with redundant bits from the redundant PUF array 124. *See generally* OKTA-1004, [0058]-[0073].

[6.1] generating a PUF response from said response bits (S1),

82. The Wu-Müelich Combination would have been configured to produce, from the bits of main PUF array 122 (**response bits**) read at Wu’s steps 410 and 424, a “final raw PUF value” (**PUF response**). *Above*, §IX.A.1. Wu’s method 400 reads PUF cell array 120 at step 410 to obtain an initial raw PUF value.” OKTA-1004, [0058]-[0059], FIG. 4. The method then applies entries in redirection list L at step 424 to substitute redundant-bit values for bits in the initial raw PUF value at the positions identified in those entries. *Id.*, [0061], [0063]. At step 432, method 400 generates the “final raw PUF value” by taking “a first portion of the updated PUF value” for a main PUF array of 128 bits, “the first 128 bits of the updated PUF value

are used as the final raw PUF value.” *Id.*, [0063]. The final raw PUF value is a bitstring generated from main PUF array 122’s response bits, with list-driven substitutions (if any) applied in accordance with the then-current redirection list L. OKTA-1004, [0058]-[0063], FIG. 4.

83. The final raw PUF value produced at step 432 is also ***generated from said response bits***. The final raw PUF value includes response bits as modified by any then-current redirection list L entries. OKTA-1004, [0061], [0063], FIG. 4. For any iteration of method 400 in which L is empty, the final raw PUF value reduces to the unmodified response-bit readout from main PUF array 122. In contrast, for iterations in which L contains one or more entries written during a prior iteration, the final raw PUF value incorporates the corresponding redundant-bit substitutions at step 424 in accordance with Wu’s baseline list-driven replacement mechanism. OKTA-1004, [0061].



OKTA-1004, FIG. 4 (annotated)

[6.2] detecting an error in the PUF response (S2),

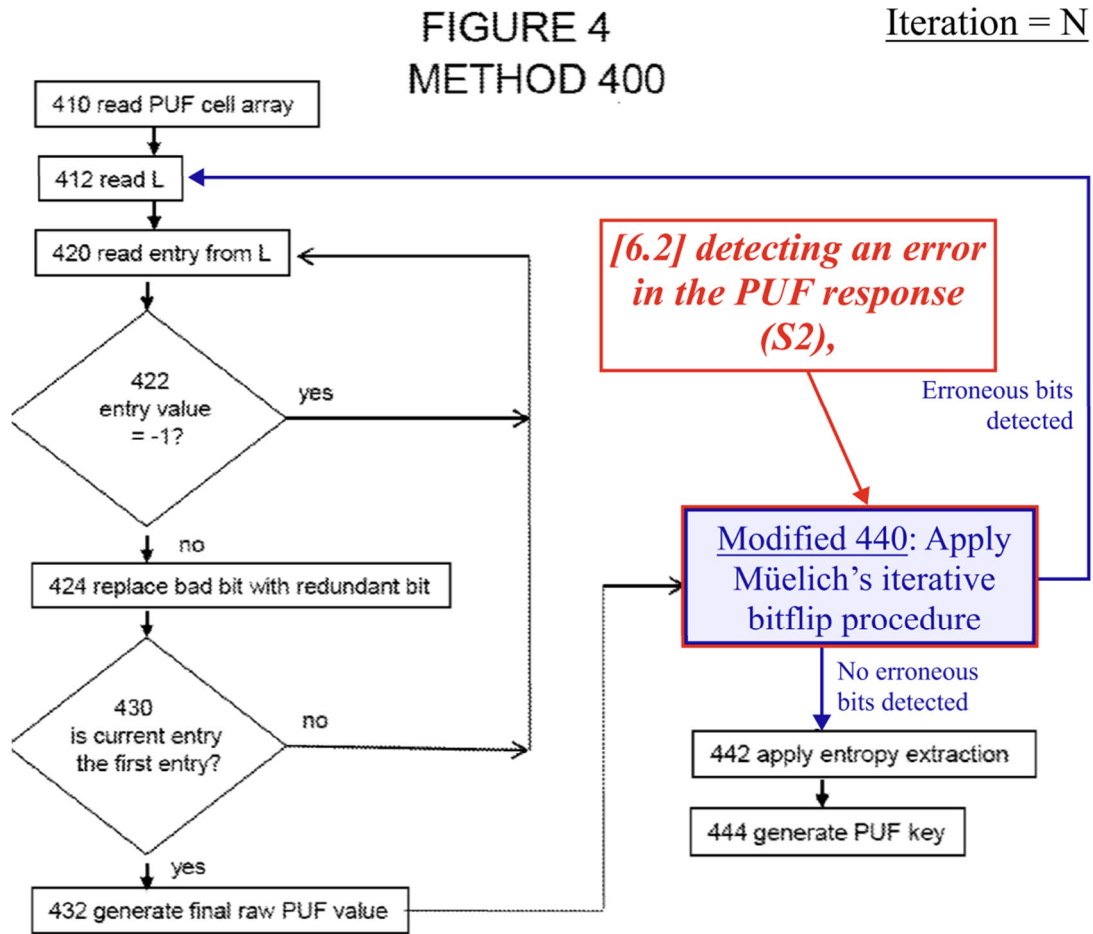
84. The Wu-Muelich Combination would have been configured to detect errors in the final raw PUF value at modified step 440 using Muelich's iterative bitflip procedure (see OKTA-1005, 1-3), where a nonzero syndrome weight ($w \neq 0$) indicates the presence of errors in the final raw PUF value and thereby **detecting an error in the PUF response**.

85. As discussed above in §IX.A.1, Wu's step 440 processes the final raw PUF value using error correction logic 140. OKTA-1004, [0063]. In the Wu-Müelich Combination, this stage is implemented using Müelich's iterative bitflip procedure, which begins by computing the syndrome weight $w = WT(r)$ of the final raw PUF value (treated as a received word r). OKTA-1005, 5. A POSITA would have understood that this initial syndrome evaluation *detect[s] an error in the PUF response* because a nonzero syndrome upon processing indicates the presence of an error. OKTA-1005, 4-5. For example, if the syndrome weight is zero ($w = 0$), then no error is present. However, if the syndrome weight is non-zero, the final raw PUF value contains an error, which the procedure proceeds to identify and correct (*see below* [6.3]). OKTA-1005, 5 (Algorithm 1, lines 1-4).

86. Wu further confirms through a separate manufacturing-time workflow (method 300), that bit-level disagreement in a raw PUF value is the mark of error. Specifically, Wu detects per-cell instability by generating an initial raw PUF value R , changing an operating condition (e.g., temperature/voltage), and generating a second raw PUF value R' under that changed condition. OKTA-1004, [0035]-[0041], FIG. 3. Wu detects cell-level disagreement by performing a bitwise XOR between R and R' , the output of which "may serve to indicate which PUF cells have changed their value between the initial measurement and the new measurement, and may therefore be considered bad." *Id.*, [0039]. Although Wu's method 300

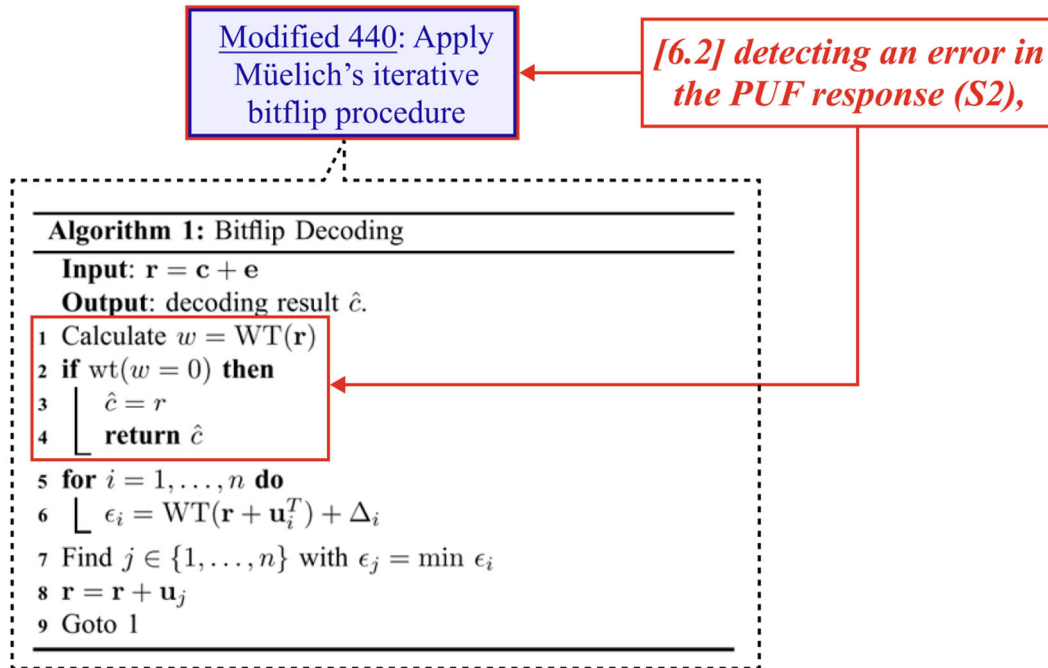
performs this detection at manufacturing time using two measurements under differing conditions, this disclosure demonstrates that Wu at least contemplates bit-level disagreement in a raw PUF value as an indicator of error. The Wu-Müelich Combination applies similar underlying principles at evaluation time, detecting such bit-level disagreement in the final raw PUF value via Müelich's syndrome-based mechanism at modified step 440. *Id.*

87. Wu's redundancy logic 130 and error correction logic 140 together provide multiple layers of error detection and correction in Wu's PUF key generation architecture. OKTA-1004, [0023]-[0025]. In the Wu-Müelich Combination, Müelich's iterative bitflip procedure at modified step 440 operates as the evaluation-time error-detection layer, applied to the final raw PUF value produced in method 400.



OKTA-1004, FIG. 4 (annotated)

Iteration = N



OKTA-1005, 5 (annotated)

[6.3] determining erroneous response bits of the PUF response (S3) comprising replacing one by one, in a [predetermined] order, each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is corrected,

88. As explained above in §VI.C, based on plain meaning, [6.3] does not mandate replacement of every single bit to satisfy the step of **determining erroneous response bits of the PUF response (S3)**. OKTA-1001, 6:29-36, FIG. 4. Consistent with this plain meaning understanding, the Wu-Muelich Combination renders [6.3] obvious through Muelich's iterative bitflip procedure at modified step 440, which, upon detecting a nonzero syndrome weight (*see above*, [6.2]) sweeps through each

bit position of the final raw PUF value in positional order ($i = 1, \dots, n$), tentatively replaces the bit at each position with its complement, and evaluates the resulting syndrome weight to determine whether that replacement corrects the error. *See above*, §IX.A.1; OKTA-1004, [0058]-[0064], FIG. 4; OKTA-1005, 3-5.

89. As discussed below, the Wu-Müelich Combination would have used an iterative bitflip procedure to tentatively *replace*, one by one, each bit of the final raw PUF value with its complement (*arbitrary value*), and determined whether that replacement corrects the error by evaluating the resulting syndrome weight (*determining if the error in the PUF response is...corrected*). The complement of a PUF response bit is arbitrary as the bit complement is not based on a PUF primitive. Said simply, flipping a PUF response bit produces a result that is not based strictly on the PUF itself (and is thus arbitrary from the perspective of the response). In the context of binary bit flipping, a POSITA also would have understood that the complement of a randomly flipped bit is arbitrary, as binary PUF bits included only two possible values (i.e., 1, or 0). OKTA-1005, 1.

90. More specifically, Wu's error correction logic 140 "provide[s] for correcting errors in the generation of PUF keys from PUF cell array 120," and NVM 170 stores "error correction data 174" for use by the error correction logic. OKTA-1004, [0058]-[0064], FIG. 4. During evaluation time, Wu's method 400 involves reading/evaluating the PUF cell array (step 410) and performing list-driven

substitution using redirection list L (step 424) to yield a “final raw PUF value,” which is then “processed by error correction logic 140” at step 440. *Id.* Wu contemplates an error correction routine applied to a PUF response (final raw PUF value generated upstream in method 400) as part of the key generation pipeline. *Id.*

91. Muelich’s Algorithm (reproduced below) depicts the iterative bitflip procedure that the Wu-Muelich Combination applies at modified step 440, rendering obvious the step of *replacing...[a] bit of the PUF response by an arbitrary value and determining if the error in the PUF response is corrected.*

Algorithm 1: Bitflip Decoding

Input: $\mathbf{r} = \mathbf{c} + \mathbf{e}$
Output: decoding result $\hat{\mathbf{c}}$.
1 Calculate $w = \text{WT}(\mathbf{r})$
2 **if** $\text{wt}(w = 0)$ **then**
3 $\hat{\mathbf{c}} = \mathbf{r}$
4 **return** $\hat{\mathbf{c}}$
5 **for** $i = 1, \dots, n$ **do**
6 $\epsilon_i = \text{WT}(\mathbf{r} + \mathbf{u}_i^T) + \Delta_i$
7 Find $j \in \{1, \dots, n\}$ with $\epsilon_j = \min \epsilon_i$
8 $\mathbf{r} = \mathbf{r} + \mathbf{u}_j$
9 Goto 1

OKTA-1005, 5

92. More specifically, where the syndrome weight is nonzero (indicating the presence of an error per [6.2] above), Muelich’s Algorithm 1 proceeds to sweep each bit position i of the final raw PUF value in a predetermined positional order ($i = 1, \dots, n$). For each position i , Algorithm 1 computes $\epsilon_i = \text{WT}(\mathbf{r} + \mathbf{u}_i^T) + \Delta_i$, which quantifies the syndrome weight that would result if the bit at position i were replaced

with its complement. OKTA-1005, 5 (Algorithm 1, lines 5-6). A POSITA would have understood that this per-bit computation is an example of *replacing one by one, in a predetermined order, each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is corrected*. Specifically, bit positions are evaluated in a predetermined positional order such that the $r + u_i^T$ term replaces the bit at position i with its complement (*arbitrary value* for a binary bit). Further, $WT(r + u_i^T)$ evaluates whether that replacement would correct the error by measuring the resulting syndrome weight.

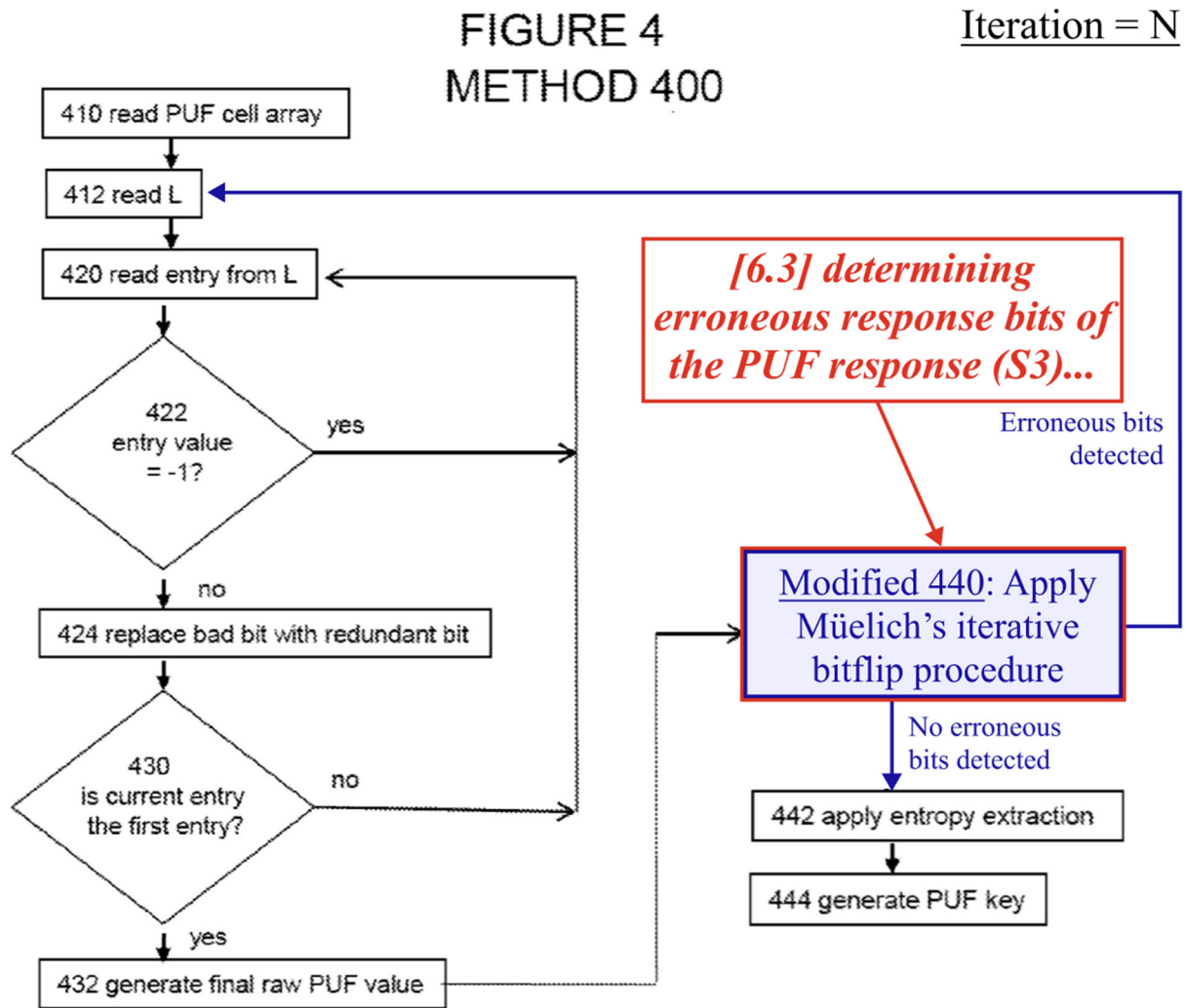
93. Muelich's per-bit evaluation is performed in a *predetermined order* within the meaning of [6.3] in various ways. For example, Algorithm 1 of Muelich prescribes an *order* of its sweep since line 5 executes the sweep as for $i = 1, \dots, n$, which sweeps the bit positions in fixed positional sequence from position 1 through position n . OKTA-1005, 5 (Algorithm 1, line 5). This sequence is established by the algorithm in advance of the decoder's execution and does not depend on the contents of the received word, the identity of any erroneous bit, or any other data-dependent parameter (i.e., is deterministic), and thus is *predetermined*. Said simply, the predetermined order is the order between positions 1 and n .

94. As another example, as would have been apparent to a POSITA, Muelich's process involves bit-by-bit replacements according to the calculation, at each iteration, of a minimal ε_j that indicates the next most likely erroneous bit. The

execution of these steps is deterministic (i.e., predetermined) in that the order in which steps may be executed based on the minimal ε_i calculation at each iteration. OKTA-1005, 5. Said simply, the order of the process is the order dictated by presence of the errors themselves. *Id.* This understanding is also consistent with the '359 patent specification, which explains that an example of a bit-flipping process being performed in “a predetermined order” involves “knowledge about how reliable the PUF response bits are to test the least reliable bits first.” OKTA-1001, 5:12-25 (identifying “a predetermined order” as “test[ing] the least reliable bits first”). A POSITA would have understood and/or found it obvious that Muelich’s Algorithm 1 similarly may leverage the minimal ε_i calculation at each iteration to execute bit-flipping according to the value of each minimal ε_i , such that the mechanism produces a commit order in which less reliable bit positions are biased towards being committed earlier, with the bias arising from the contribution of Δ_i to minimal ε_i .

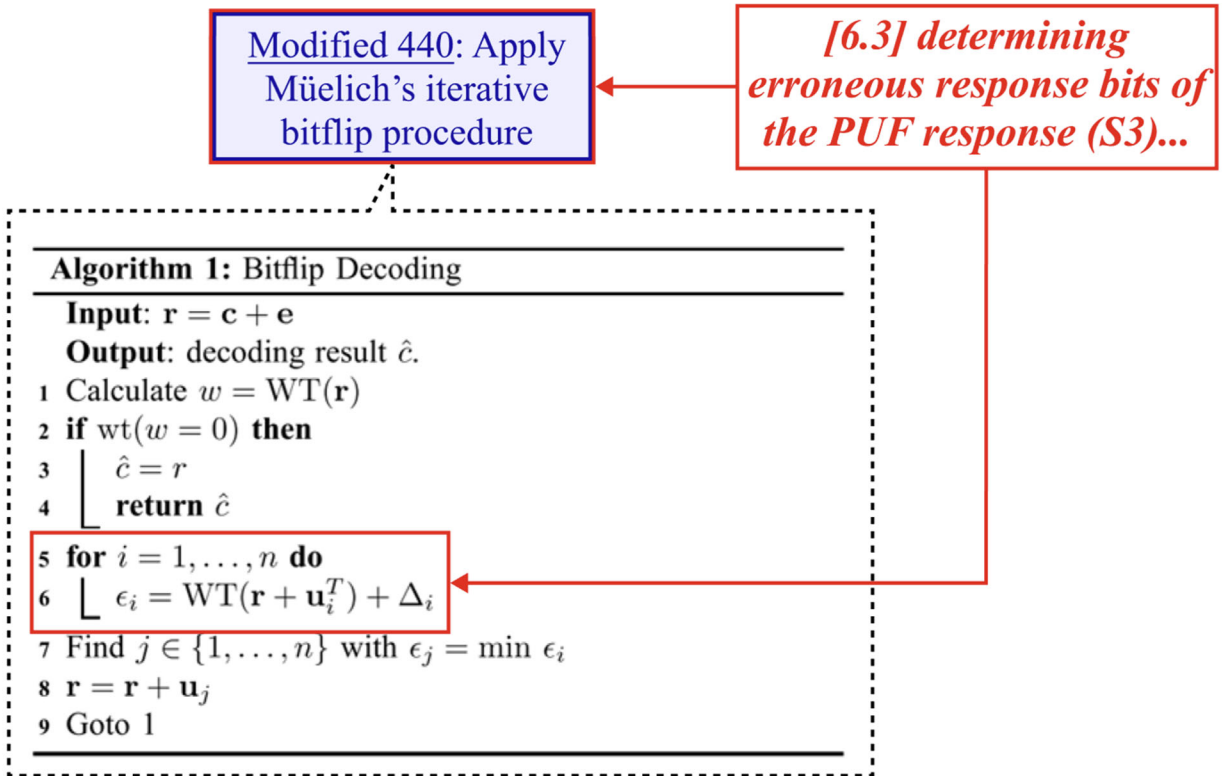
95. Further, as discussed above in §IX.A.1, the Wu-Muelich Combination implements Wu’s error correction stage at modified step 440 using Muelich’s iterative bitflip procedure. OKTA-1004, [0064], FIG. 4; OKTA-1005, 4-5. When Muelich’s procedure sweeps bit positions $i = 1, \dots, n$ of the final raw PUF value and evaluates $\varepsilon_i = \text{WT}(r + u_i^T) + \Delta_i$ at each position, this involves, on a per-bit, predetermined-order basis, replacement and error-correction test operations recited in [6.3]. OKTA-1005, 5 (Algorithm 1, lines 5-6). The Wu-Muelich Combination

satisfies [6.3], or at least renders it obvious, for any iteration of method 400 in which Muelich's procedure determines a nonzero syndrome weight.



OKTA-1004, FIG. 4 (annotated)

Iteration = N



OKTA-1005, 5 (annotated)

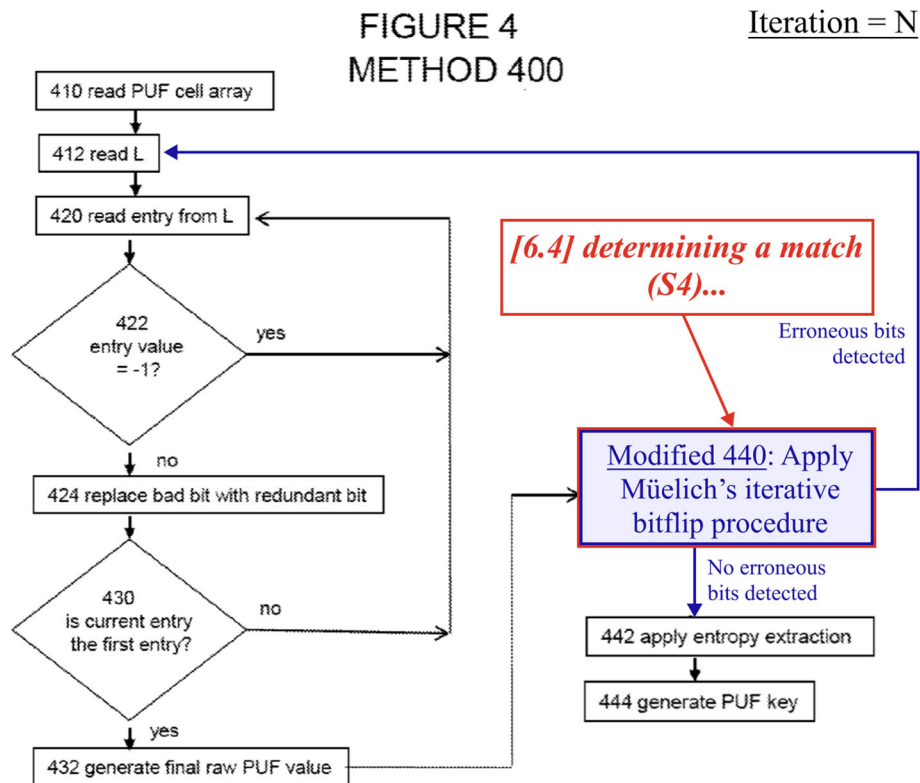
[6.4] determining a match (S4) between each determined erroneous bit and a selected additional output bit such that replacing the erroneous response bits by the matching additional output bits corrects the error in the PUF response,

96. As explained in §VI.C and discussed above in [6.3], based on plain meaning, the claim language does not mandate that multiple erroneous bits be identified and addressed in every iteration. OKTA-1001, 6:29-36, FIG. 4. Consistent with this plain meaning understanding, the Wu-Muelich Combination renders [6.4] obvious in the context of the evaluation-time workflow discussed above in [6.3],

where Muelich's bitflip procedure at modified step 440 (e.g., Algorithm 1, line 7) identifies an erroneous bit position in the final raw PUF value (i.e., a ***determined erroneous response bit***). See above, §IX.A.1; OKTA-1004, [0064], FIG. 4; OKTA-1005, 3-5. The Wu-Muelich Combination would have determined a ***match*** upon the procedure identifying bit position *j* by selecting an available redundant bit from redundant PUF array 124 (***selected additional output bit***) and associating that redundant bit with the erroneous bit position via Wu's redirection list mapping. *Id.* The resulting (*j*, redundant-bit) pairing involves a match that ***correct[s] the error in the PUF response*** because Muelich's iterative bitflip procedure selects bit position *j* as the position whose flipping drives the syndrome weight towards zero (i.e., corrects the error).

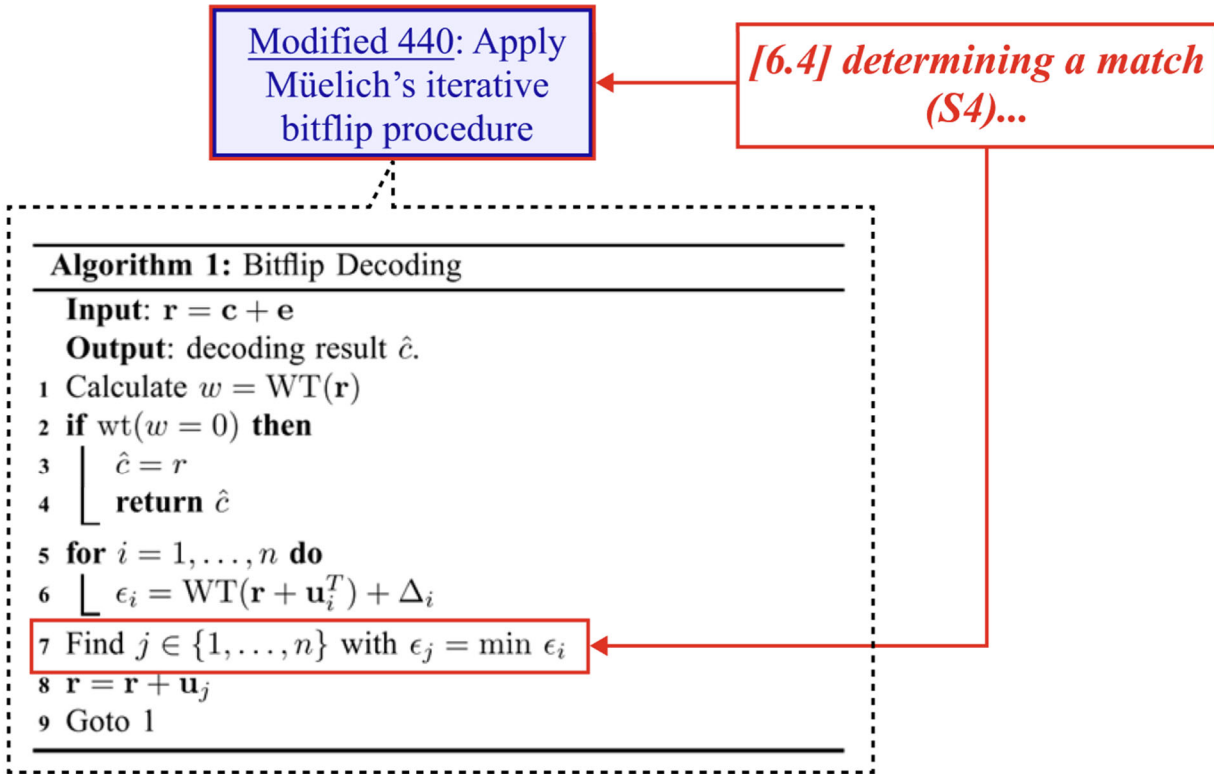
97. As discussed above in [6.pre], Wu's PUF cell array 120 includes a main PUF array 122 (***response bits***) and a redundant PUF array 124 (***additional output bits***), and redundancy logic 130 is configured to evaluate PUF cells and replace PUF cells failing stability requirements to reduce noise and error rate in PUF key generation. OKTA-1004, [0023]-[0024]. In Wu, the redirection list L stored in NVM 172 records, for each main array bit position to be replaced, an entry identifying the redundant bit selected to replace it. OKTA-1004, [0059]-[0060], FIG. 4. A POSITA would have understood that this association constitutes ***determining a match*** because the redirection-list entry establishes the

correspondence between (i) the identified erroneous bit position in the main array (e.g., bit position j identified at modified step 440 by Muelich's iterative bitflip procedure), and (ii) a particular redundant bit selected from redundant PUF array 124 for use in replacement at that position. OKTA-1004, [0024], [0059]-[0060], FIG. 4. Wu describes that the redirection list, used by its "redundancy logic 130," is for identifying "bad bits," and therefore, the correlation between erroneous bits in the main array and the redundant bits in the redundant PUF array is evident based on Wu's FIG. 4 process where bad bits are read and then replaced with their corresponding redundant bit (indicated in the redirection list). *Id.*



OKTA-1004, FIG. 4 (annotated)

Iteration = N



OKTA-1005, 5 (annotated)

[6.5] storing said determined match in a repair list (S5),

98. The Wu-Muelich Combination would have *stored* a redundant bit selected from redundant PUF array 124 in [6.5] in a redirection list L stored in NVM 172 (*repair list*). Above, §IX.A.1. Consistent with Wu, each entry within the redirection list L identifies a main-array bit position along with the redundant bit selected to replace it. OKTA-1004, [0028]-[0030]. The stored match persists in NVM 172 across successive iterations of method 400. At the beginning of each

subsequent iteration, Wu's step 412 reads the current contents of L from NVM 172 and provides them as input to the sequence of steps 420, 422, and 424. OKTA-1004, [0059]-[0061], FIG. 4. This enables step 424 of a subsequent iteration to apply a stored match as a list-driven replacement. *See below*, [6.6].

[6.6] replacing ([S6]) in the PUF response said erroneous response bits with the matching additional output bits in the repair list.

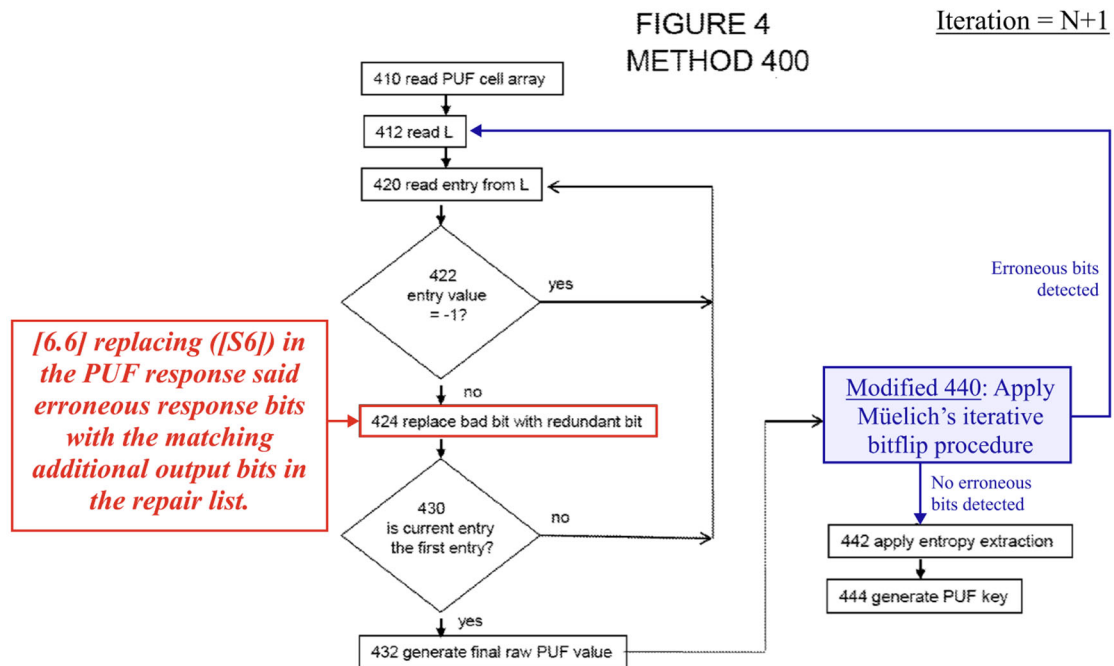
99. The Wu-Müelich Combination would have satisfied the requirements of [6.6] through Wu's list-driven replacement mechanism at step 424, applied during the iteration of the method 400 that follows the iteration in which the current match is written to redirection list L. *See above*, §IX.A.1. Following the storage of a current match in NVM 172, control reverts from modified step 440 to step 412 of a subsequent iteration where the updated redirection list L is read. OKTA-1004, [0059], FIG. 4. The execution of steps 420, 422, and 424 then proceeds through entries of the redirection list L. Once traversal reaches the stored entry, step 424 substitutes the redundant bit's value at bit position *j* in the PUF response (*replacing...erroneous response bits with the matching additional output bits in the repair list*). *Id.*, [0059]-[0063], FIG. 4.

100. The configuration of the Wu-Müelich Combination discussed above is also consistent with the '359 specification's disclosure of repair list matching across successive PUF response generations. For example, as shown in Figure 4, during a

generation of a PUF response, “the PUF response bits are first corrected according to the matching stored in the repair list,” after which additional erroneous bits (if any) are identified and added to the list. OKTA-1001, 6:28-40, FIG. 4. Additionally, the method “may be applied repeatedly at successive generations of the PUF response” and that, at the first step of a subsequent generation, previously-stored matchings are used for “replacing in the PUF response the matched response bits with the matching additional output bits in the repair list.” *Id.*, 5:22-28. Consistent with this disclosure, the Wu-Müelich Combination similarly would have stored a match identified in the modified step 440 in redirection list for use in the list-driven substitution at step 424 of a subsequent iteration of method 400. This in-memory update corresponds to replacing, in the PUF response, the erroneous bit at position j with the matching additional output bit (the redundant bit) identified in the repair list updated in the preceding step S5.

101. The configuration discussed above is consistent with Wu’s teaching of using the redirection list values to update PUF responses. Wu explains that “the value of the redundant bit corresponding to the current entry is used to replace the value of the bit whose address ... is stored in the current entry,” yielding an updated value where the bad bit is replaced. OKTA-1004, [0061]. While Wu’s baseline method performs this replacement at step 424 for *previously* known errors, in the Wu-Müelich Combination, Müelich’s iterative routine would have identified the

erroneous bit position j , and Wu's redirection list mapping is applied to select an available redundant bit from redundant PUF array 124 as the replacement value for that position (ensuring the final PUF key is generated from corrected data). Furthermore, because the redirection list L is updated in non-volatile memory (*see above* [6.5]), subsequent evaluation-time invocations of method 400 would have involved applying this replacement at step 424. OKTA-1004, [0061].



OKTA-1004, FIG. 4 (annotated)

(b) Claim 7

[7.1] The method according to claim 6, wherein said device being configured for having access to a memory storing a preventive toggling list of the least reliable response bits of said physical unclonable function,

102. The Wu-Muelich Combination would have provided IC 100 (*said*

device) configured to access a sorted list of response-bit positions stored in NVM 172, where the list is sorted from least reliable to most reliable based on per-bit reliability characterizations produced by Muelich's multi-readout mechanism during device setup (*preventive toggling list of the least reliable response bits*). OKTA-1005, 4; OKTA-1004, [0010], [0017]-[0018], [0028]-[0030].

103. Muelich describes a per-bit reliability characterization mechanism that classifies response-bit positions as least reliable or more reliable. *See* OKTA-1005, 4. During device setup (e.g., during device enrollment before evaluation-time operation), Muelich's mechanism obtains multiple readouts of each response bit position (e.g., $m = 3$ readouts) and computes, for each bit position $i \in \{1, \dots, n\}$, a reliability score Δ_i based on whether the readouts agree at that position. For example, when $\Delta_i = \delta_1$ when all readouts agree (indicating higher reliability) and $\Delta_i = \delta_2$ when the readouts disagree (indicating lower reliability), where $\delta_1 > \delta_2 > 0$. OKTA-1005, 4. A POSITA would have understood that per-bit reliability characterization identifies, for each device, which response bit positions of the PUF are least reliable (positions that exhibit non reproducible behavior across readouts). In Muelich, these reliability scores are used during evaluation-time error correction at Algorithm 1, where Δ_i is incorporated into the ε_i computation (*see above* [6.4]) favor bit positions that are less reliable. OKTA-1005, 5.

104. The configuration discussed above is supported by Wu's suggestion of

generating per-bit characterization data during device setup and persistent storage of the characterization data in NVM. In Wu's method 300, PUF cell array 120 is characterized during manufacturing time under varying conditions to identify stable cells. OKTA-1004, [0035]-[0045], FIG. 3. The results of that characterization are stored in redirection list L in NVM 172 and accessible for evaluation-time use during method 400. *Id.*, [0028]-[0030], [0045], [0059]. A POSITA would have recognized that this storage architecture (list-structured per-bit data in NVM 172) supports persistent storage of Muelich's setup-time reliability characterization for evaluation-time access (e.g., in the modified step 440 as discussed above in [6.2]-[6.6]).

105. In the Wu-Muelich Combination, Muelich's setup-time reliability characterization is applied to each response bit position of main PUF array 122, producing a per-bit Δ_i value for each of the n response bit positions. OKTA-1005, 4. The combination sorts these bit positions from least reliable ($\Delta_i = \delta_2$) to most reliable ($\Delta_i = \delta_1$) and stores the sorted list of bit position references in NVM 172, which is accessible alongside redirection list L. OKTA-1004, [0028]-[0030]. Consistent with Wu's architecture for NVM 172, during evaluation-time operation of method 400, IC 100 is thereby configured to access the sorted list (stored in memory configured for evaluation-time access) (*configured for having access to a memory storing a preventive toggling list of the least reliable response bits*). OKTA-1004, [0059], FIG. 4.

[7.2] determining erroneous response bits of the PUF response to be repaired (S3) comprises replacing first bits of the Preventive toggling list before replacing other bits of the PUF response.

106. The Wu-Müelich Combination renders [7.2] obvious through reliability-weighted operation of Müelich's iterative bitflip procedure, which causes bit positions identified as least reliable by a sorted list of bit position references in NVM 172 (***Preventive toggling list***) to be committed for replacement (***replacing first bits***) before more reliable bit positions are committed (***replacing other bits***). OKTA-1005, 4-5; OKTA-1004, [0024], [0028]-[0030].

107. Specifically, Müelich's Algorithm 1 computes $\varepsilon_i = \text{WT}(\mathbf{r} + \mathbf{u}_i^T) + \Delta_i$ at each bit position i (*see above* [6.3]), where Δ_i is the per-bit reliability score that governs the ordering of bit position references in NVM 172 (*see above* [7.1]). OKTA-1005, 4-5. Less-reliable positions are assigned lower Δ_i values, and thus, their corresponding ε_i values are lower than those of more-reliable positions. Because ε_i incorporates both the error-correction metric ($\text{WT}(\mathbf{r} + \mathbf{u}_i^T)$) and the reliability weight (Δ_i), Algorithm 1's execution favors error-correcting flips at less reliable bit positions. OKTA-1005, 5. At each iteration, Algorithm 1 commits the single flip whose bit position minimizes ε_i such that less-reliable positions (corresponding to the first entries of the preventive toggling list) are committed first, followed by progressively more reliable positions in subsequent iterations until the syndrome weight is driven to zero. OKTA-1005, 5.

108. The reliability-weighted commit order described above is consistent with the '359 specification's description of the "preventive toggling list." Specifically, the preventive toggling list "comprise[s] the reference numbers of the response bits ... sorted from the least to the most reliable response bits," such that "[t]he most probable erroneous bits may therefore be tested first." OKTA-1001, 5:15-21. As discussed above, the Wu-Müelich Combination's execution of Müelich's Algorithm 1 (in which least reliable bit positions are evaluated prior to more reliable bit positions) functions in a similar manner. The Wu-Müelich Combination therefore discloses the features of [7.2] or at least renders them obvious.

(c) **Claim 8**

[8.1] The method according to claim 6, wherein said electronic device is configured for access to a memory storing said repair list comprising at least one match between a response bit and an additional output bit,

109. The Wu-Müelich Combination would have provided an electronic device (e.g., IC 100) (***said electronic device***) configured to access non-volatile memory storing a redirection list L (***memory storing said repair list***), where redirection list L specifies a match between a main array response bit position and a redundant array additional output bit source (***match between a response bit and an additional output bit***). OKTA-1004, [0058]-[0072].

110. As discussed above in [6.5] and [6.6], NVM 172 stores redirection list

L, and Wu's evaluation-time workflow (method 400) includes reading redirection list L from the non-volatile memory (step 412) for use in repairing a generated PUF response. OKTA-1004, [0059]. A POSITA would have understood that this configuration (e.g., retrieving list data from non-volatile storage during evaluation-time processing) reflects the Wu-Müelich Combination device *configured for access to a memory storing the repair list*.

111. Moreover, as discussed above in [6.pre] and [6.4], PUF cell array 120 includes a main PUF array 122 (*response bits*) and a redundant PUF array 124 (*additional output bits*). OKTA-1004, [0020]-[0022]. The redirection list L records, for a selected main array bit position to be repaired (*response bit* position), which redundant bit is selected as a replacement source (*additional output bit*). OKTA-1004, [0024], [0059], [0067]-[0069]. For example, during evaluation time, entries of the redirection list L are iterated and, for each entry, "the value of the redundant bit corresponding to the current entry is used to replace the value of the bit whose address...is stored in the current entry." OKTA-1004, [0061]. A POSITA would have understood that each stored entry embodies a match (i.e., an association) between a response bit address in the main PUF array 122 and a selected redundant bit source in the redundant PUF array 124.

112. As discussed in §IX.A.1, in the Wu-Müelich Combination, Wu's error-correction stage (modified step 440) incorporates Müelich's iterative bitflip

procedure to identify residual erroneous bit positions and then persists those results into Wu's existing redirection list framework by updating and storing the redirection list L in non-volatile memory for subsequent evaluation time use across successive iterations of method 400. OKTA-1004, [0024], [0059], [0067]-[0069]; OKTA-1005, 1, 3-5. The Wu-Müelich Combination thereby incorporates Wu's configuration for accessing a memory storing the repair list during operation (step 412), and reinforces that configuration by persisting additional match entries (between main PUF array bit positions and redundant bit sources) into the stored repair list structure over time across successive iterations of method 400, including to address post-manufacturing aging-related instability. OKTA-1004, [0059]; OKTA-1005, 1, 3, 4.

[8.2] the step of generation of a PUF response (S1) comprises replacing in the PUF response said matched response bits with the matching additional output bits in the repair list.

113. See above [6.6]. During the step recited in [6.1] (***S1***), the Wu-Müelich Combination would have formed the final raw PUF value (***PUF response***) using the redirection list L to ***replace*** one or more bits in a PUF response (***said matched response bits*** in the ***PUF response***) with a corresponding redundant bit value (***matching additional output bits***), consistent with Wu's redirection list replacement mechanism. OKTA-1004, [0058]-[0072].

114. More specifically, as discussed above in [6.1] and [6.6], the evaluation-

time workflow provided by the Wu-Müelich Combination involves reading/evaluating the PUF cell array to obtain an initial raw PUF value (step 410) and applying redirection list substitutions (using redirection list L read from NVM 172 at step 412) to form an “updated PUF value” from which a “final raw PUF value” is generated. OKTA-1004, [0063]. Wu explains that, for each entry in redirection list L, “the value of the redundant bit corresponding to the current entry is used to replace the value of the bit whose address ... is stored in the current entry” (step 424). OKTA-1004, [0061]. A POSITA would have understood that this list-driven replacement is part of generating the final raw PUF value (*PUF response*), and therefore this process involves *replacing* matched response bit positions (*said matched response bits*) with corresponding redundant-bit values identified in the repair list (*the matching additional output bits*).

115. Further, as discussed in §IX.A.1, the Wu-Müelich Combination also persists newly-identified erroneous bit positions identified during the modified error correction stage at modified step 440 (*see above* [6.3]-[6.5]). This involves updating redirection list L to include additional match entries, which are then available to be applied by the redirection list driven substitution mechanism during subsequent iterations of method 400. OKTA-1004, [0058]-[0072]; OKTA-1005, 1, 3, 4. Specifically, during a subsequent iteration of method 400, step 412 reads the updated redirection list L from NVM 172 and the process involving steps 420, 422, and 424

involves applying the newly-added match entry at step 424. This involves substituting the redundant bit's value at the erroneous position j during the generation of that subsequent iteration's final raw PUF value. OKTA-1004, [0059]-[0063].

(d) Claim 9

[9] The method of claim 6, wherein said additional output bit are bits among most reliable output bits of said physical unclonable function.

116. As discussed for [7.1], the Wu-Müelich Combination would have selected redundant bits from redundant PUF array 124 (***additional output bits***) using reliability metrics calculated using Müelich's reliability characterization mechanism (Δ_i) (***most reliable output bits***). OKTA-1005, 4. During device setup, Müelich's multi-readout mechanism characterizes each PUF cell of PUF cell array 120 (including cells of both main PUF array 122 and redundant PUF array 124) by obtaining multiple readouts and computing a per-bit reliability score Δ_i based on readout agreement, with $\Delta_i = \delta_1$ for reliable cells (all readouts agree) and $\Delta_i = \delta_2$ for less reliable cells (readouts disagree). *Id.*; see OKTA-1004, [0028]-[0030], [0059].

117. As discussed above in [7.1], Müelich's mechanism obtains multiple readouts (e.g., $m = 3$ readouts) of each response bit position, evaluates readout agreement, and assigns a reliability score Δ_i to each bit position. OKTA-1005, 4. A POSITA would have understood that this characterization mechanism applies

uniformly to PUF cells in PUF cell array 120 (both main PUF array 122 and redundant PUF array 124) because the underlying physical phenomenon (measurement-to-measurement variability in a PUF cell's output) is a property of the cell itself, not of its functional role as a "main" or "redundant" bit. The output of the characterization (e.g., Δ_i per bit) is therefore available for every cell in PUF cell array 120, and selection of ***additional output bits*** can draw from those redundant PUF array 124 cells characterized as most reliable ($\Delta_i = \delta_i$). *Id.*

118. Moreover, the selection of reliable redundant bits is consistent with the repair mechanism provided by the Wu-Müelich Combination because it facilitates execution across successive iterations of method 400. As discussed above in §IX.A.1 and [6.6], over multiple iterations, matches associated with erroneous bit positions j are written to redirection list L and persisted in NVM 172. OKTA-1004, [0059]-[0063], FIG. 4. A POSITA would have recognized that if redundant bits were themselves unstable (i.e., their value varied across readouts), substitution at step 424 would propagate that instability into the final raw PUF value, undermining the repair mechanism. Accordingly, to avoid such instability, the combination would have been configured to select redundant bits from among the most reliable PUF output bits (i.e., cells for which Müelich's multi-readout characterization produced $\Delta_i = \delta_i$, indicating consistent output across readouts). OKTA-1005, 4.

119. The reliability-based selection is also consistent with Wu's

architecture, which treats redundant PUF array 124 as a repair resource for replacing “bad bits” in main PUF array 122. OKTA-1004, [0023]-[0024]. A POSITA would have understood that this functional role presupposes the redundant bits are themselves reliable because Wu contemplates device setup characterization of PUF cells under varying conditions (method 300) to identify unstable cells. OKTA-1004, [0035]-[0045], FIG. 3. A POSITA implementing the Wu-Müelich Combination would have thereby applied a similar device setup characterization principle to redundant PUF array 124 (e.g., using Wu’s method 300 process, Müelich’s multi-readout mechanism, or both) and selected the most reliable redundant cells for use as *additional output bits* in match entries.

(e) **Claim 11**

[11] The method of claim 9, wherein detecting an error in the PUF response (S2) comprises using an error correcting code.

120. The Wu-Müelich Combination renders claim 11 obvious because certain implementations would have detected PUF errors *using an error correcting code*, e.g., using Müelich’s iterative bitflip procedure incorporated into Wu’s modified step 440. OKTA-1004, [0058]-[0072]; OKTA-1005, 1, 3, 4. Such implementations would have provided an evaluation-time workflow that generates a “final raw PUF value” and then processes that word using “error correction logic 140” at modified step 440, which Wu permits use of “any known error correction

technique.” OKTA-1004, [0025], [0064]. A POSITA would have understood and found obvious that implementing the error-correction stage of modified step 440 using Muelich’s iterative bitflip procedure entails use of an ECC (e.g., an LDPC code defined by a parity-check matrix H) to detect whether the final raw PUF value contains an error condition. OKTA-1005, 1, 3, 4.

121. As discussed in [6.2] above, error correction in PUF-based key generators in Muelich may be improved by constructing an LDPC code (via a parity-check matrix H) and decoding a newly-generated PUF response r using syndrome weight criterion. OKTA-1005, 1, 3, 4. In Muelich’s iterative bitflip procedure (e.g., Algorithm 1), a syndrome weight ($w = WT(r)$) is computed for a received word r and evaluated. OKTA-1005, 5.

122. A POSITA would have understood that syndrome weight evaluation is an example of an ECC-based detection operation. For example, if the syndrome weight is equal to zero (e.g., $w = 0$), the word is treated as a valid codeword (no error), whereas if the criterion is not satisfied (e.g., $w \neq 0$), the received word includes an error condition (i.e., an “error pattern” e) and therefore an error is detected in the PUF response. OKTA-1005, 5. Only upon such detection does the decoding routine proceed to corrective operations (e.g., evaluating candidate bit flips), confirming that the initial syndrome weight computation is the step by which an error is detected using the ECC. OKTA-1005, 1, 3, 4.

(f) Claim 14

123. Claim 14 is substantially similar to claim 6, reciting generic preamble language identifying conventional computer components. Accordingly, the Wu-Müelich Combination renders the limitations of claim 14 obvious in a similar manner as explained above for corresponding claim limitations listed in the table below. In addition, Wu plainly provides the active electronic primitives, PUF circuits, processor, and memory of [14.pre]. *See, e.g.*, OKTA-1004, [0010], [0015]-[0018], [0020]-[0030], FIG. 1.

Limitation	Corresponding Limitation
[14.pre]	[6.pre]
[14.1]	[6.1]
[14.2]	[6.2]
[14.3]	[6.3]
[14.4]	[6.4]
[14.5]	[6.5]
[14.6]	[6.6]

B. GROUND 1B—Claim 10 is obvious in view of Wu, Müelich, and Tuyls

1. Combination of Wu, Müelich, and Tuyls

124. As discussed above, Wu contemplates an evaluation-time workflow in which a PUF response is generated and then processed by an “error correction logic”

stage (step 440 of method 400 depicted in Figure 4). OKTA-1004, [0058], [0064], FIG. 4. Method 400 is performed “during evaluation time,” e.g., by a customer, including actions performed by the internal operation of IC 100. *Id.*, [0058]. After reading/evaluating the PUF cell array (step 410) and performing a list-driven replacement of identified “noisy” (bad) bits using redundant bits (step 424), a “final raw PUF value” is then “processed by error correction logic 140” at step 440. *Id.*, FIG. 4, [0063]-[0064]; *above* §IX.A.1.

125. Wu itself recognizes that it is desirable to enable “post-manufacturing (e.g., during customer use) identification and replacement of bad bits,” for example, to “counter device aging effects.” *Id.*, [0073]. As devices are used over an extended period of time, they naturally encounter wear and thermal cycles that can affect PUF primitives. Wu also leaves the choice of the step-440 “error correction logic” open, stating that “[a]ny known error correction technique may be used,” and describes storage of error-correction data in non-volatile memory (NVM) for later use during operation. *Id.*, [0025]. Thus, a POSITA would have sought to supplement the error-correction stage of step 440 with well-known PUF-oriented diagnostic/error-correction techniques that iteratively identify/correct erroneous bit positions in the post-manufacturing instances contemplated by Wu. A POSITA would have been driven to implement post-manufacturing error correction due to the well-established need for such measures. For example, Schrijen-886 discloses that “generation of the

response pattern in a semiconductor memory may be sensitive to degradation after repeated applications of the activation signal,” and subsequently proposes writing “anti-aging data” to memory that documents the states of bits before they degrade. OKTA-1023, ¶¶[0153]-[0161]; *see* OKTA-1025 (“As the PUF circuits age, the PUF values generated by the PUF circuits may drift away from the enrollment values”); *see also* OKTA-1026 (“PUFs are subjected to a set of physical parameters...that cause the output of a PUF to change over time” and “this varying output can result in an increased and undesirable error rate”). Given that post-manufacturing error correction was well known, and that the issues with PUG degradation were well documented, a POSITA would have naturally sought to implement some form of post-manufacturing error correction (as suggested by Wu itself). OKTA-1004, [0073].

126. Like Muelich, Tuyls discloses techniques for determining when errors have occurred in a PUF. OKTA-1005, 1, 3-4; OKTA-1009, [0008]-[0021], [0095]-[0099]. Specifically, Tuyls’ “authentication unit 162” computes a “first hash” and compares the computed first hash with a “stored first hash” and determines if the two hash values match. OKTA-1009, [0100]-[0114]. If the hash values differ, it is determined that an error has occurred in the data. *Id.* A POSITA would have combined Tuyls’ additional error detection techniques into the Wu-Muelich Combination for the following reasons.

127. **First**, Tuyls discloses that its techniques are computationally more efficient than “large error correcting codes.” OKTA-1009, [0016]. Tuyls’ techniques would have thus been an efficient way to implement Wu’s described “error correction logic” on a portable electronic device where computation resources were limited (e.g., a mobile phone). OKTA-1004, [0058], [0064], FIG. 4; OKTA-1009, [0017]-[0018] (disclosing that the memory may be one “a mobile electronic device, such as a mobile phone”). The need to conserve computational resources in mobile computing environments was one that was widely recognized around the time of the Critical Date. Hashing techniques were also well known long before the Critical Date. For example, the Secure Hash Algorithm (SHA) set of standards all used hashing techniques to provide authentication in commonly used internet communication protocols, such as Transport Layer Security (TLS)/Secure Sockets Layer (SSL) and IPsec.

128. **Second**, Tuyls discloses that its techniques can identify faulty data faster than traditional error correcting codes. OKTA-1009, [0016] (“By computing multiple hashes over overlapping data, faulty memory cells may be found even more quickly.”). As with computational efficiency, a POSITA would have recognized that a more time-efficient method of error detection would reduce computational requirements. OKTA-1004, [0058], [0064], FIG. 4; OKTA-1009, [0017]-[0018].

129. A POSITA would have also had a reasonable expectation of success in

implementing Tuyls' additional error detection techniques into the Wu-Müelich Combination. Wu identifies step 440 as the evaluation-time stage for applying error correction logic to the final raw PUF value and expressly permits the use of "any known error correction technique," including with error-correction data stored in NVM for later use. OKTA-1004 [0025], [0058], [0063], [0064]. Tuyls provides a computationally efficient method of detecting errors in data based on comparing a hash of the data with a reference hash. OKTA-1009, [0016], [0100]-[0114]. Incorporating Tuyls' error detection techniques into Wu's step-440 error-correction logic and Müelich's iterative bitflip procedure would have been implementable through routine engineering choices. This modification would have required only routine algorithmic modifications that would have been well within the knowledge and skill of a POSITA. *Above*, §IV.

2. Analysis

(a) Claim 10

[10.1] The method of claim 6, wherein said device is configured for access to a memory storing a reference hash value of said response bits,

130. Tuyls discloses a "validation unit 164" that compares a computed "first hash" to a "stored first hash" that is saved in memory (***said device is configured for access to a memory storing a reference hash value of said response bits***). OKTA-1009, [0100]-[0114]. As discussed below, the validation unit 164 references the stored first hash for comparison against a computed first hash when determining if

errors have occurred in the data represented by the computed first hash. *Id.* In the combination, this would have involved checking if there is an error before iterative bit flipping identifying the error using Muelich’s iterative bitflip procedure. *Above*, §IX.B.1(d).

[10.2] detecting an error in the PUF response (S2) comprises: generating a hash value by applying a hash function on said bits of the PUF response, comparing said hash value with said stored reference hash value.

131. Tuyls also discloses an “authentication unit 162” that computes a “first hash” over a set of data (*generating a hash value by applying a hash function on said bits of the PUF response*). OKTA-1009, [0100]. Next, a “validation unit 164” compares the computed first hash with a “stored first hash” and determines if the two hash values match (*comparing said hash value with said stored reference hash value*). OKTA-1009, [0101]-[0114]. If the hash values differ, it is determined that an error has occurred in the data (*detecting an error in the PUF response ...*). *Id.*

C. GROUND 1C—Claims 1-3, 5, 12, and 13 are obvious in view of Wu, Muelich, and Yang

1. Combination of Wu, Muelich, and Yang

132. A POSITA would have known that PUF error correcting techniques presented security vulnerabilities, as acknowledged by Wu and Muelich. OKTA-1004, [0023] (“it is desirable for the testing process to not reveal information ... for that would compromise the security benefits provided to the customer”); OKTA-1005, 3 (“since the procedure is designed such that knowing the helper data does not

reveal information about r_I that can be exploited by an adversary”). For example, PUF error correction presents various “side-channels” where information about the PUF can be exposed from monitoring how the PUF was repaired. OKTA-1007, [0017]-[0019]. A POSITA would have been aware that power and time side channels could have been used to infer PUF characteristics without directly accessing the PUF itself.

133. Schrijen illustrates the existence of these vulnerabilities and describes that “power consumption” and “time consumption” can be monitored during the repair process to infer information about the PUF. OKTA-1007, [0017]-[0019], [0155]. These are similar vulnerabilities as those described in the ’359 patent. *See* OKTA-1001, 5:9-12. Other PUF publications similarly acknowledge such vulnerabilities. OKTA-1008, 1-2; OKTA-1009, [0021]-[0025]. For example, an adversary could time a PUF repair operation and infer what bits of the PUF were replaced based on the elapsed repair time, or alternatively, could measure power consumption of the circuit to determine when repair operations were occurring. *Id.* Because side channels were well known PUF vulnerabilities, a POSITA would have known to mitigate the vulnerabilities presented by side channels (as evidenced by the above examples). OKTA-1007, [0017]-[0019], [0155]; OKTA-1009, [0021]-[0025]. In addition to Schrijen, Yao’s disclosure also demonstrates the existence of side channel vulnerabilities in PUFs before the Critical Date. OKTA-1033, [0001]-

[0009] (“However, using various side-channel attacks, such as optical emissions or power analysis, an attacker may be able to learn the PUF response and gain access to protected information.”).

134. In view of these well-known vulnerabilities, a POSITA would have been motivated to implement the error correcting techniques of the Wu-Müelich combination in a random and non-deterministic manner. *Id.* There was a well-known need to ensure that PUF error correction was performed without exposing PUF information to side-channels. OKTA-1007, [0017]-[0019], [0155]; OKTA-1008, 1-2; OKTA-1009, [0021]-[0025]. And, in the context of the Wu-Müelich combination, a POSITA would have recognized and found obvious that there were only a finite number of ways that bits could be flipped according to Müelich – either in-order, or randomly. *See above*, §IX.A.1. From a security-focused perspective, the latter would have been preferable to a POSITA that was mindful of the vulnerabilities presented by side-channels, and specifically, because an in-order repair process could have been used by an attacker to identify the status of specific PUF bits as repairs occurred. OKTA-1007, [0017]-[0019], [0155]; OKTA-1008, 1-2; OKTA-1009, [0021]-[0025]. The deliberate injection of randomness into PUF applications is also supported by Rostami, which discloses inserting “random bits” into a PUF response to obfuscate the response from attackers. OKTA-1027, 14:15-17:14, FIGS. 6A-6B.

135. A POSITA mindful of a security-inspired need to implement the Wu-Müelich combination's techniques in a random and non-deterministic manner would have discovered Yang and realized it as a potential solution. OKTA-1006, [0004], [0010]-[0019], FIG. 1. Yang's process of randomly selecting bits from a PUF would have provided an adequate level of randomness that would have protected Wu-Müelich combination from side-channel attacks, as an attacker monitoring repair time or power consumption would have no way of correlating repairs to a particular PUF bit. *Id.*; OKTA-1007, [0017]-[0019], [0155]; OKTA-1008, 1-2; OKTA-1009, [0021]-[0025]. For example, randomly selecting bits of the PUF for flipping in Wu-Müelich combination would have obfuscated the overall repair time, as an attacker would have no way of knowing how many PUF bits were repaired during any given repair operation. *Id.* Further, an adversary monitoring power consumption may have been able to infer how many repairs were performed, but they would have been unable to correlate those repairs to a particular PUF bit. *Id.* In this way, even if an adversary monitored side channels hoping to observe PUF characteristics, no useful information could be determined.

136. A POSITA would have also had a reasonable expectation of success in implementing the combination of Wu-Müelich combination and Yang. Like Wu and Müelich, Yang is concerned with information security techniques using and relating to PUFs. OKTA-1004, [0010], [0017]-[0018]; OKTA-1005, 1-3; OKTA-

1006, [0004], [0010]-[0019]. Further, Yang's techniques are implemented in a memory device including a PUF – structure that is also disclosed by Wu in the form of its PUF Unit 110 that includes NVM 170. OKTA-1006, [0004], [0010]-[0019]; OKTA-1004, [0015]-[0017], [0028]-[0029]. Finally, Wu envisions that “modifications and alterations” of its techniques can be performed “while retaining the teachings of the invention” (e.g., the modifications of Yang that would have increased the security of Wu's techniques). OKTA-1004, [0028]. Thus, the modification of Wu-Müelich combination and Yang is nothing more than (i) the application of known techniques (e.g., Yang's random selection of PUF bits) to a known structure (e.g., the PUF repair system of the Wu-Müelich combination) to yield predictable results; (ii) the use of known techniques to improve similar devices in similar ways (e.g., improving the security of a PUF repair system with repair randomization); and (iii) applying a known technique (e.g., Yang's random selection of PUF bits) to a known device ready for improvement to yield predictable results (e.g., implementing repair randomization to mitigate side-channel vulnerabilities).

2. Analysis

(a) Claim 1

137. Except as discussed below, Wu-Müelich-Yang renders the limitations of claim 1 obvious in a similar manner as explained above for corresponding claim limitations listed in the table below.

Limitation	Corresponding Limitation
[1.pre]	[6.pre]
[1.1]	[6.1]
[1.2]	[6.2]
[1.4]	[6.4]
[1.5]	[6.5]
[1.6]	[6.6]

[1.3] determining erroneous response bits of the PUF response (S3) comprising replacing one by one, in a random order, each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is s [sic] corrected,

138. See above, [6.3]. The Wu-Müelich-Yang combination would have used Müelich's iterative bitflip procedure at modified step 440 to tentatively ***replac[e]***, each bit of the final raw PUF value (***each bit of the PUF response***) with its complement (***arbitrary value*** for a binary bit), and then determined whether the error is corrected by evaluating a syndrome weight of the resulting bitstring (***determining if the error in the PUF response is...corrected***). OKTA-1005, 5.

139. Further, it would have been obvious for the bit replacement to be performed in ***a random order***, consistent with the suggestions in Yang, for the purpose of frustrating side-channel attacks. OKTA-1006, [0014]-[0015], FIG. 1; OKTA-1007, [0017]-[0019], [0155]; OKTA-1008, 1-2; OKTA-1009, [0021]-

[0025]. Specifically, Yang discloses a controller that “extract[s] a random bit sequence from [a] random bit pool in [a] PUF unit 105,” which a POSITA would have used with Muelich’s Algorithm 1 to determine the order in which bits are swept by the algorithm. OKTA-1006, [0014]-[0015], FIG. 1. In so doing, Muelich’s Algorithm 1 would have swept bit positions in a in a random order (rather than a default positional order discussed for [6.3] above), thereby performing per-bit replacement *in a random order*. *Id.*

(b) Claims 2, 3, 5

140. Wu-Muelich-Yang renders the limitations of claims 2, 3, and 5 obvious in a similar manner as explained above for corresponding claim limitations listed in the table below.

Limitation	Corresponding Limitation
[2.1]	[8.1]
[2.2]	[8.2]
[3]	[9]
[5]	[11]

(c) Claim 12

141. Claim 12 is substantially similar to claims 1 and 6, reciting a similar series of steps with generic preamble language identifying conventional computer components. Accordingly, the Wu-Muelich-Yang Combination renders the

limitations of claim 12 obvious in a similar manner as explained above for corresponding claim limitations listed in the table below. In addition, Wu plainly provides the active electronic primitives, PUF circuits, processor, and memory of [12.pre]. *See, e.g.*, OKTA-1004, [0010], [0015]-[0018], [0020]-[0030], FIG. 1.

Limitation	Corresponding Limitation
[12.pre]	[1.pre], [6.pre]
[12.1]	[1.1], [6.1]
[12.2]	[1.2], [6.2]
[12.3]	[1.3], [6.3]
[12.4]	[1.4], [6.4]
[12.5]	[1.5], [6.5]
[12.6]	[1.6], [6.6]

D. GROUND 1D—Claim 4 is obvious in view of Wu, Muelich, Yang, and Tuyls

1. Combination of Wu, Muelich, Yang, Tuyls

142. A POSITA would have combined both Yang and Tuyls into Wu and Muelich for the reasons discussed above with respect to each individual reference. *Above*, §§IX.B.1, IX.C.1.

2. Analysis

(a) Claim 4

[4.1] The method of claim 1, wherein said device is configured for access to a memory storing a reference hash value of said response bits,

143. See Ground 1B, [10.1], *above*.

[4.2] detecting an error in the PUF response (S2) comprises: generating a hash value by applying a hash function on said bits of the PUF response, comparing said hash value with said stored reference hash value.

144. See Ground 1B, [10.2], *above*.

E. GROUND 2A—Claims 6-9, 11, and 14 are obvious in view of Wu, Muelich, and Chan⁵

1. Combination of Wu, Muelich, and Chan

145. As discussed above in Ground 1A, a POSITA would have been motivated to apply Muelich’s teachings on iterative bit-flipping error correction to identify and correct erroneous bits in a PUF response (e.g., Wu’s “final raw PUF

⁵ Grounds 2A and 2C apply Chan’s teachings to demonstrate obviousness under an interpretation of the phrase “replace, one by one...each bit of the PUF response” as requiring replacement of all bits of the “PUF response.” As explained in §VI.C.1 (and applied in Grounds 1A and 1C), the correct ordinary and customary meaning of this phrase is that the process requires replacing individual bit values until such replacement corrects the PUF response error and, thus, reveals an erroneous bit. OKTA-1001, 2:51-52, 3:1-3, 5:1-8.

value”). *See above*, §IX.A.1. A POSITA would have been motivated to make this combination for multiple reasons. *Id.*

146. **First**, Chan teaches that bit-flipping algorithms were vulnerable to converging on spurious local extrema, and a POSITA would have been motivated to introduce taboo numbers to prevent the combination bit-flipping algorithm from converging on an incorrect local solution (and thus wasting computing resources and time). OKTA-1011, 1-3. Computational power is a precious resource in mobile computing environments, where PUFs are traditionally used. Wadayama demonstrates that as early as 2008, POSITAs were implementing measures to mitigate the tendency of LDPC algorithms to converge on spurious local extrema (in the case of Wadayama, using a larger “step size”). OKTA-1031, 1-6. Pavithra is another example from 2015 where a “multi bit flipping” algorithm was proposed to escape local extrema in GDBF codes. OKTA-1032, 1-5.

147. **Second**, Chan’s techniques improve performance of the underlying bit-flipping algorithm while maintaining simplicity (and are thus compatible with PUF environments that may have limited computational resources). *Id.* Although performance increases are desirable by themselves, Chan’s techniques offer performance with efficiency – something that would have been particularly desirable in a mobile computing environment.

148. A POSITA would have also had a reasonable expectation of success in

implementing the combination of Chan and the Wu-Müelich combination. Müelich describes improving error identification and correction using iterative, bit-level correction in a LDPC algorithm. This is a similar category of algorithms described by Chan. OKTA-1005, 4-5; OKTA-1011, 1-3; *see above* §III.A.1(b). Accordingly, incorporating Chan's taboo numbers into Wu-Müelich's error correction techniques would have been implementable through routine engineering choices, as the Wu-Müelich combination is already structured in a manner that is explicitly described by Chan. These low level algorithmic modifications would have been well within the general knowledge and skill of a POSITA. *Above*, §IV. Indeed, Cui demonstrates the prevalence of taboo lists ("tabu-list," as described by Cui) to "help [a] decoder escape from trapping sets" that are indicative of local extrema. OKTA-1029, 1-5. Zhang also discloses the use of a "tabu-list" in LDPC coding. OKTA-1028, 1-5.

2. Analysis

(a) Claims 6-9, 11, and 14

149. Except as discussed below, Wu-Müelich-Chan renders the limitations of claims 6-9, 11, and 14 obvious in a similar manner as explained above in Ground 1A. *Above*, §IX.A.2. As previously noted above, Ground 2A applies Chan's teachings to demonstrate obviousness under an interpretation of the phrase "replace, one by one...each bit of the PUF response" as requiring replacement of all bits of

the “PUF response.” As explained in §VI.C (and applied in Ground 1A), the correct ordinary and customary meaning of this phrase is that the process requires replacing individual bit values until such replacement corrects the error in the PUF response and, thus, reveals an erroneous bit. OKTA-1001, 2:51-52, 3:1-3, 5:1-8. This interpretation is consistent with the ’359 patent’s emphasis on efficiency. *Id.*

[6.3] determining erroneous response bits of the PUF response (S3) comprising replacing one by one, in a [predetermined] [sic] order, each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is corrected,

150. *See above*, Ground 1A, [6.3]. The Wu-Müelich-Chan combination would have used Müelich’s iterative bitflip procedure at modified step 440 to tentatively ***replace*** each bit of the final raw PUF value (***each bit of the PUF response***) with its complement (***arbitrary value*** for a binary bit), and then determined whether the error is corrected by evaluating the syndrome weight of the resulting bitstring (***determining if the error in the PUF response is...corrected***).

151. For example, Chan discloses a bit-flipping algorithm with a “taboo” function that “avoids repeated flipping of the same bit” by assigning a “taboo number” to each bit when it is flipped, which only allows subsequent flipping of that bit “if the current iteration number is equal to or less than the taboo number of that variable.” OKTA-1011, 1-3. As discussed above in the combination, a POSITA would have been motivated to combine Chan with Wu-Müelich to assign taboo

numbers to bit positions flipped by the Muelich's iterative bitflip procedure, such that the bit at a given bit position could not be flipped again until Chan's taboo conditions were satisfied. OKTA-1011, 1-3; *see above* §IX.E.1.

152. Further, a POSITA would have recognized that, in scenarios involving dense or complex error patterns, use of Chan's taboo function would result in a process in which *each bit of the PUF response* (i.e., each of the n bit positions of the final raw PUF value) is replaced. OKTA-1011, 1-3. This is achievable because Chan's taboo function prevents the bitflip procedure from locally oscillating on a single bit position, thereby forcing it to traverse the sequence of bit positions and address unsatisfied constraints distributed across the final raw PUF value. OKTA-1011, 1-3. While Chan notes that taboos may be ignored if all high-priority candidates are exhausted, a POSITA would recognize that in a high-error final raw PUF value, valid candidates for flipping (unsatisfied constraints) would exist across the sequence. In such a "worst-case" convergence scenario, Chan's taboo function ensures that Muelich's iterative bitflip procedure iteratively targets and flips these widely distributed bit positions, extending to the entire sequence under certain conditions. *Id.*

F. GROUND 2B— Claim 10 is obvious in view of Wu, Muelich, Chan, Tuyls

1. Combination of Wu, Muelich, Chan, and Tuyls

153. A POSITA would have combined both Chan and Tuyls into Wu and Muelich for the reasons discussed above with respect to each individual reference.

Above, §§IX.B.1, IX.E.1.

2. Analysis

(a) Claim 10

[10.1] The method of claim 6, wherein said device is configured for access to a memory storing a reference hash value of said response bits,

154. *See* Ground 1B, [10.1], *above*.

[10.2] detecting an error in the PUF response (S2) comprises: generating a hash value by applying a hash function on said bits of the PUF response, comparing said hash value with said stored reference hash value.

155. *See* Ground 1B, [10.2], *above*.

G. GROUND 2C—Claims 1-3, 5, 12, and 13 are obvious in view of Wu, Muelich, Chan, and Sundararajan⁶

1. Combination of Wu, Muelich, Chan, and Sundararajan

156. As discussed above in Ground 1A, a POSITA would have been motivated to apply Muelich’s teachings on iterative bit-flipping error correction to

⁶ Ground 2C applies Sundararajan as an alternative basis to demonstrate obviousness with respect to the claimed phrase “random order” within Elements [1.3] and [12.3].

identify and correct erroneous bits in a PUF response (e.g., Wu’s “final raw PUF value”). *Above*, §IX.A.1. Additionally, a POSITA would have been motivated to combine Sundararajan with Wu, Muelich, and Chan to apply a random perturbation to the Wu-Muelich-Chan bit-flipping algorithm to achieve a benefit described in Sundararajan, e.g., by improving performance of bit-flipping algorithms for “low-complexity” applications. OKTA-1010, 1-4. A POSITA would have been further motivated to make this combination for multiple reasons. *Id.*

157. Like Chan, Sundararajan also teaches that bit-flipping algorithms were vulnerable to converging on spurious local maxima, and a POSITA additionally would have been motivated to introduce randomness into the combination’s iterative bit-flipping to prevent the algorithm from converging on an incorrect local solution (and thus wasting computing resources and time). OKTA-1010, 1-4. Further, Sundararajan’s techniques improve performance of the underlying bit-flipping algorithm and are described as “low-complexity” (and thus compatible with PUF environments that may have limited computational resources). *Id.* Generally, algorithmic improvements that are low complexity and result in the reduction of required computing resources are desirable in mobile computing environments where such resources are limited (e.g., cell phone or smartphones).

158. A POSITA would have also had a reasonable expectation of success in implementing the combination of Sundararajan and the Wu-Muelich-Chan

Combination. Muelich describes improving error identification and correction using iterative, bit-level correction in a LDPC algorithm. OKTA-1005, 4-5. This is a similar category of algorithms described by Sundararajan. OKTA-1010, 1-4; *above*, §III.A.1(b). Accordingly, incorporating Sundararajan's random perturbations into Wu-Muelich-Chan's error correction techniques would have been implementable through routine engineering choices, as the Wu-Muelich-Chan combination is already structured in a manner that is explicitly described by Sundararajan. This is nothing more than the implementation of simple algorithmic improvements that would have been well within the knowledge and skill of a POSITA. *Above*, §IV.

2. Analysis

(a) Claims 1-3, 5, 12, and 13

159. Except as discussed below, Wu-Muelich-Chan-Sundararajan renders the limitations of claims 1-3, 5, 12, and 13 obvious in a similar manner as explained above. *Above*, §IX.C. As previously noted, this ground applies Chan's teachings to demonstrate obviousness under an interpretation of the phrase "replace, one by one...each bit of the PUF response" as requiring replacement of all bits of the "PUF response." As explained above in §VI.C (and applied in the Wu-Muelich-Yang ground above), the correct ordinary and customary meaning of this phrase is that the process requires replacing individual bit values until such replacement corrects the PUF response error and, thus, reveals an erroneous bit. OKTA-1001, 2:51-52, 3:1-

3, 5:1-8. This is consistent with the teachings of the '359 patent, which emphasizes computational efficiency. *Id.*

[1.3] determining erroneous response bits of the PUF response (S3) comprising replacing one by one, in a random order, each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is s [sic] corrected,

160. See above, Grounds 1A, 2A, [6.3]. The Wu-Müelich-Chan-Sundararajan combination would have used Müelich's iterative bitflip procedure at modified step 440 to tentatively ***replace*** each bit of the final raw PUF value (***each bit of the PUF response***) with its complement (***arbitrary value*** for a binary bit), and then determined whether the error is corrected by evaluating the syndrome weight of the resulting bitstring (***determining if the error in the PUF response is...corrected***). Further, it would have been obvious for the bit replacement to be performed in ***a random order***, consistent with the suggestions in Sundararajan, for the purpose of improving efficiency and avoiding local extrema. OKTA-1010, 1-4. This would have improved computational efficiency, which is desirable in mobile computing environments where PUFs are frequently used.

161. For example, Sundararajan's "Noisy GDBF" introduces a "random perturbation" (vector "z" of "identically distributed Gaussian random variables") that randomizes bit-flipping decisions made by the bitflip procedure. *Id.* As discussed above, a POSITA would have found it obvious that, in the context of the

Wu-Müelich-Chan iterative bitflip procedure, tentative replacement using such a random perturbation involves tentative replacement of bit positions of the final raw PUF value *in a random order*. OKTA-1010, 1-4. Consistent with Sundararajan, a POSITA would have understood that a well-known configuration would have involved applying a random perturbation to the Wu-Müelich-Chan iterative bitflip procedure, such that the selection of a bit position to flip is influenced by a random perturbation, thereby introducing a stochastic element to the gradient descent process. OKTA-1010, 1-4; *above*, §IX.G.1.

162. A POSITA would have found it obvious that, in the context of the Wu-Müelich-Chan iterative bitflip procedure, tentative replacement of bit positions using such a random perturbation (as described in Sundararajan) involves, in some circumstances, a randomized sequence of bit position replacements (*random order*). OKTA-1010, 1-4. Specifically, the introduction of random noise to the selection metric renders the sequence in which bit positions are replaced non-deterministic, which would have been understood by a POSITA to represent bit replacement of bit positions in a *random order* to avoid local minima. *Id.* Because a random perturbation is added to the decision metric for every bit position in every iteration, the relative ranking of candidate bit positions is shuffled by the random noise values, causing bit positions to be selected in a sequence that differs from a deterministic gradient. This stochastic reordering ensures that the specific sequence of bit

replacements is not fixed by input data and follows a random trajectory through the search space. *Id.* Doing so avoids spurious extrema which often emerge, as recognized by Sundararajan. OKTA-1010, 1.

H. GROUND 2D—Claim 4 of the '359 patent is obvious in view of Wu, Muelich, Chan, Sundararajan, and Tuyls

1. Combination of Wu, Muelich, Chan, Sundararajan, and Tuyls

163. A POSITA would have combined Chan, Sundararajan, and Tuyls into Wu and Muelich for the reasons discussed above with respect to each individual reference. *Above*, §§IX.B.1, IX.E.1, IX.G.1.

2. Analysis

(a) Claim 4

[4.1] The method of claim 1, wherein said device is configured for access to a memory storing a reference hash value of said response bits,

164. *See* Ground 1B, [10.1], *above*.

[4.2] detecting an error in the PUF response (S2) comprises: generating a hash value by applying a hash function on said bits of the PUF response, comparing said hash value with said stored reference hash value.

165. *See* Ground 1B, [10.2], *above*.

I. GROUND 3—Claims 1-14 of the '359 patent would lack written description support if interpreted to avoid the prior art

1. Legal Principles

166. I have been informed by counsel that, to meet the written description requirements of 35 U.S.C. §112(a), the specification must describe the claimed

invention sufficiently to convey to a person of skill in the art that the patentee had possession of the claimed invention at the time of the application (i.e., that the patentee invented what is claimed). This requires that the written description of the specification actually or inherently discloses each claim element.

167. I understand that the question is not whether a claimed invention is an obvious variant of that which is disclosed in the specification, but rather, whether POSITA could have clearly concluded that the inventor invented the claimed invention as of the filing date sought. I also understand that a proper written description analysis must take each claim as an integrated whole rather than as a collection of independent limitations.

2. Review of the '359 patent

168. Independent claims 1, 6, 12, and 14 each require “determining erroneous response bits of the PUF response” by “replacing one by one...each bit of the PUF response by an arbitrary value and determining if the error in the PUF response is corrected.” As discussed, under the ordinary and customary meaning in view of the specification, these limitations do *not* require, in a single error-detection pass, [i] determining multiple erroneous response bits or [ii] replacing the value of each and every single bit of the PUF response. *See above*, §VI.C; OKTA-1001, 2:51-52, 3:1-3, 5:1-8, 6:29-36, Fig. 4. Thus, the prior art combinations in Grounds 1A–2D render these limitations obvious through a PUF repair architecture that

improves Wu's evaluation-time workflow by implementing error correction using an iterative bit-flipping technique that evaluates candidate single-bit replacements. *See above*, §§IX.A–D.

169. If an alternative interpretation sought to distinguish these limitations from the prior art, the specification does not provide descriptions sufficiently to convey to a person of skill in the art that the patentee had possession of the claimed configuration as interpreted according to the alternative interpretation. As explained below, the '359 patent's specification does not demonstrate possession of a PUF response repair technique involving either an exhaustive replacement of each and every bit or the simultaneous discovery of multiple erroneous bits using a single bit-by-bit replacement scheme.

3. Analysis

(a) Determining multiple erroneous response bits in a single error-detection pass

170. As discussed above, the '359 patent's specification juxtaposes two alternative bit replacement schemes. *Above*, §VI.C. First, the specification explains that “the PUF circuits may ***replace one by one each bit*** of the PUF response by an arbitrary value and determine if the error in the PUF response is corrected by comparing the hash of the modified PUF response with the reference hash value.” OKTA-1001, 5:1-5. Next, when discussing a distinct hypothetical where “multiple

bits are erroneous in the PUF response,” the specification says that “***multiple bits may be replaced by their complement at the same time*** in order to determine all of the erroneous bits of the PUF response.” *Id.*, 5:5-8.

171. In summary, the inventors possessed two distinct and alternative bit replacement schemes. The first one involves replacing individual bits to discover individual bit errors and another that replaces multiple bits simultaneously to discover multiple bit errors. There is no explicit, implicit, or inherent disclosure that the claimed singular bit-by-bit replacement scheme can simultaneously discover multiple bit errors in the PUF response. Accordingly, if the claims were interpreted in this fashion, the specification lack the express or inherent disclosure that would have sufficiently conveyed to a person of skill in the art that the patentee had possession of the claimed configuration as interpreted. I understand that the mixing and matching of isolated disclosures noted above cannot be used to show such possession.

(b) Replacing each and every single bit of the PUF response in a single error-detection pass

172. As discussed above, the '359 patent's specification fails to disclose replacing the value of each and every single bit of the PUF response in a single error-detection pass. *See, e.g.*, OKTA-1001, 6:29-36, FIG. 4; *above*, §VI.C. Instead, the specification describes that the disclosed error-detection process concludes (i.e.,

makes no further bit replacements) when the erroneous bit (or bits) is (or are) identified and matched with a corresponding additional output bit. *Id.* An exhaustive bit-flipping pass is inconsistent with the '359 patent's stated goals; specifically, to "*minimize the computations* required to find the erroneous bits of the PUF response" and to "*reduc[e] the number* of response bits to be replaced." OKTA-1001, 2:51-52, 3:1-3. I understand that, even if replacing each and every single bit were an obvious variant of the '359 patent's disclosure (I do not believe it is), that still would not be sufficient to convey to a person of skill in the art that the patentee had possession of the obvious variant.

J. GROUND 4— Claims 1-14 of the '359 patent would be indefinite if interpreted to avoid the prior art

1. Legal Principles

173. I understand that a claim is indefinite if its language, read in light of the specification and prosecution history, fails to inform a POSITA about the scope of the claimed configuration with reasonable certainty. I further understand that claim scope is not reasonably certain, for example, when the language might mean several different things and no informed and confident choice is available among the contending definitions.

2. Review of the '359 patent

174. Independent claims 1, 6, 12, and 14 each require "replacing one by one...each bit of the PUF response by an arbitrary value and determining if the

error...is corrected.” As discussed above, under the ordinary and customary meaning in view of the specification, these limitations do *not* require, in a single error-detection pass, replacing the value of each and every single bit of the PUF response. OKTA-1001, 2:51-52, 3:1-3, 5:1-8, 6:29-36, FIG. 4; *above*, §VI.C. The prior art combinations in Grounds 1A–2D render this limitation obvious through a PUF repair architecture that improves Wu’s evaluation-time workflow by implementing error correction using Muelich’s iterative bitflip procedure that evaluates candidate single-bit replacements. *Above*, §§IX.A–D. If the claims were interpreted to distinguish this limitation from the prior art, a POSITA would have been unable to ascertain claim scope with reasonable clarity. Specifically, because the specification supports the interpretation discussed above at least as strongly as any alternative, a POSITA would have been left with no objective basis to confidently choose between the competing definitions.

3. Analysis

175. The specification explicitly defines the inventors’ purpose as “minimiz[ing] the computations required to find the erroneous bits” and “reducing the number of response bits to be replaced.” OKTA-1001, 2:51-52, 3:1-3. Consistent with the interpretation discussed above, this expressly stated purpose of promoting efficiency is served by an iterative process that stops when the erroneous PUF response bit is found.

176. Any alternative interpretation that involves indiscriminately flipping each and every single bit of the PUF response demands a process that is deliberately inefficient. A POSITA would have recognized that it requires the system to continue testing every remaining bit even after the error has been resolved. To adopt this interpretation, the specification's core objective of efficiency would need to be discarded in favor of pointless additional testing. This interpretation requires flipping each and every single bit of the PUF response, and further requires disregarding the '359 patent's preferred embodiment. As discussed, the only provided flowchart and associated description make clear that the error-detection process concludes when the erroneous bit is identified and matched with a corresponding additional output bit. *Above*, §VI.C; OKTA-1001, 6:29-36, FIG. 4.

177. If the specification's guidance is dismissed, a POSITA would have been left with only the words used in the claims. When read without the technical context provided by the specification, the phrase "replacing...each bit" can be understood to mean two conceptually distinct interpretations:: (1) replacing each bit in the PUF response until the error in the PUF response is corrected (my interpretation), or (2) replacing each and every single bit in the PUF response, regardless of when the PUF response error is corrected (which is nonsensical). Without the specification's context, the POSITA would have had no objective basis to choose between these two interpretations because the words used by the claims alone would not have

informed the POSITA to determine which of the two conceptually distinct interpretations was intended as the patentee.

X. ADDITIONAL REMARKS

178. I currently hold the opinions 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.

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

Dated: 7 May 2026

By: 
Patrick Gerard Traynor

Appendix CV

Patrick Gerard Traynor

Professor

Interim Department Chair in CISE

John and Mary Lou Dasburgh Preeminent Chair in Engineering

Department of Computer & Information Science & Engineering (CISE)

University of Florida

1889 Museum Rd,

Gainesville, FL 32611 USA

traynor@cise.ufl.edu

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

Table of Contents

EDUCATIONAL BACKGROUND	4
EMPLOYMENT HISTORY	4
CURRENT FIELDS OF INTEREST	4
I. TEACHING	6
A. Courses Taught	6
B. Continuing Education	6
C. Curriculum Development	6
D. Individual Student Guidance	7
E. Teaching Honors and Awards	12
II. RESEARCH AND CREATIVE SCHOLARSHIP	13
A. Thesis	13
B. Published Journal Papers (Refereed)	13
C. Published Books and Parts of Books	15
D. Edited Proceedings	15
E. Conference Presentations	15
E.1. Conference Papers with Proceedings (Refereed)	15
E.2. Conference Presentations with Proceedings (Non-Refereed)	23
E.3. Conference Presentations without Proceedings	23
F. Other	23
F.1. Submitted Journal Papers	23
F.2. Refereed Research Reports	23
F.3. Software	23
F.4. Published Papers (Non-Refereed)	23
F.5. Books in Preparation	24
F.6. Workshops and External Courses	24
G. Research Proposals and Grants (Principal Investigator)	24
H. Research Proposals and Grants (Contributor)	27
I. Research Honors and Awards	29
III. SERVICE	30
A. Professional Activities	30
A.1. Memberships and Activities in Professional Societies	30
A.2. Conference Committee Activities	30
B. On-Campus Committees	31
B.1. University of Florida	31
B.2. Georgia Tech	32
C. Special Assignments	32
D. Ph.D. Examining Committees	32
E. External Member of M.S. Examining Committee	36
F. Consulting and Advisory Appointments	36
G. Civic Activities	36
IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION	37
A. Honors and Awards	37
B. Invited Conference Session Chairmanships	37
C. Professional Registration	37
D. Patents	37
E. Editorial and Reviewer Work for Technical Journals and Publishers	38
F. Expert Witness Services	40
V. OTHER CONTRIBUTIONS	42
A. Seminar Presentations (Invited Papers and Talks at Meetings and Symposia)	42

B. Special Activities	46
---------------------------------	----

EDUCATIONAL BACKGROUND

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

EMPLOYMENT HISTORY

Title	Organization	Years
Interim Department Chair	University of Florida	July 2025–Present
Associate Chair for Research	University of Florida	August 2018–Present
Professor	University of Florida	August 2018–Present
Associate Professor	University of Florida	August 2014–July 2018
Associate Professor	Georgia Institute of Technology	March 2014–August 2014
Assistant Professor	Georgia Institute of Technology	2008–March 2014
Research Assistant	Pennsylvania State University	2004–2008
Teaching Assistant	Pennsylvania State University	2004

CURRENT FIELDS OF INTEREST

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

Recognizing this, my research focuses on three specific themes: (1) developing efficient techniques to allow telephony providers and customers to authenticate the origin of incoming calls; (2) measuring and

improving the security of emerging mobile financial systems and (3) efficient and strong privacy-preserving techniques for mobile devices. Additionally, I have significant expertise in fraud detection, particularly for payment systems.

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

I. TEACHING

A. Courses Taught

Semester/Year	Course Number & Title	Number of Students	Comments
Fall 2024	CNT 4007 Computer Networks 1	310	Revamped Course
Fall 2023	CIS 6930 Cellular and Mobile Network Security	19	New Topics
Fall 2022	CNT 5410 Computer and Network Security	75	New Topics
Fall 2021	CNT 5410 Computer and Network Security	45	New Topics
Fall 2019	CNT 5410 Computer and Network Security	28	New Topics
Fall 2018	CIS 6930 Cellular and Mobile Network Security	16	New Course
Fall 2017	CNT 5410 Computer and Network Security	27	New Topics
Fall 2016	CNT 5410 Computer and Network Security	60	New Topics
Spring 2016	CNT 5410 Computer and Network Security	13	New Topics
Spring 2015	CNT 5410 Computer and Network Security	12	New Topics
Fall 2014	CNT 5410 Computer and Network Security	30	New Course
Spring 2014	CS 6262 Network Security	55	New Projects
Fall 2013	CS 3251 Computer Networks I	73	Expanded Syllabus
Spring 2013	CS 6262 Network Security	65	All New Projects
	CS 8001 Information Security Seminar	20	New Speakers
Fall 2012	CS 8803 Cellular & Mobile Network Security	17	New Topics
	CS 8001 Information Security Seminar	20	New Speakers
Spring 2011	CS 8001 Information Security Seminar	20	New Speakers
Fall 2011	CS 6262 Network Security	27	Expanded Syllabus
	CS 8001 Information Security Seminar	35	New Speakers
Spring 2011	CS 3251 Computer Networks I	61	Expanded Syllabus
	CS 8001 Information Security Seminar	20	New Speakers
Fall 2010	CS 8803/4803 Cellular & Mobile Network Security	16	New Course
	CS 8001 Information Security Seminar	31	New Speakers
Fall 2009	CS 6262 Network Security	55	Expanded Syllabus
Spring 2009	CS 3251 Computer Networks I	45	Expanded Syllabus
Fall 2008	CS 8003 Destructive Research	10	New Course

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

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

B. Continuing Education

None.

C. Curriculum Development

University of Florida

CNT 4007 Computer Networks 1: *Fall 2024.* Provided the first major overhaul to this course in a number of years. While I have relied on the same book used by other faculty, I have created new homeworks, projects, and slides to better represent the current state of computer networks. I have also significantly expanded the discussion of security in this course.

CIS 6930 Cellular and Mobile Network Security: *Fall 2018, 2023.* Developed an entirely new course around security issues facing cellular and mobile networks. Students learned about wireless basics, spectrum issues, core network architectures (GSM, ISDN, IMS, SIP), air interfaces (2G-5G), mobility management, voice-driven systems, authentication, mobile phone operating systems (Android, iPhone), Android security, congestion and denial of service, privacy and eavesdropping. Students also complete a research project and aim towards publishing this work at a major venue. My aim is for this class to become part of the regular offering of security courses. Semester projects were also judged and encouraged using a “venture capital” model, in which students had to pretend as if they were pitching their ideas for a start-up company to potential investors.

CNT 5410 Computer and Network Security: *Fall 2014-2022.* Totally rewrote the syllabus and slide material, giving the class its first major overhaul in a number of years. While many old themes remain, new lecture blocks including Web Security, Cellular Security and Social Engineering were developed from scratch. This new course material was made available to all other faculty members teaching this class, who have since used my slides and syllabus.

Georgia Tech In addition to the above courses, I also developed the following course while serving as a faculty member at Georgia Tech.

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

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

D. Individual Student Guidance

1. Research Scientists Supervised

None.

2. Ph.D. Students Graduated

Hadi Abdullah University of Florida

Fall 2016–Summer 2022

Evaluating the security of ML-driven voice interfaces. Now: Research Scientist at Visa Research

Chaitrali Amrutkar Georgia Institute of Technology

Fall 2009–Fall 2013

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

- Logan Blue** University of Florida
Fall 2016–Summer 2022
Investigated biological feature reconstruction from voice recordings. Now: Research Scientist at Harbor Labs
- Jasmine Bowers** University of Florida
Fall 2015–Summer 2020
Her research focuses on mobile applications, and the development of tools for building secure systems. Now: Research Scientist, MITRE
- Henry “Hank” Carter** Georgia Institute of Technology
Fall 2010–Spring 2016
Developing techniques for secure function evaluation for privacy-preserving applications on constrained mobile devices. Now: Assistant Professor, Villanova University
- Italo Dacosta** Georgia Institute of Technology
Fall 2008–Summer 2012
Co-advised with Mustaque Ahamad. Research on scaling performance of SIP network components. Graduated Summer 2012, currently research scientist at EPFL.
- David Dewey** Georgia Institute of Technology
Fall 2011–Summer 2015
Investigated compiler techniques to remove software vulnerabilities from code. Now CTO of MailChimp.
- Cassidy Gibson** University of Florida
Fall 2019–Spring 2025
Investigated the use of AI in generating non-consensual imagery. Now Research Scientist at ActiveFence.
- Seth Layton** University of Florida
Fall 2020–Summer 2025
Detecting deepfakes in audio samples. Now Research Scientist at PinDrop.
- Christian Peeters** University of Florida
Fall 2016–Summer 2022
Develop techniques to detect and defend against call and message interception attacks in cellular networks. Now: Research Scientist at Harbor Labs
- Brad Reaves** University of Florida
Fall 2014–Spring 2017
Develop strong authentication techniques for cellular networks. Now: Assistant Professor at North Carolina State University.
- Nolen Scaife** University of Florida
Fall 2014–Spring 2019
Developed techniques to detect credit card skimming. First: Assistant Professor at the University of Colorado Boulder. Now: Director, Global Cyber Intelligence at Walmart
- Imani Sherman** University of Florida
Fall 2018–Summer 2021
Developing usable interfaces against robocalls. Co-advised with Juan Gilbert. Now: Assistant Professor at the University of California, San Diego
- Tyler Tucker** University of Florida
Fall 2021–Summer 2025
Evaluating the security of Bluetooth/cellular radios. Now: Associate at the Analysis Group.

Luis Vargas University of Florida
Fall 2016–Summer 2021
Developing techniques for network-based detection and mitigation of malware in a healthcare environment. Now: Data Scientist at the Alethia Group

Kevin Warren University of Florida
Fall 2019–Summer 2025
Detecting deepfake audio through linguistic information. Now: Associate at the Analysis Group.

2. Ph.D. Students Supervised

Nathaniel Bennett University of Florida
Fall 2022–Present
Finding vulnerabilities in cellular core networks via fuzzing.

Jordan Greene University of Florida
Fall 2025–Present
Defending voice-driven systems and cellular network security.

Allison Lu University of Florida
Fall 2022–Present
Measuring repeatability in computer security.

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

3. Ph.D. Students - Other

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

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

Ryon Kennedy University of Florida
Fall 2020–Spring 2023
Finding vulnerabilities in cellular core networks via fuzzing. Left to join UFIT.

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

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

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

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

Aviva Smith University of Florida
Fall 2025–Spring 2026
Detecting attacks on voice-driven systems and deepfakes.

4. M.S. Students Supervised

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

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

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

Christopher Grayson Georgia Institute of Technology
Fall 2012–Fall 2013
Developed continuous authentication mechanisms using the multitude of sensors available on a mobile phone. Now at Bishop Fox Consulting (industry).

Young Seuk Kim Georgia Institute of Technology
Fall 2012–Fall 2013
Performed research that compared the security vulnerabilities found in the traditional and mobile web.

Daniel Komaromy Georgia Institute of Technology
Fall 2008–Summer 2009
Research on building a real-time streaming audio system using attribute-based crypto for broadcast encryption.

Nigel Lawrence Georgia Institute of Technology
Fall 2011–Spring 2012
Discovered hijacking attacks in SNMPv3, a widely used and thought to be secure network management protocol. Now at Solute (industry).

Philip Marquardt Georgia Institute of Technology
Fall 2009–Present
Research on developing an iPhone application to prevent individuals from being profiled by Shopper Loyalty Programs. First with MIT Lincoln Labs, now Raytheon

Rishikesh Naik Georgia Institute of Technology
Fall 2008–Spring 2010
Research on converting expensive cryptographic primitives (e.g., Secure Function Evaluation) into efficient applications for mobile phones. Now with Cisco Systems.

Ashish Nautiyal University of Florida

Fall 2015–Spring 2016

Research on connecting telephone calls to the larger authentication infrastructure.

Nilesh Nipane Georgia Institute of Technology

Fall 2008–Spring 2010

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

Walter “Nolen” Sciafe Georgia Institute of Technology

Spring 2012–Spring 2014

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

Tyler Tucker University of Florida

Fall 2018–Spring 2021

Evaluating the security of Bluetooth radios.

5. M.S. Special Problems Students

Siddhant Deshmukh University of Florida

Fall 2016–Present

Developed tools for analysis of mobile digital financial services.

Chinmay Gangakhedkar Georgia Institute of Technology

Spring 2009

Research on multi-factor authentication using mobile phones.

Christopher Grayson Georgia Institute of Technology

Spring 2013

Research on continuous authentication using mobile phones.

Aarushi Karnany University of Florida

Fall 2016–Present

Developed tools for analysis of mobile digital financial services.

Rohit Matthews Georgia Institute of Technology

Spring 2011

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

Ashwin Narasimhan Georgia Institute of Technology

Spring 2009

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

Aamir Poonawalla Georgia Institute of Technology

Spring 2010

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

Erin Reddick Georgia Institute of Technology

Fall 2008–Fall 2009

Research on IPTV security with GTRI.

Lalanthika Vasudevan Georgia Institute of Technology

Spring 2009

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

6. Undergraduate Special Problems Students

Ethan Shernan Georgia Institute of Technology

Spring 2014

Developed an infrastructure for detecting billing bypass fraud attacks.

Young Seuk Kim Georgia Institute of Technology

Fall 2011–Spring 2012

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

Dane Van Dyck Georgia Institute of Technology

Summer 2009

Research on virtualization support for mobile phones.

E. Teaching Honors and Awards

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

II. RESEARCH AND CREATIVE SCHOLARSHIP

A. Thesis

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

B. Published Journal Papers (Refereed)

1. Kevin Butler, Kurt Hugenberg, Eakta Jain, Apu Kapadia Tadayoshi Kohno, Elissa Redmiles, Franziska Roesner, Mattea Sim, Patrick Traynor, and Hanna Barakat. Extending the Heilmeier Catechism to Evaluate Security and Privacy Systems: Who is Left Out? *IEEE Security and Privacy*, May/June 2025.
2. Cassidy Gibson, Vanessa Frost, Katie Platt, Washington Garcia, Luis Vargas, Sara Rampazzi, Vincent Bindschaedler, Patrick Traynor, and Kevin Butler. Analyzing the Monetization Ecosystem of Stalkerware. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2022. (Acceptance rate: 24%).
3. Bradley Reaves, Luis Vargas, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Characterizing the Security of the SMS Ecosystem with Public Gateways. *ACM Transactions on Privacy and Security (TOPS)*, 22(1), 2018.
4. Patrick Traynor, Kevin Butler, Jasmine Bowers, and Bradley Reaves. FinTechSec: Addressing the Security Challenges of Digital Financial Services. *IEEE S&P Magazine*, 15(5):85–89, 2017.
5. Nolen Scaife, Henry Carter, Rachel Jones, Lyrissa Lidsky, and Patrick Traynor. OnionDNS: A Seizure-Resistant Top-level Domain. *International Journal of Information Security (IJIS)*, 2017.
6. Bradley Reaves, Jasmine Bowers, Nolen Scaife, Adam Bates, Arnav Bharatiya, Patrick Traynor, and Kevin Butler. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. *ACM Transactions on Privacy and Security (TOPS)*, 2017.
7. Henry Carter and Patrick Traynor. OPFE: Outsourcing Computation for Private Function Evaluation. *International Journal of Information and Computer Security (IJICS)*, 2017.
8. Stephan Heuser, Bradley Reaves, Praveen Kumar Pendyala, Henry Carter, Alexandra Dmitrienko, William Enck, Negar Kiyavash, Ahmad-Reza Sadeghi, and Patrick Traynor. Phonion: Practical Protection of Metadata in Telephony Networks. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2017.
9. Bradley Reaves, Jasmine Bowers, Sigmond A. Gorski III, Olabode Anise, Rahul Bobhate, Raymond Cho, Hiranava Das, Sharique Hussain, Hamza Karachiwala, Nolen Scaife, Byron Wright, Kevin Butler, William Enck, and Patrick Traynor. *droid: Assessment and Evaluation of Android Application Analysis Tools. *ACM Computing Surveys (CSUR)*, 2016.
10. Chaitrali Amrutkar, Young Seuk Kim, and Patrick Traynor. Detecting Mobile Malicious Webpages in Real Time. *IEEE Transactions on Mobile Computing (TMC)*, To Appear 2016.
11. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Outsourcing Secure Two-Party Computation as a Black Box. *Journal of Security and Communication Networks (SCN)*, To Appear 2016.
12. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Secure Outsourced Garbled Circuit Evaluation for Mobile Devices. *Journal of Computer Security (JCS)*, 24(2):137–180, 2016.
13. Adam Bates, Kevin Butler, Micah Sherr, Clay Shields, Patrick Traynor, and Dan Wallach. Accountable Wiretapping -or- I Know They Can Hear You Now. *Journal of Computer Security (JCS)*, 23(2):167–195, 2015.

14. Henry Carter, Chaitrali Amrutkar, Italo Dacosta, and Patrick Traynor. For Your Phone Only: Custom Protocols for Efficient Secure Function Evaluation on Mobile Devices. *Journal of Security and Communication Networks (SCN)*, 7(7):1165–1176, 2014.
15. Chaitrali Amrutkar, Patrick Traynor, and Paul van Oorschot. An Empirical Evaluation of Security Indicators in Mobile Web Browsers. *IEEE Transactions on Mobile Computing (TMC)*, 14(5), 2015.
16. Andrew Harris, Seymour Goodman, and Patrick Traynor. Privacy and Security Concerns Associated with Mobile Money Applications in Africa. *Washington Journal of Law, Technology & Arts*, 8(3), 2013.
17. Italo Dacosta, Saurabh Chakradeo, Mustaque Ahamad, and Patrick Traynor. One-Time Cookies: Preventing Session Hijacking Attacks with Stateless Authentication Tokens. *ACM Transactions on Internet Technology (TOIT)*, 12(1), 2012.
18. Cong Shi, Xiapu Luo, Patrick Traynor, Mostafa Ammar, and Ellen Zegura. ARDEN: Anonymous networkRking in Delay tolErant Networks. *Journal of Ad Hoc Networks*, 10(6):918–930, 2012.
19. Patrick Traynor. Characterizing the Security Implications of Third-Party EAS Over Cellular Text Messaging Services. *IEEE Transactions on Mobile Computing (TMC)*, 11(6):983–994, 2012.
20. Italo Dacosta, Vijay Balasubramaniyan, Mustaque Ahamad, and Patrick Traynor. Improving Authentication Performance of Distributed SIP Proxies. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 22(11):1804–1812, 2011.
21. Patrick Traynor, Chaitrali Amrutkar, Vikhyath Rao, Trent Jaeger, Patrick McDaniel, and Thomas La Porta. From Mobile Phones to Responsible Devices. *Journal of Security and Communication Networks (SCN)*, 4(6):719 – 726, 2011.
22. Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure Attribute-Based Systems. *Journal of Computer Security (JCS)*, 18(5):799–837, 2010.
23. Patrick Traynor, Kevin Butler, William Enck, Kevin Borders, and Patrick McDaniel. malnets: Large-Scale Malicious Networks via Compromised Wireless Access Points. *Journal of Security and Communication Networks (SCN)*, 2(3):102–113, 2010.
24. Patrick Traynor. Securing Cellular Infrastructure: Challenges and Opportunities. *IEEE Security & Privacy Magazine*, 7(4), 2009.
25. Kevin Butler, Sunam Ryu, Patrick Traynor, and Patrick McDaniel. Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, 20(12):1803–1815, 2009.
26. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Mitigating Attacks On Open Functionality in SMS-Capable Cellular Networks. *IEEE/ACM Transactions on Networking (TON)*, 17(1), 2009.
27. Patrick Traynor, Michael Chien, Scott Weaver, Boniface Hicks, and Patrick McDaniel. Non-Invasive Methods for Host Certification. *ACM Transactions on Information and System Security (TIS-SEC)*, 2008.
28. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. *Journal of Computer Security (JCS)*, 16(6):713–742, 2008.
29. Patrick Traynor, Raju Kumar, Heesook Choi, Sencun Zhu, Guohong Cao, and Thomas La Porta. Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. *IEEE Transactions on Mobile Computing (TMC)*, 6(6), 2007.

C. Published Books and Parts of Books

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

D. Edited Proceedings

None.

E. Conference Presentations

E.1. Conference Papers with Proceedings (Refereed)

1. Seaver Thorn, Nathaniel Bennett, Kevin Butler, Patrick Traynor, and William Enck. ASN1spect: Uncovering ASN.1 Compiler-Generated Vulnerabilities in Critical Infrastructure. In *ACM Secure Development (SecDev) Conference*, 2026.
2. Eric Pauley and Yohan Beugin and Paul Barford and Patrick Traynor and Patrick McDaniel. Honey Configs: Measuring the Disclosure and Exploitation of Cloud IP Address Reuse. In *Proceedings of the ACM Internet Measurement Conference (IMC)*, 2026.
3. Nathaniel Bennett, Tyler Tucker, Carson Stillman, William Enck, Patrick Traynor, and Kevin Butler. Fizzle: A Framework for Deterministic and Reproducible Network Fuzzing. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2026.
4. Seth Layton, Bernardo Medeiros, Kevin Butler, and Patrick Traynor. AI Wrote My Paper and All I Got Was This False Negative: Measuring the Efficacy of Commercial AI Text Detectors. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2026.
5. Carson Stillman, Jennifer Sheldon, Ian Garrett, Patrick Traynor, Ryan Gerdes, Sara Rampazzi, and Kevin Butler. EM-Flow: Advanced Electromagnetic Control Flow Verification for Embedded Systems. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2025.
6. Daniel Olszewski, Tyler Tucker, Kevin Butler, and Patrick Traynor. SoK: Towards a Unified Approach to Applied Replicability for Computer Security. In *Proceedings of the USENIX Security Symposium (Security)*, 2025.
7. Daniel Olszewski, Allison Lu, Anna Crowder, Nathaniel Bennett, Seth Layton, Sri Hrushikesh Varma Bhupathiraju, Tyler Tucker, Siddhant Kalgutkar, Hunter Ver Helst, Carson Stillman, Kevin Butler, Sara Rampazzi, and Patrick Traynor. Reproducibility in Applied Security Conferences: An 11-Year Review on Artifacts and Evaluation Committees. In *Proceedings of ACM Conference on Reproducibility and Replicability (REP)*, 2025.
8. Cassidy Gibson and Daniel Olszewski and Natalie Grace Brigham and Anna Crowder and Kevin R. B. Butler and Patrick Traynor and Elissa M. Redmiles and Tadayoshi Kohno. Analyzing the AI Nudification Application Ecosystem. In *Proceedings of the USENIX Security Symposium (Security)*, 2025.

9. Tyler Tucker, Nathaniel Bennett, Martin Kotuliak, Simon Erni, Srdjan Capkun, Kevin Butler, and Patrick Traynor. Detecting IMSI-Catchers by Characterizing Identity Exposing Messages in Cellular Traffic. In *Symposium on Network and Distributed System Security (NDSS)*, 2025. (Acceptance Rate 16.1%).
10. Anna Crowder, Allison Lu, Kevin Childs, Carson Stillman, Patrick Traynor, and Kevin Butler. Data to Infinity and Beyond: Examining Data Sharing and Reuse Practices in the Computer Security Community. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2025.
11. Magdalena Pasternak, Kevin Warren, Daniel Olszewski, Susan Nittroueri, Patrick Traynor, and Kevin Butler. Characterizing the Impact of Audio Deepfakes in the Presence of Cochlear Implant Simulated Audio. In *Symposium on Network and Distributed System Security (NDSS)*, 2025. (Acceptance Rate 16.1%).
12. Anna Crowder, Daniel Olszewski, Patrick Traynor, and Kevin R. B. Butler. I Can Show You the World (of Censorship): Extracting Insights from Censorship Measurement Data Using Statistical Techniques. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, December 2024. (Acceptance Rate 21.8%).
13. Kevin Childs, Cassidy Gibson, Anna Crowder, Kevin Warren, Carson Stillman, Elissa Redmiles, Eakta Jain, Patrick Traynor, and Kevin Butler. “I Had Sort of a Sense that I Was Always Being Watched... Since I Was”: Examining Interpersonal Discomfort From Continuous Location-Sharing Applications. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2024. (Acceptance Rate 16.9%).
14. Nathaniel Bennett, Weidong Zhu, Benjamin Simon, Ryon Kennedy, William Enck, Patrick Traynor, and Kevin Butler. RANsacked: A Domain-Informed Approach for Fuzzing LTE and 5G RAN-Core Interfaces. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2024. (Acceptance Rate 16.9%).
15. Kevin Warren, Tyler Tucker, Anna Crowder, Daniel Olszewski, Allison Lu, Caroline Fedele, Magdalena Pasternak, Seth Layton, Kevin Butler, Carrie Gates, and Patrick Traynor. Better Be Computer or I’m Dumb”: A Large-Scale Evaluation of Humans as Audio Deepfake Detectors. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2024. (Acceptance Rate 16.9%).
16. K. Virgil English, Nathaniel Bennett, Seaver Thorn, Kevin Butler, William Enck, and Patrick Traynor. Examining Cryptography and Randomness Failures in Open-Source Cellular Cores. In *Proceedings of the ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2024. (Acceptance rate: 21.3%)(Best Paper).
17. Seth Layton, Tyler Tucker, Daniel Olszewski, Kevin Warren, Carrie Gates, Kevin Butler, and Patrick Traynor. SoK: The Good, The Bad, and The Unbalanced: Measuring Structural Limitations of Current Deepfake Datasets. In *Proceedings of the USENIX Security Symposium (Security)*, 2024. (Acceptance Rate 18.3%).
18. Imani Munyaka, Daniel Delgado, Juan Gilbert, Jaime Ruiz, and Patrick Traynor. “I used to live in Florida”: Exploring the Impact of Spam Call Warning Accuracy on Callee Decision-Making. In *Symposium on Usable Security and Privacy (USEC)*, 2024.
19. Jianliang Wu, Patrick Traynor, Dongyan Xu, Dave (Jing) Tian, and Antonio Bianchi. Finding Traceability Attacks in the Bluetooth Low Energy Specification and Its Implementations. In *Proceedings of the USENIX Security Symposium (Security)*, 2024. (Acceptance Rate 18.3%).

20. Daniel Olszewski, Allison Lu, Carson Stillman, Kevin Warren, Cole Kitroser, Alejandro Pascual, Divyajyoti Ukirde, Kevin Butler, and Patrick Traynor. “Get in Researchers; We’re Measuring Reproducibility”: A Reproducibility Study of Machine Learning Papers in Tier 1 Security Conferences. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2023. (Acceptance rate: 19.8%).
21. Christian Peeters, Tyler Tucker, Anushri Jain, Kevin Butler, and Patrick Traynor. LeopardSeal: Detecting Call Interception via Audio Rogue Base Stations. In *Proceedings of the ACM International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2023. (Acceptance rate: 21%).
22. Tyler Tucker, Hunter Searle, Kevin Butler, and Patrick Traynor. Blue’s Clues: Practical Discovery of Non-Discoverable Bluetooth Devices. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2023. (Acceptance rate: 14%).
23. Hadi Abdullah, Aditya Karlekar, Saurabh Prasad, Muhammad Sajidur Rahman, Logan Blue, Luke Bauer, Vincent Bindschaedler, and Patrick Traynor. Attacks as Defenses: Designing Robust Audio CAPTCHAs Using Attacks on Automatic Speech Recognition Systems. In *Symposium on Network and Distributed System Security (NDSS)*, 2023. (Acceptance rate: 16%).
24. Daniel Olszewski, Sandeep Sathyanarayana, Weidong Zhu, Kevin Butler, and Patrick Traynor. HallMonitor: A Framework for Identifying Network Policy Violations in Software. In *IEEE Conference on Communications and Network Security (CNS)*, 2022.
25. Hadi Abdullah, Aditya Karlekar, Vincent Bindschaedler, and Patrick Traynor. Demystifying Limited Adversarial Transferability in Automatic Speech Recognition Systems. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. (Acceptance rate: 32%).
26. Logan Blue, Kevin Warren, Hadi Abdullah, Cassidy Gibson, Luis Vargas, Jessica O’Dell, Kevin Butler, and Patrick Traynor. Who Are You (I Really Wanna Know)? Detecting Audio DeepFakes Through Vocal Tract Reconstruction. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2022. (Acceptance rate: 17.2%).
27. Grant Hernandez, Marius Muench, Dominik Maier, Alyssa Milburn, Shinjo Park, Tobias Scharnowski, Tyler Tucker, Patrick Traynor, and Kevin R. B. Butler. FirmWire: Transparent Dynamic Analysis for Cellular Baseband Firmware. In *Symposium on Network and Distributed System Security (NDSS)*, 2022. (Acceptance rate: 16.2%).
28. Christian Peeters, Christopher Patton, Imani N. Sherman, Daniel Olszewski, Thomas Shrimpton, and Patrick Traynor. SMS OTP Security (SOS): Hardening SMS-Based Two Factor Authentication. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2022. (Acceptance rate: 18.2%).
29. Hadi Abdullah, Muhammad Sajidur Rahman, Christian Peeters, Cassidy Gibson, Washington Garcia, Vincent Bindschaedler, Thomas Shrimpton, and Patrick Traynor. Beyond L_p Clipping: Equalization based Psychoacoustic Attacks against ASRs. In *The Asian Conference on Machine Learning (ACML)*, 2021.
30. Imani Sherman and Daniel Delgado and Juan Gilbert and Jaime Ruiz and Patrick Traynor. Characterizing User Comprehension in the STIR/SHAKEN Anti-Robocall Standard. In *Proceedings of the Annual Research Conference on Communications Information and Internet Policy (TPRC 49)*, 2021.
31. Hadi Abdullah, Kevin Warren, Vincent Bindschaedler, Nicolas Papernot, and Patrick Traynor. The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2021. (Acceptance rate: 12.1%).

32. Hadi Abdullah, Muhammad Sajidur Rahman, Washington Garcia, Logan Blue, Kevin Warren, Anurag Swarnim Yadav, Tom Shrimpton, and Patrick Traynor. Hear “No Evil”, See “Kenansville”: Efficient and Transferable Black-Box Attacks on Speech Recognition and Voice Identification Systems. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2021. (Acceptance rate: 12.1%).
33. Imani Sherman, Jasmine Bowers, Liz-Laure Laborde, Juan E. Gilbert, Jaime Ruiz, and Patrick Traynor. Truly Visual Caller ID? An Analysis of Anti-Robocall Applications and their Accessibility to Visually Impaired Users. In *IEEE International Symposium on Technology and Society (IEEE ISTAS)*, 2020.
34. Imani Sherman, Jasmine Bowers, Keith McNamara, Juan Gilbert, Jaime Ruiz, and Patrick Traynor. Are You Going to Answer That? Measuring User Responses to Anti-Robocall Application Indicators. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2020. (Acceptance rate: 17.4%).
35. Joseph Choi, Dave Tian, Grant Hernandez, Christopher Patton, Benjamin Mood, Thomas Shrimpton, Patrick Traynor, and Kevin Butler. A Hybrid Approach to Secure Function Evaluation Using SGX. In *Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACCS’19)*, 2019. (Acceptance Rate: 17.0% for full papers).
36. Vanessa Frost, Dave Tian, Christie Ruales, Patrick Traynor, and Kevin Butler. Examining DES-based Cipher Suite Support within the TLS Ecosystem. In *Proceedings of the ACM ASIA Conference on Computer and Communications Security (ASIACCS’19)*, 2019. (Acceptance Rate: 22.0% for all papers).
37. Dave Tian, Joseph Choi, Grant Hernandez, Patrick Traynor, and Kevin Butler. A Practical Intel SGX Setting for Linux Containers in the Cloud. In *Proceedings of the ACM Conference on Data and Application Security and Privacy (CODASPY’19)*, 2019. (Acceptance rate: 23.5%).
38. Nolen Scaife, Jasmine Bowers, Christian Peeters, Grant Hernandez, Imani Sherman, Lisa Anthony, and Patrick Traynor. Kiss from a Rogue: Evaluating Detectability of Pay-at-the-Pump Card Skimmers. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2019. (Acceptance rate: 12.0%).
39. Jasmine Bowers, Imani Sherman, Kevin Butler, and Patrick Traynor. Characterizing Security and Privacy Practices in Emerging Digital Credit Applications. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019. (Acceptance rate: 25.6%).
40. Hadi Abdullah, Washington Garcia, Christian Peeters, P. Traynor, K. Butler, and J. Wilson. Practical Hidden Voice Attacks against Speech and Speaker Recognition Systems. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2019. (Acceptance Rate: 17.1%).
41. Lius Vargas, Logan Blue, Vanessa Frost, Christopher Patton, N. Scaife, K. Butler, and P. Traynor. Digital Healthcare-Associated Infection Analysis of a Major Multi-Campus Hospital System. In *Proceedings of the ISOC Network & Distributed Systems Security Symposium (NDSS)*, 2019. (Acceptance Rate: 17.1%).
42. Dominik Wermke, Nicolas Huaman, Yasemin Acar, Bradley Reaves, Patrick Traynor, and Sascha Fahl. A Large Scale Investigation of Obfuscation Use in Google Play. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2018. Acceptance Rate: 20.1%.
43. Nolen Scaife, Christian Peeters, and Patrick Traynor. Fear the Reaper: Characterization and Fast Detection of Card Skimmers. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2018. Acceptance Rate: 19.0%.

44. Dave (Jing) Tian, Grant Hernandez, Joseph Choi, Vanessa Frost, Christie Raules, Kevin Butler, Patrick Traynor, Hayawardh Vijayakumar, Lee Harrison, Amir Rahmati, and Mike Grace. AT-tention Spanned: Comprehensive Vulnerability Analysis of AT Commands Within the Android Ecosystem. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2018. Acceptance Rate: 19.0%.
45. Luis Vargas, Gyan Hazarika, Rachel Culpepper, Kevin Butler, Thomas Shrimpton, Doug Szajda, and Patrick Traynor. Mitigating Risk while Complying with Data Retention Laws. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2018.
46. Logan Blue, Luis Vargas, and Patrick Traynor. Hello, Is It Me You're Looking For? Differentiating Between Human and Electronic Speakers for Voice Interface Security. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2018.
47. Logan Blue, Hadi Abdullah, Luis Vargas, and Patrick Traynor. 2MA: Verifying Voice Commands via Two Microphone Authentication. In *ACM Asia Conference on Computer and Communications Security (ASIACCS)*, 2018. (Acceptance Rate: 20.0%).
48. Nolen Scaife, Christian Peeters, Camilo Velez, Hanqing Zhao, Patrick Traynor, and David Arnold. The Cards Aren't Alright: Detecting Counterfeit Gift Cards Using Encoding Jitter. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2018. (Acceptance Rate: 10.4%).
49. Christian Peeters, Hadi Abdullah, Nolen Scaife, Jasmine Bowers, Patrick Traynor, Bradley Reaves, and Kevin Butler. Sonar: Detecting SS7 Redirection Attacks Via Call Audio-Based Distance Bounding. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2018. (Acceptance Rate: 10.4%).
50. Tyler Ward, Joseph Choi, Kevin Butler, John M. Shea, Patrick Traynor, and Tan Wong. Privacy Preserving Localization Using a Distributed Particle Filtering Protocol. In *IEEE MILCOM*, 2017. (Acceptance Rate: 56%).
51. Bradley Reaves and Logan Blue and Hadi Abdullah and Luis Vargas and Patrick Traynor and Thomas Shrimpton. AuthentiCall: Efficient Identity and Content Authentication for Phone Calls. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2017. (Acceptance Rate: 16.3%).
52. Jasmine Bowers and Bradley Reaves and Imani N. Sherman and Patrick Traynor and Kevin Butler. Regulators, Mount Up! Analysis of Privacy Policies for Mobile Money Applications. In *Proceedings of the USENIX Symposium on Usable Privacy and Security (SOUPS)*, 2017. (Acceptance Rate: 26.5%).
53. Bradley Reaves, Logan Blue, and Patrick Traynor. AuthLoop: End-to-End Cryptographic Authentication for Telephony over Voice Channels. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2016. (Acceptance Rate: 15.5%).
54. Dave Tian, Nolen Scaife, Adam Bates, Kevin Butler, and Patrick Traynor. Making USB Great Again with USBFILTER. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2016. (Acceptance Rate: 15.5%).
55. Bradley Reaves, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Detecting SMS Spam in the Age of Legitimate Bulk Messaging. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2016. (Acceptance Rate: 35.0%).
56. Nolen Scaife, Henry Carter, Patrick Traynor, and Kevin Butler. CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data. In *IEEE International Conference on Distributed Computing Systems (ICDCS)*, 2016. (Acceptance Rate: 17.6%).
57. Bradley Reaves, Nolen Scaife, Dave Tian, Logan Blue, Patrick Traynor, and Kevin Butler. Sending out an SMS: Characterizing the Security of the SMS Ecosystem with Public Gateways. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 2016. (Acceptance Rate: 13.0%).

58. Benjamin Mood, Debayan Gupta, Henry Carter, Kevin Butler, and Patrick Traynor. Frigate: A Validated, Extensible, and Efficient Compiler and Interpreter for Secure Computation. In *Proceedings of the IEEE European Symposium on Security and Privacy*, 2016. (Acceptance Rate: 17.3%).
59. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Outsourcing Secure Two-Party Computation as a Black Box. In *Proceedings of the International Conference on Cryptology and Network Security*, 2015. (Acceptance Rate: 52.9%).
60. Nolen Scaife, Henry Carter, and Patrick Traynor. OnionDNS: A Seizure-Resistant Top-level Domain. In *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*, 2015. (Acceptance Rate: 28.1%).
61. Bradley Reaves, Nolen Scaife, Adam Bates, Patrick Traynor, and Kevin Butler. Mo(bile) Money, Mo(bile) Problems: Analysis of Branchless Banking Applications in the Developing World. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2015. (Acceptance Rate: 15.7%).
62. Bradley Reaves, Ethan Shernan, Adam Bates, Henry Carter, and Patrick Traynor. Boxed Out: Blocking Cellular Interconnect Bypass Fraud at the Network Edge. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2015. (Acceptance Rate: 15.7%).
63. David Dewey, Bradley Reaves, and Patrick Traynor. Uncovering Use-After-Free Conditions In Compiled Code. In *Proceedings of the International Conference on Availability, Reliability and Security (ARES)*, 2015. (Acceptance Rate: 22%).
64. Ethan Shernan, Henry Carter, Dave Tian, Patrick Traynor, and Kevin Butler. More Guidelines Than Rules: CSRF Vulnerabilities from Noncompliant OAuth 2.0 Implementations. In *Proceedings of the International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA)*, 2015. (Acceptance Rate: 22.7%).
65. Henry Carter, Charles Lever, and Patrick Traynor. Whitewash: Outsourcing Garbled Circuit Generation for Mobile Devices. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2014. (Acceptance Rate: 19.9%).
66. Henry Carter, Benjamin Mood, Patrick Traynor, and Kevin Butler. Secure Outsourced Garbled Circuit Evaluation for Mobile Devices. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2013. (Acceptance Rate: 16.2%).
67. Chaitrali Amrutkar, Matti Hiltunen, Shobha Venkataraman, Kaustubh Joshi, Patrick Traynor, Trevor Jim, and Oliver Spatscheck. Why is My Smartphone Slow? On The Fly Diagnosis of Poor Performance on the Mobile Internet. In *Proceedings of The 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2013. (Acceptance Rate: 19.6%).
68. Saurabh Chakradeo, Brad Reaves, Patrick Traynor, and William Enck. MAST: Triage for Market-scale Mobile Malware Analysis. In *Proceedings of the ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2013. (Acceptance Rate: 15.0%)(Best Paper).
69. Charles Lever, Manos Antonakakis, Brad Reaves, Patrick Traynor, and Wenke Lee. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2013. (Acceptance rate: 18.8%).
70. Chaitrali Amrutkar, Kapil Singh, Arunabh Verma, and Patrick Traynor. VulnerableMe: Measuring Systemic Weaknesses in Mobile Browser Security. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, 2012. (Acceptance rate: 25%) (Best Paper - SAIC Student Paper Competition (GT)) (Finalist - CSAW AT&T Applied Security Research Best Paper Competition 2012).

71. Chaitrali Amrutkar, Patrick Traynor, and Paul van Oorschot. A Measurement Study of SSL Indicators on Mobile Browsers: Extended Life, or End of the Road? In *Proceedings of the Information Security Conference (ISC)*, 2012. (Acceptance rate: 32%) (Best Student Paper).
72. Italo Dacosta, Mustaque Ahamad, and Patrick Traynor. Trust No One Else: Detecting MITM Attacks Against SSL/TLS Without Third-Parties. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*, 2012. (Acceptance Rate: 20.2%).
73. Adam Bates, Kevin Butler, Micah Sherr, Clay Shields, Patrick Traynor, and Dan Wallach. Accountable Wiretapping -or- I Know They Can Hear You Now. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2012. (Acceptance Rate: 17.8%).
74. Yacin Nadji, Jon Giffin, and Patrick Traynor. Automated Remote Repair for Mobile Malware. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2011. (Acceptance Rate: 18.5%).
75. Nilesh Nipane, Italo Dacosta, and Patrick Traynor. "Mix-In-Place" Anonymous Networking Using Secure Function Evaluation. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2011. (Acceptance Rate: 18.5%).
76. Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. (sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2011. (Acceptance Rate: 13.9%).
77. Philip Marquardt, David Dagon, and Patrick Traynor. Impeding Individual User Profiling in Shopper Loyalty Programs. In *Proceedings of the International Conference on Financial Cryptography and Data Security (FC)*, 2011. (Acceptance Rate: 35.1%).
78. David Dewey and Patrick Traynor. No Loitering: Exploiting Lingerin Vulnerabilities in Default COM Objects. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2011. (Acceptance Rate: 20.1%).
79. Vijay Balasubramaniyan, Aamir Poonawalla, Mustaque Ahamad, Michael Hunter, and Patrick Traynor. PinDr0p: Using Single-Ended Audio Features to Determine Call Provenance. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2010. (Acceptance Rate: 17.2%).
80. Patrick Traynor, Joshua Schiffman, Thomas La Porta, Patrick McDaniel, Abhrajit Ghosh, and Farooq Anjum. Constructing Secure Localization Systems with Adjustable Granularity. In *IEEE Global Communications Conference (GLOBECOM)*, 2010. (Acceptance Rate: 35.6%).
81. Patrick Traynor. Characterizing the Security Implications of Third-Party EAS Over Cellular Text Messaging Services. In *Proceedings of the Second IEEE International Conference on Security and Privacy in Communication Networks (SecureComm)*, 2010. (Acceptance Rate: 25.0%).
82. Kapil Singh, Samrit Sangal, Nehil Jain, Patrick Traynor, and Wenke Lee. Evaluating Bluetooth as a Medium for Botnet Command and Control. In *Proceedings of the International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA)*, 2010. (Acceptance Rate: 30.7%).
83. Italo Dacosta and Patrick Traynor. Proxychain: Developing a Robust and Efficient Authentication Infrastructure for Carrier-Scale VoIP Networks. In *Proceedings of the USENIX Annual Technical Conference (ATC)*, 2010. (Acceptance Rate: 17.0%).
84. Frank S. Park, Chinmay Gangakhedkar, and Patrick Traynor. Leveraging Cellular Infrastructure to Improve Fraud Prevention. In *Proceedings of the Annual Computer Security Applications Conference (ACSAC)*, 2009. (Acceptance Rate: 19.0%).

85. Patrick Traynor, Michael Lin, Machigar Ongtang, Vikyath Rao, Trent Jaeger, Thomas La Porta, and Patrick McDaniel. On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2009. (Acceptance Rate: 18.4%).
86. Brendan Dolan-Gavitt, Abhinav Srivastava, Patrick Traynor, and Jonathon Giffin. Robust Signatures for Kernel Data Structures. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2009. (Acceptance Rate: 18.4%).
87. Italo Dacosta, Vijay Balasubramaniyan, Mustaque Ahamad, and Patrick Traynor. Improving Authentication Performance of Distributed SIP Proxies. In *Proceedings of the Conference on Principles, Systems and Applications of IP Telecommunications (IPTComm)*, 2009. (Acceptance Rate: 43.3%).
88. Patrick Traynor, Kevin Butler, William Enck, and Patrick McDaniel. Realizing Massive-Scale Conditional Access Systems Through Attribute-Based Cryptosystems. In *Proceedings of the ISOC Network & Distributed System Security Symposium (NDSS)*, 2008. (Acceptance Rate: 17.7%).
89. Patrick Traynor, Patrick McDaniel, and Thomas La Porta. On Attack Causality in Internet-Connected Cellular Networks. In *Proceedings of the USENIX Security Symposium (SECURITY)*, 2007. (Acceptance Rate: 12.3%).
90. Sunam Ryu, Kevin Butler, Patrick Traynor, and Patrick McDaniel. Leveraging Identity-based Cryptography for Node ID Assignment in Structured P2P Systems. In *Proceedings of the IEEE International Symposium on Security in Networks and Distributed Systems (SSNDS)*, 2007. (Acceptance Rate: 40%).
91. Luke St. Clair, Lisa Johansen, William Enck, Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Trent Jaeger. Password Exhaustion: Predicting the End of Password Usefulness. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, 2006. (Invited Paper).
92. Kevin Butler, William Enck, Jennifer Plasterr, Patrick Traynor, and P. McDaniel. Privacy-Preserving Web-Based Email. In *Proceedings of the International Conference on Information Systems Security (ICISS)*, December 2006. (Acceptance Rate: 30.4%).
93. Matthew Pirretti, Patrick Traynor, Patrick McDaniel, and Brent Waters. Secure Attribute-Based Systems. In *Proceedings of the Thirteenth ACM Conference on Computer and Communications Security (CCS)*, November 2006. (Acceptance Rate: 14.8%).
94. Patrick Traynor, William Enck, Patrick McDaniel, and Thomas La Porta. Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networks. In *Proceedings of the Twelfth Annual ACM International Conference on Mobile Computing and Networking (MobiCom)*, September 2006. (Acceptance Rate: 11.7%).
95. Patrick Traynor, Michael Chien, Scott Weaver, Boniface Hicks, and Patrick McDaniel. Non-Invasive Methods for Host Certification. In *Proceedings of the Second IEEE International Conference on Security and Privacy in Communication Networks (SecureComm)*, August 2006. (Acceptance Rate: 25.4%).
96. Patrick Traynor, JaeShung Shin, Barat Madan, Shashi Phoha, and Thomas La Porta. Efficient Group Mobility for Heterogeneous Sensor Networks. In *Proceedings of the IEEE Vehicular Technology Conference (VTC Fall)*, September 2006. (Acceptance Rate: 58%).
97. Patrick Traynor, Raju Kumar, Hussain Bin Saad, Guohong Cao, and Thomas La Porta. LIGER: Implementing Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks. In *Proceedings of the 4th ACM International Conference on Mobile Systems, Applications and Services (MobiSys)*, June 2006. (Acceptance Rate: 15.4%).

98. Patrick Traynor, Guohong Cao, and Thomas La Porta. The Effects of Probabilistic Key Management on Secure Routing in Sensor Networks. In *Proceedings of the 2006 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2006. (Acceptance Rate: 38.8%).
99. Patrick Traynor, Heesook Choi, Guohong Cao, Sencun Zhu, and Thomas La Porta. Establishing Pair-Wise Keys In Heterogeneous Sensor Networks. In *Proceedings of the 25th Annual IEEE Conference on Computer Communications (INFOCOM)*, April 2006. (Acceptance Rate: 18%).
100. William Enck, Patrick Traynor, Patrick McDaniel, and Thomas La Porta. Exploiting Open Functionality in SMS-Capable Cellular Networks. In *Proceedings of the Twelfth ACM Conference on Computer and Communications Security (CCS)*, November 2005. (Acceptance Rate: 15%).

Removed for external version.

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

None.

E.3. Conference Presentations without Proceedings

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

F. Other

F.1. Submitted Journal Papers

None.

F.2. Refereed Research Reports

None.

F.3. Software

1. *GSM Air Interface Simulator*: Developed a full voice, data and SMS capable simulator for the wireless portion of a GSM network. Models communications down to the timeslot for highest possible accuracy. Used in the majority of our work on cellular security.
2. *Malicious Telephony Load Tester*: Built a system on top of the TM1 Telecom Database testing suite to allow for a comparison of malicious traffic of varying composition.

F.4. Published Papers (Non-Refereed)

1. Patrick Traynor. Characterizing the Limitations of Third-Party EAS Over Cellular Text Messaging Services. Technical report, 3G Americas Whitepaper, 2008.

2. Lisa Johansen, Kevin Butler, William Enck, Patrick Traynor, and Patrick McDaniel. Grains of SANs: Building Storage Area Networks from Memory Spots. Technical Report NAS-TR-0060-2006, Network and Security Research Center, Department of Computer Science and Engineering, Pennsylvania State University, University Park, PA, 2007.

F.5. Books in Preparation

None.

F.6. Workshops and External Courses

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

G. Research Proposals and Grants (Principal Investigator)

1. Approved and Funded

1. **Artus Protocol STTR Phase II - Extension**
Sponsor: Office of Naval Research
Investigator(s): Patrick Traynor (PI), Kevin Butler
Amount: \$300,000 over 2 years
Awarded: August 2023
2. **Testing Audio Deep Fake Detectors**
Sponsor: Bank of America
Investigator(s): Patrick Traynor (PI), Kevin Butler

Amount: \$150,000 over 1 year
Awarded: August 2023

3. **Reconstructing Human Physiological Features from Audio Samples**
Sponsor: Office of Naval Research
Investigator(s): Patrick Traynor (PI), Kevin Butler
Amount: \$700,000 over 3 years
Awarded: August 2021
4. **Testing Audio Deep Fake Detectors**
Sponsor: Bank of America
Investigator(s): Patrick Traynor (PI), Kevin Butler
Amount: \$274,000 over 2 years
August 2021
5. **Deploying Defenses for Cellular Networks Using the AWARE Testbed**
Sponsor: Department of Homeland Security: CISA:
Investigator(s): Patrick Traynor (PI), Kevin Butler, Guofei Gu, Radu Stoleru, Walter Magnussen, P. R. Kumar
Amount: \$3,100,000 over 4 years
Awarded: October 2019
6. **SaTC: CORE: Medium: Securing the Voice Processing Pipeline Against Adversarial Audio**
Sponsor: NSF Secure and Trustworthy Cyberspace
Investigator(s): Patrick Traynor (PI), Thomas Shrimpton, Vincent Bindschaedler
Amount: \$1,199,999 over 4 years
Awarded: October 2019
7. **Artus Protocol STTR Phase II**
Sponsor: Office of Naval Research
Investigator(s): Patrick Traynor (PI), Kevin Butler
Amount: \$800,000 over 4 years
Awarded: August 2019
8. **Evaluating the Security of QR Code-Based Payments**
Sponsor: Discover Financial
Investigator(s): Patrick Traynor (PI)
Amount: \$50,000 over 1 year
Awarded: September 2018
9. **Workshop: Addressing the Technical Security Challenges of Emerging Digital Financial Services**
Sponsor: NSF Secure and Trustworthy Cyberspace
Investigator(s): Patrick Traynor (PI), Kevin Butler
Amount: \$50,000 over 1 year
Awarded: September 2017
10. **Designing Strong End-to-End Authentication Mechanisms for Modern Telephony Systems**
Sponsor: NSF Secure and Trustworthy Cyberspace
Investigator(s): Patrick Traynor (PI)
Amount: \$500,000 over 3 years
Awarded: July 2016
11. **Digital Healthcare-Associated Infection: Measurement, Defense and Prevention in a Modern Digital Healthcare Ecosystem**

Sponsor: National Science Foundation
Investigator(s): Patrick Traynor (PI), Kevin Butler, Shigang Chen
Amount: \$1,200,000 over 4 years
Awarded: June 2016

12. Evaluating and Improving Security in Emerging Branchless Banking Systems

Sponsor: NSF Secure and Trustworthy Cyberspace
Investigator(s): Patrick Traynor (PI)
Amount: \$500,000 over 3 years
Awarded July 2015

13. Prevention and Detection of Disallowed Connections in Mobile and Pervasive Systems

Sponsor: CISE-ECE Harris Endowed Seed Fund Program
Investigator(s): Patrick Traynor (PI), Renato Figueiredo (PI)
Amount: \$40,000 over 1 year
Awarded December 2014

14. Mobile Excursion Study Support

Sponsor: Hanscom AFB Electronic Systems Command Development Planning Division (ESC/XR)
Investigator(s): Patrick Traynor (PI), Mustaque Ahamad, Jeff Evans, Chuck Bokath
Amount: \$280,000 over 3 months
Awarded July 2012

15. Characterizing the Security Limitations of Accessing the Mobile Web

Sponsor: NSF Secure and Trustworthy Cyberspace
Investigator(s): Patrick Traynor (PI) and William Enck (NC State)
Amount: \$334,000 over 3 years
Awarded July 2012

16. Mitigating Attacks on Mobile Devices and Critical Cellular Infrastructure

Sponsor: US Department of Defense - Defense University Research Instrumentation Program (DURIP)
Investigator(s): Patrick Traynor (PI), Jon Giffin, Mustaque Ahamad
Amount: \$210,081 over 1 year
Awarded June 2011

17. Characterizing and Implementing Efficient Primitives for Privacy-Preserving Computation

Sponsor: DARPA PROgramming Computation on EncryptEd Data (PROCEED) – Broad Agency Announcement
Investigator(s): Patrick Traynor (PI) and Kevin Butler (UOregon)
Amount: \$580,000 over 4 years
Awarded May 2011

18. Security for Converged IMS Networks

Sponsor: US Department of Defense
Investigator(s): Patrick Traynor (PI), Mustaque Ahamad and Russ Clark
Amount: \$242,401 over 1 year
Awarded August 2010

19. CAREER: Protecting User Data on Lost, Stolen and Damaged Mobile Phones

Sponsor: NSF Trustworthy Computing
Investigator(s): Patrick Traynor (PI)
Amount: \$400,000 over 5 years
Awarded: May 2010

20. **Provably Anonymous Networking Through Secure Function Evaluation**

Sponsor: NSF Trustworthy Computing

Investigator(s): Patrick Traynor (PI)

Amount: \$200,000 over 2 years

Awarded: July 2009

2. Pending

Removed for external version.

H. Research Proposals and Grants (Contributor)

1. Approved and Funded

1. **RADIOBEAR**

Sponsor: Nightwing Intelligence / US Air Force

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

Amount: \$249,327 for 1 year

Awarded September 2025

2. **Artus Protocol STTR Phase II - Extension**

Sponsor: Office of Naval Research

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

Amount: \$300,000 over 1 year

Awarded: August 2024

3. **SaTC: Frontier: Securing the Future of Computing for Marginalized and Vulnerable Populations**

Sponsor: NSF SaTC

Investigator(s): Kevin Butler (PI), Patrick Traynor, Tadayoshi Kohno, Franz Roesner, Apu Kapadia, Eakta Jain.

Amount: \$7,500,000 for 5 years

Awarded October 2022

4. **Mobile Interface Security Analysis**

Sponsor: Office of Naval Research

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

Amount: \$700,000 over 3 years

January 2020

5. **AFOSR University Center of Excellence in Assured Autonomy in Contested Environments (AACOE)**

Sponsor: Air Force Office of Scientific Research and Air Force Research Labs

Investigator(s): Warren Dixon (PI), Patrick Traynor

Amount: \$5,999,515 for 5 years

March 2019

6. **ROCKY: Reliable Obfuscated Communications Kit for everYone**

Sponsor: DARPA Resilient Anonymous Communication for Everyone (RACE) – Broad Agency Announcement

Investigator(s): Thomas Shrimpton (PI), Patrick Traynor, Kevin Butler, Vincent Bindschaedler, Nadia Heninger

Amount: \$1,600,000 over 4 years

Awarded May 2019

7. **Smartphones and Mobile Money: Analysis and Recommendations for Security**
 Sponsor: GSMA Mobile For Development Foundation
 Investigator(s): Kevin Butler (PI), Patrick Traynor
 Amount: \$28,500
 December 2017

8. **WiFiUS: Collaborative Research: SELIOT: Securing Lifecycle of Internet-of-Things**
 Sponsor: NSF CNS WiFiUS
 Investigator(s): Gene Tsudik (PI), Patrick Traynor
 Amount: \$300,000 for 2 years
 December 2016

9. **Cloud-based Oblivious Spectrum Mapping and Allocation**
 Sponsor: NSF CNS EARS
 Investigator(s): John Shea (PI), Tan Wong, Patrick Traynor
 Amount: \$532,952 for 2 years
 October 2016

10. **DURIP: Developing Research Capability in Cyber-Physical Systems at the University of Florida**
 Sponsor: Small
 Investigator(s): Kevin Butler (PI), Patrick Traynor, My Thai
 Amount: \$200,000 for 2 years
 September 2015

11. **Securing the New Converged Telephony Landscape**
 Sponsor: NSF TWC: Small
 Investigator(s): Mustaque Ahamad (PI) and Patrick Traynor
 Amount: \$500,000 for 3 years
 Submitted: December 2012

12. **Facilitating Free and Open Access to Information on the Internet**
 Sponsor: NSF Trustworthy Computing
 Investigator(s): Nick Feamster (PI), Wenke Lee, Patrick Traynor, Hans Klein, Roger Dingledine, Michael Freedman and Edward W. Felten
 Amount: \$1,500,000 for 4 years
 Awarded: June 2011

13. **Monitoring Free and Open Access to Information on the Internet**
 Sponsor: Google Focus Program
 Investigator(s): Nick Feamster (PI), Wenke Lee, Mustaque Ahamad, Patrick Traynor, Henry Owen, Ellen Zegura, Zvi Galil
 Amount: \$1,000,000 for 2 years
 Awarded: November 2011

14. **Dynamic-attribute-based Disclosure of Health Information in Emergency Care Scenarios**
 Sponsor: Health Systems Institute (HSI) Seed Grant Program
 Investigator(s): Doug Blough (PI), Mustaque Ahamad, Patrick Traynor and Jim Jose
 Amount: \$50,000 over 1 year
 Awarded: August 2009

15. **Federal Cyber Service Scholarships at Georgia Tech**
 Sponsor: NSF SFS Scholarships
 Investigator(s): Seymour Goodman (PI), Patrick Traynor
 Amount: \$1,250,682 over 5 years
 Awarded: June 2009

16. **Security for IMS-Enabled Converged Applications**

Sponsor: US Department of Defense

Investigator(s): Mustaque Ahamad (PI), Patrick Traynor (PI), Michael Hunter, Russ Clark

Amount: \$146,121 for 1 year

Awarded: August 2008

2. Pending

Removed for external version.

I. Research Honors and Awards

1. Social Impact Award, New York University CSAW Applied Research Competition, 2025.
2. Distinguished Paper (Runner Up), USENIX Security Symposium, 2025.
3. Best Paper Award, ACM Conference on Reproducibility and Replicability (REP), 2025.
4. Distinguished Paper, ACM Conference on Computer and Communications Security (CCS), 2024.
5. Best Paper Award, ACM Conference on Data and Application security and Privacy (CODASPY), 2024.
6. Distinguished Paper, USENIX Security Symposium, 2018.
7. Fellow, Center for Financial Inclusion at Accion, 2017.
8. Sloan Research Fellow, Alfred P. Sloan Foundation, 2014.
9. Best Paper, The ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec); Budapest, Hungary, 2013.
10. Best Student Paper, The Information Security Conference (ISC); Passau, Germany, 2012
11. Lockheed Inspirational Young Faculty Award, 2012
12. Best Demo, "Is Browsing the Internet on Your Mobile Phone Secure?" Chaitrali Amrutkar (Ph.D Advisee), CoC Research Day, 2011
13. Best Poster, "(sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers" Arunabh Verma, Henry Carter (MS, Ph.D Advisees), CoC Research Day, 2011
14. National Science Foundation CAREER Award, 2010
15. Pennsylvania State University Alumni Association Dissertation Award, 2007
16. Pennsylvania State University CSE Graduate Research Assistant Award, 2007
17. AT&T Wireless Fellowship, 2005

III. SERVICE

A. Professional Activities

A.1. Memberships and Activities in Professional Societies

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

A.2. Conference Committee Activities

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

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

B. On-Campus Committees

B.1. University of Florida

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

B.2. Georgia Tech

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

C. Special Assignments

None.

D. Ph.D. Examining Committees

Ph.D. Examining Committees

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

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

External Member of Ph.D. Research Committee

None.

External Member of Ph.D. Examining Committee

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

Ph.D. Thesis Proposal Committees

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

Ph.D. Qualifying Exam Committees—Georgia Tech

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

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

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

E. External Member of M.S. Examining Committee

M.S. Thesis Defense Committees None.

F. Consulting and Advisory Appointments

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

G. Civic Activities

None.

IV. NATIONAL AND INTERNATIONAL PROFESSIONAL RECOGNITION

A. Honors and Awards

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

B. Invited Conference Session Chairmanships

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

C. Professional Registration

None.

D. Patents

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

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

E. Editorial and Reviewer Work for Technical Journals and Publishers

Associate Editor:

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

Guest Editor:

Journals

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

Reviewer for:

Journals

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

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

Conferences and Workshops

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

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

F. Expert Witness Services

1. *Natalie Delgado, et al. v. Meta Platforms Inc., Case No.: 23-cv-04181 (N.D. Cal.):* Expert witness for the Defense (via Gibson, Dunn & Crutcher, LLP). *October 2025 - Present.*
2. *Google LLC v Headwater Research LLC:* Expert witness for the Plaintiff for Inter Partes Review (via Wolf, Greenfield & Sacks, P.C.). *August 2025 - Present.*
3. *Amazon v Headwater Research LLC:* Expert witness for the Plaintiff for Inter Partes Review (via Perkins Coie, LLP). *August 2025 - Present.*
4. *HBCU Messaging US LP v. Apple, Inc. et al., Civil No. 1:24-cv-1199,:* Expert witness for the Defense (via Fish & Richardson, LLP). *June 2025 - Present.*
5. *PACid v. Citibank, N.A., 1:24-cv-00272-DAE (W.D. Tex.):* Expert witness for the Defense (via Troutman Pepper Locke LLP). *April 2025 - November 2025.*
6. *Facebook Inc. Derivative Litigation., Consolidated C.A. No. 2018-0307-JTL (Del. Ch.):* Expert witness for the Defense (via Wachtell, Lipton, Rosen & Katz, LLP) *January 2025 - July 2025.*
7. *Averon US, Inc. vs AT&T Inc., Case No. 1:22-cv-01341-TMH:* Expert witness for the Defense (via O'Melveny & Myers, LLP). *November 2024 - September 2025.*
8. *RightQuestion, LLC v. AT&T Inc. et al.,, Case No. 2-24-cv-00094 -JRG:* Expert witness for the Defense (via Duane Morris, LLP). *September 2024 - November 2025.*
9. *ByteDance Ltd. vs CellSpin Soft, Inc.:* Expert witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *February 2024 - October 2024.*
10. *Bank of America, N.A., vs PACid:* Expert witness for the Plaintiff for Inter Partes Review (via McNish PLLC). *December 2023 - March 2024.*
11. *Microsoft vs Proxense, LLC:* Expert witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *November 2023 - July 2025.*
12. *Samsung vs Headwater Research LLC:* Expert witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *August 2023 - October 2025.*

13. *Advanced Coding Technologies LLC v. ByteDance PTE. Ltd., and TikTok PTE. Ltd.*: Expert witness for the Defense for Non-Infringement (via Fish & Richardson, LLP). *July 2023 - September 2023.*
14. *Epic Games, Inc. & Anor v Google LLC & Ors - Federal Court of Australia Proceeding NSD 190 of 2021*: Expert witness for the Defendant (via Corrs Chambers Westgarth). *January 2023 - June 2024.*
15. *Rubin vs KAHOOT! ASA and KAHOOT! EDU*: Expert witness for the Defendant for Inter Partes Review (via Vasquez Benisek & Lindgren, LLP). *December 2022 - June 2024.*
16. *Telefonaktiebolaget LM Ericsson vs Apple, Inc.*: Expert witness for the Defendant, Non-Infringement and Invalidity (via WilmerHale LLP). *February 2022 - December 2022.*
17. *Wepay Global Payments, LLC v. Bank of America N. A.*: Expert Witness for the Defendant (via WilmerHale LLP) *September 2022 - November 2022.*
18. *Apple vs. R.N Nehushtan Trust Ltd.*: Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *August 2022 - June 2023.*
19. *Apple/Microsoft vs. Zipit Wireless*: Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *May 2021 - November 2022.*
20. *Blackberry Inc v MobileIron, Inc.*: Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *January 2021 - March 2021.*
21. *Apple Inc v Seven Networks, LLC*: Expert Witness for the Plaintiff for Inter Partes Review (via Fish & Richardson, LLP). *August 2019 - November 2020.*
22. *mSIGNIA, Inc. v. InAuth, Inc.*: Expert Witness for the Defendant for Inter Partes Review, Non-Infringement and Invalidity, Trade Secrets (via Quinn Emanuel Urquhart and Sullivan, LLP). *October 2017 - December 2018.*
23. *Huawei v. T-Mobile*: Expert Witness for the Defendant for Non-Infringement (via WilmerHale LLP, Alston & Bird LLP) *June 2016 - December 2017.*
24. *Telefonaktiebolaget LM Ericsson v Apple*: Expert Witness for the Defendant for Non-Infringement, Invalidity (via WilmerHale LLP). *June 2015 - December 2015.*
25. *Mayfonk v Nike*: Expert Witness for the Plaintiff for Infringement/Trade Secrets (via Paul Hastings). *June 2015 - November 2015.*
26. *Maxim Integrated Products v Bank of the West*: Expert Witness for the Defendant for Non-Infringement (via Paul Hastings LLP). *January 2014 - August 2014.*
27. *Maxim Integrated Products v Comerica Inc, et al.*: Expert Witness for the Defendant for Non-Infringement (via McKenna, Long & Aldridge LLP). *June 2014 - August 2014.*
28. *William Grecia v. Apple Inc. et al.*: Expert Consultant for the Defendant for Invalidity (via Kirkland & Ellis LLP). *July 2014 - August 2014.*
29. *Intertrust Technologies Corp. v. Apple Inc.*: Expert Consultant for Defendant for Invalidity and Non-Infringement (via Kirkland & Ellis LLP). *October 2013 - February 2014.*
30. *Maxim Integrated Products v KeyCorp Bank*: Expert Witness for the Defendant for Non-Infringement (via Calfee, Halter & Griswold LLP) *April 2013 - June 2013.*
31. *Intellectual Ventures LLC vs. Check Point; et al.*: Expert Consultant for the Plaintiff for Infringement (via Susman Godfrey LLP), *October 2012 - February 2015.*

V. OTHER CONTRIBUTIONS

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

1. Keynote: Well, It Worked on My Computer: Reproducibility in Computer Security Research. Learning from Authoritative Security Experiment Results (LASER) Workshop, December 2024. École Polytechnique Fédérale de Lausanne (EPFL, Switzerland).
2. Social Engineering and Two-Factor Authentication. Cyber Security Training Eastern Indonesia, August 2024. Financial Innovation Lab (EIFIL) (Denpasar, Bali, Indonesia).
3. DFS Security and Mobile Money Analysis. Cyber Security Training Eastern Indonesia, August 2024. Financial Innovation Lab (EIFIL) (Denpasar, Bali, Indonesia).
4. Keynote: Well, It Worked on My Computer: Reproducibility in Computer Security Research. EPFL Summer Research Institute (SURI), July 2024. École Polytechnique Fédérale de Lausanne (EPFL, Switzerland).
5. Humans vs The Computer Interfaces: The Challenge of Separating Deepfakes/Bots from People. North Central Florida Institute of Internal Auditors (IIA), May 2024.
6. Humans vs The Computer Interfaces: The Challenge of Separating Deepfakes/Bots from People. UF Quest 2: Siri is my Superpower: Communicating with AI, March 2024. University of Florida.
7. Humans vs The Computer Interfaces: The Challenge of Separating Deepfakes/Bots from People. Federal Information Integrity Research and Development (FIIRD) Interworking Group (IWG), March 2024. via NITRD, OSTP.
8. AI driven voice cloning scams. Discussion at the White House with Anne Neuberger (Deputy National Security Advisor for Cyber and Emerging Technologies), Jessica Rosenworcel (Chair of the Federal Communications Commission) and Lina Khan (Chair of the Federal Trade Commission), January 2024. Lead Academic facilitator.
9. Keynote: Well, It Worked on My Computer: Reproducibility, Tech Transfer, and Computer Security Research. National Science Foundation Secure and Trustworthy Cyberspace (SaTC) Vision 2.0 Workshop, March 2023. University of Texas at Dallas.
10. Humans vs The Computer Interfaces: Separating Deepfakes/Bots from People Using Psychoacoustics. UCLA Electrical and Computer Engineering Distinguished Seminar, February 2023. University of California, Los Angeles.
11. Keynote: Exploiting the Gaps Between Human and Machine Understanding of Audio: Frameworks, Attacks, and Defenses. ISCA Symposium on Security and Privacy in Speech Communication (SPSC), November 2021. Virtual.
12. The State of Voice Cloning Technology. Federal Trade Commission (FTC) Workshop on Voice Cloning Technologies, January 2020. Washington, DC.
13. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. North Carolina State University Department of Computer Science Colloquium, January 2020. Raleigh, NC.
14. Moving from research to practice: How to maximize the impact of SaTC projects. National Science Foundation Secure and Trustworthy Cyberspace (SaTC) PI Meeting, October 2019. Alexandria, VA.
15. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. Purdue University Computer Science Excellence Lecture Series, October 2019. West Lafayette, IN.

16. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. Bank of America - Colloquium Series, March 2019. Charlotte, NC.
17. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. CISP – Helmholtz Center for Information Security, Saarland University, February 2019. Saarbrücken, Germany.
18. Because That's Where the Money Is: Designing Defenses for Entrenched Legacy Payment Systems. University of Maryland - Distinguished Colloquium, February 2019. College Park, MD.
19. Responsible Finance for the Digital Client. Foromic Conference, October 2018. Barranquilla, Colombia.
20. Panel: Authentication Challenges for New Interfaces, Devices, and Wireless Networks. ACM Conference on Security and Privacy in Wireless and Mobile Networks, June 2018. Stockholm, Sweden.
21. Sonar: Detecting SS7 Redirection Attacks Via Call Audio-Based Distance Bounding. CyberSecurity@KAIST Workshop - KAIST, June 2018. Daejeon, South Korea.
22. Why Caller-ID Spoofing Is So Easy (and Why End-To-End Solutions Are the Way Forward). IEEE Workshop on Technology and Consumer Protection (ConPro'18), May 2018. San Francisco, CA.
23. Panel: The Future of Cybersecurity. SEC Academic Conference - Auburn University, May 2018. Auburn, AL.
24. Sound Principles: Verifying Voice Commands in an IoT World. IoT Security Workshop - Aalto University, September 2017. Helsinki, Finland.
25. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Eurecom Institute, September 2017. Sophia Antipolis, France.
26. Panel: Infrastructure Stability. ITU-T Focus Group Digital Financial Services, December 2016. Geneva, Switzerland.
27. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. ETH Zurich, December 2016. Zurich, Switzerland.
28. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. University of Richmond, October 2016. Richmond, Virginia.
29. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Indiana University, September 2016. Bloomington, Indiana.
30. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Aalto University Computer Science Department Forum, August 2016. Helsinki, Finland.
31. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. KAIST Information Security Seminar - Korean Advanced Institute of Science and Technology, June 2016. Daejeon, South Korea.
32. Updated Mobile Money Vulnerability Report. International Telecommunications Union Digital Financial Services Working Group Workshop, May 2016. Washington, DC.
33. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. UF Eye Opener Discovery Breakfast - University of Florida, May 2016. Gainesville, FL.

34. Who Do I Think You Are? Challenges and Opportunities in Telephony Authentication. Illinois Science of Security (SoS) Lablet Speaker Series - University of Illinois, Urbana-Champaign, April 2016. Urbana-Champaign, Illinois.
35. New Trends in Cybersecurity: Vulnerabilities in Branchless Banking Systems. United States Departments of State and Justice - Cybersecurity and Cybercrime Workshop for Lusophone Africa, September 2015. Maputo, Mozambique.
36. New Trends in Cybersecurity: Vulnerabilities in Branchless Banking Systems. United States Departments of State and Justice - ECCAS Cybersecurity and Cybercrime Workshop, August 2015. Kinshasa, Democratic Republic of Congo.
37. Chasing Telephony Security: Where the Wild Things... Are? University of Florida - Department Colloquium, January 2014. Gainesville, FL.
38. Chasing Telephony Security: Where the Wild Things... Are? Verizon Wireless RNC/Data Center, October 2013. Alpharetta, GA.
39. Chasing Telephony Security: Where the Wild Things... Are? University of Waterloo - CrySP Speaker Series on Privacy, October 2013. Waterloo, ON, Canada.
40. Analyzing Malicious Traffic in Cellular Networks. GSM Association's (GSMA) Mobile Malware Community Workshop, July 2013. Mountain View, CA.
41. Threats to Mobile Devices. US Federal Trade Commission (FTC) Public Forum - Invited Speaker, June 2013. Washington, D.C.
42. Chasing Telephony Security: Where the Wild Things... Are? University of Wisconsin - Madison, Security Seminar, March 2013. Madison, WI.
43. The Core of the Matter: Analyzing Malicious Traffic in Cellular Carriers. Invited Talk: Centre for Secure Information Technologies (CSIT) Queen's University, March 2013. Belfast, Northern Ireland.
44. Chasing Telephony Security: Where the Wild Things... Are? Stanford Security Seminar, March 2013. Stanford, CA.
45. Chasing Telephony Security: Where the Wild Things... Are? University of California, Berkeley, Security Group, March 2013. Berkeley, CA.
46. Chasing Telephony Security: Where the Wild Things... Are? Carnegie Mellon University CyLab Seminar, February 2013. Pittsburgh, PA.
47. Chasing Telephony Security: Where the Wild Things... Are? University of Oregon Department of Computer Science Colloquium, November 2012. Eugene, OR.
48. Chasing Telephony Security: Where the Wild Things... Are? University of Washington Department of Electrical Engineering, Network Security Lab (NSL): Invited Talk, November 2012. Seattle, WA.
49. Needles and Haystacks: Digging for Ground Truth on Mobile Malware. ZISC Workshop on Secure Mobile and Cloud Computing, ETH Zurich, June 2012. Zurich, Switzerland.
50. Panel: Advice for Early Career Faculty. CRA Career Mentoring Workshop, February 2012. Washington, D.C.
51. Research Challenges in Cellular and Mobile Network Security. US-China Software Workshop (Co-Sponsored by NSF and NSFC), September 2011. Beijing, China.

52. Mobile Security: Understanding Risks to Critical Infrastructure. Invited Talk: US Department of State East African Workshop on Cyberspace Security, July 2011. Nairobi, Kenya.
53. Tomorrow's Issues: Solving the Mobile Security Threat. Invited Talk: Centre for Secure Information Technologies (CSIT) Queen's University, March 2011. Belfast, Northern Ireland.
54. PinDr0p: Using Single-Ended Audio Features to Determine Call Provenance. Invited Talk: MITRE Corporation, March 2011. Burlington, MA.
55. Defeating Session Hijacking Attacks with Disposable Web Credentials. Invited Talk: Facebook, February 2011. Palo Alto, CA.
56. Understanding the Disruptive Potential of Malware in Cellular Networks. Invited Talk: RSA Conference, February 2011. San Francisco, CA.
57. Panel: Voice Security – Now Just a False Sense of Security and Privacy. Invited Panelist: Mobile Security Symposium, February 2011. San Francisco, CA.
58. Understanding the Disruptive Potential of Malware in Cellular Networks. Invited Talk: Concordia University, May 2010. Montreal, QC, Canada.
59. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Qualcomm Research, March 2010. San Diego, CA.
60. Privacy and Security Concerns for Personal and Mobile Health Devices. Invited Talk: Workshop to Set A Research Agenda for Privacy and Security of Healthcare Technologies, October 2009. Indianapolis, IN.
61. Considerations for EAS Over Cellular Text Messaging Services. 3G Americas Webinar, July 2009.
62. University Telephony Research Panel. Conference on Principles, Systems and Applications of IP Telecommunications (IPTCOMM), July 2009.
63. The Evolving Mobile Landscape: Emerging Security Threats. Mobile Security eConference, SC Magazine, June 2008.
64. Characterizing the Impact of Rigidity on the Security of Cellular Networks. University of Washington, February 2009. Seattle, WA.
65. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Microsoft Research, February 2009. Redmond, WA.
66. Next Year's Problems. Secure Computing (SC) Magazine Webinar, November 2008.
67. Panel: Embedded Systems and their Increasing Impact on Infrastructure Security. Workshop on Embedded Systems Security (WESS), October 2008.
68. Can you DoS me now? Security Issues in Cellular Networks. Georgia Institute of Technology, September 2008. Atlanta, GA.
69. Characterizing the Impact of Rigidity on the Security of Cellular Networks. Georgia Institute of Technology, April 2008. Atlanta, GA.
70. Characterizing the Impact of Rigidity on the Security of Cellular Networks. AT&T Research Labs, April 2008. Florham Park, NJ.
71. Characterizing the Impact of Rigidity on the Security of Cellular Networks. University of Arizona, March 2008. Tucson, AZ.

72. Cellular Networks Security Panel. USENIX Security Symposium, August 2007. Boston, MA.
73. malnets:Large-Scale Malicious Networks via Compromised Access Points. The Pennsylvania State University - ACM Club Invited Speaker, October 2006. State College, PA.
74. malnets:Large-Scale Malicious Networks via Compromised Access Points. The University of Michigan, October 2006. Ann Arbor, MI.
75. Exploiting Open Functionality in SMS-Capable Cellular Networks. The University of Michigan, October 2006. Ann Arbor, MI.
76. Exploiting Open Functionality in SMS-Capable Cellular Networks. High Technology Crime Investigation Association (HTCIA), September 2006. Pittsburgh, PA.
77. Trends in Security: Critical Engineering in the Large. Schlumberger Innovate IT! Workshop, May 2006. Cambridge, MA.
78. Exploiting Open Functionality in SMS-Capable Cellular Networks. InfraGard Pittsburgh Chapter General Meeting, September 2006. Pittsburgh, PA.
79. Exploiting Open Functionality in SMS-Capable Cellular Networks. InfraGard Pittsburgh Chapter General Meeting, September 2006. Pittsburgh, PA.

B. Special Activities

Presentations to Lay Media

1. How technology can fight digital fakery. The Babbage Podcast/The Economist <https://shows.acast.com/theeconomistbabbage/episodes/babbage-how-to-detect-a-deepfake>, January 2023.
2. Deepfake audio has a tell and researchers can spot it. Ars Technica <https://arstechnica.com/information-technology/2022/09/researchers-use-fluid-dynamics-to-spot-deepfake-voices/>, September 2022.
3. This security tool could help stop the problem of ransomware in its tracks. TheJournal.ie <https://www.thejournal.ie/ransomware-researchers-stop-2875032-Jul2016/>, July 2016.
4. Researchers Unleash Ransomware Annihilation. BankInfoSecurity - <http://www.bankinfosecurity.com/researchers-unleash-ransomware-annihilation-a-9255>, July 2016.
5. CryptoDrop Stops Ransomware by Stopping its Encryption. Security Intelligence - https://securityintelligence.com/news/cryptodrop-stops-ransomware-by-stopping-its-encryption/?utm_source=tfeed&utm_medium=twitter, July 2016.
6. Ransomware 'stopped' by new software. BBC - <http://www.bbc.com/news/technology-36772461>, July 2016.
7. Researchers create effective anti-ransomware solution. Help Net Security - <https://www.helpnetsecurity.com/2016/07/12/anti-ransomware-solution/>, July 2016.
8. Florida U boffins think they've defeated all ransomware. http://www.theregister.co.uk/2016/07/12/ransomware_defeated/, July 2016.

9. This Anti-Ransomware Tool Could Save You Hundreds of Pounds. Huffington Post - http://www.huffingtonpost.co.uk/entry/anti-ransomware-tool-save-hundreds-pounds_uk_57838beee4b0935d4b4b30ba, July 2016.
10. Researchers develop method to stop 100% of ransomware before it encrypts all files. Myce - <http://www.myce.com/news/researchers-develop-method-stop-100-ransomware-encrypts-files-79873/>, July 2016.
11. Desarrollan una solución para detener el ransomware. ComputerHoy - <http://computerhoy.com/noticias/software/desarrollan-solucion-detener-ransomware-47972>, July 2016.
12. Why your antivirus software can't stop ransomware. Futurity - <http://www.futurity.org/ransomware-computer-files-1198242-2/>, July 2016.
13. CryptoDrop Gives Users Hope to Prevent Ransomware Infections in the Future. Softpedia - <http://news.softpedia.com/news/cryptodrop-gives-users-hope-to-prevent-ransomware-infections-in-the-future-506187.shtml>, July 2016.
14. Could this be the answer to the ransomware threat?, Consumer Affairs. Consumer Affairs - <https://www.consumeraffairs.com/news/could-this-be-the-answer-to-the-ransomware-threat-071116.html>, July 2016.
15. Extortion extinction: Researchers develop a way to stop ransomware. Phys.org - <http://phys.org/news/2016-07-extortion-extinction-ransomware.html>, July 2016.
16. Researchers Develop A Way To Stop Ransomware By Watching The Filesystem. Slashdot - <https://yro.slashdot.org/story/16/07/08/2242244/researchers-develop-a-way-to-stop-ransomware-by-watching-the-filesystem>, July 2016.
17. Mohul Ghosh. Trak.in - Digital Money Apps In India Are Unsafe and Unsecured - Researchers. <http://trak.in/tags/business/2015/08/17/digital-money-apps-india-unsafe-unsecured/>, August 2015.
18. Richard Handford. Mobile World Live - Survey finds security holes in mobile money apps. <http://www.mobileworldlive.com/money/news-money/survey-finds-security-holes-in-mobile-money-apps/#.Vc27Y-QTmsQ.twitter>, August 2015.
19. JENNIFER VALENTINO-DEVRIES. Wall Street Journal - Researchers Find Security Flaws in Developing-World Money Apps. <http://blogs.wsj.com/digits/2015/08/11/researchers-find-security-flaws-in-developing-world-money-apps/>, August 2015.
20. Jonathon Cheng. Wall Street Journal - Samsung Phone Studied for Possible Security Gap. <http://online.wsj.com/news/articles/SB10001424052702304244904579276191788427198>, December 2013.
21. N. V. The Economist - The Threat in the Pocket. <http://www.economist.com/blogs/babbage/2013/10/difference-engine-0?fsrc=scn/fb/wl/bl/thethreatinthepocket>, October 2013.

22. Antone Gonsalves. ComputerWorld - Let's Dump Anti-Virus and Move On:. <http://blogs.computerworld.com/mobile-security/22969/lets-dump-av-and-move>, October 2013.
23. Mathew J. Schwartz. InformationWeek - Google: Don't Fear Android Malware. <http://www.informationweek.com/security/mobile/google-dont-fear-android-malware/240162399>, October 2013.
24. Kirsten Doyle. ITWeb - Android Threat Exaggerated, or is it? http://www.itweb.co.za/index.php?option=com_content&view=article&id=68055, October 2013.
25. Danielle Walker. SC Magazine - Mobile malware prevalence expands, but privacy-abusing apps should be top of mind. <http://www.scmagazine.com/mobile-malware-prevalence-expands-but-privacy-abusing-apps-should-be-top-of-mind/article/300597/>, June 2013.
26. Jim Burress. WABE NPR - Mobile Web Browsers Full of Security Risks, Tech Professor Finds. <http://wabe.org/post/mobile-web-browsers-full-security-risks-tech-professor-finds>, December 2012.
27. Mark Huffman. Consumer Affairs - Georgia Tech: mobile browsers fail safety test. <http://www.consumeraffairs.com/news/georgia-tech-mobile-browsers-fail-safety-test-120612.html>, December 2012.
28. Matthew J. Schwartz. Information Week - Blame Screen Size: Mobile Browsers Flunk Security Tests. <http://www.informationweek.com/security/mobile/blame-screen-size-mobile-browsers-flunk/240143999>, December 2012.
29. Jon Gold. Network World - Ga. Tech researchers: Mobile Browsers need better HTTPS indicators. <http://www.networkworld.com/news/2012/120512-mobile-browsers-264846.html>, December 2012.
30. United Press International. Study: Most mobile Web browsers unsafe. http://www.upi.com/Science_News/Technology/2012/12/05/Study-Most-mobile-Web-browsers-unsafe/UPI-73431354743353/#ixzz2EGtQsuId, December 2012.
31. Suzanne Choney. Mobile browser woes can fool even experts: report. <http://www.nbcnews.com/technology/mobile-browser-woes-can-fool-even-experts-report-1C7451203>, December 2012.
32. Meghan Kelly. VentureBeat - 3 hot security startups to watch. <http://venturebeat.com/2012/02/27/3-security-startups-to-watch-at-the-2012-rsa-conference/>, February 2012.
33. Jacob Goodwin. Government Security News - RSA 2012 – Pindrop Security can distinguish a fraudulent phone call from a real one. <http://www.gsnmagazine.com/node/25721?c=communications>, February 2012.
34. Matt Liebowitz. Phone hack logs keystrokes from nearby computers. MSNBC.com - http://www.msnbc.msn.com/id/44993238/ns/technology_and_science-security/#.TqU5MNSjPh4, October 2011.
35. Jacob Aron. iPhone keylogger can snoop on desktop typing. New Scientist - <http://www.newscientist.com/article/dn21059-iphone-keylogger-can-snoop-on-desktop-typing.html>, October 2011.

36. iPhone Keylogger Can Snoop on Desktop Typing. Slashdot - <http://mobile.slashdot.org/story/11/10/18/2346222/iphone-keylogger-can-snoop-on-desktop-typing>, October 2011.
37. Robert Lemos. Smart Phones Could Hear Your Password. Technology Review - <http://www.technologyreview.com/computing/38913/?p1=A2>, October 2011.
38. Kevin McCaney. Bad vibrations: How smart phones could steal PC passwords. Government Computer News - <http://gcn.com/articles/2011/10/18/smart-phone-sensors-steal-keystrokes.aspx>, October 2011.
39. PhysOrg. Turning iPhone into spiPhone: Smartphones' accelerometer can track strokes on nearby keyboards. PhysOrg.com - <http://www.physorg.com/news/2011-10-iphone-siphone-smartphones-accelerometer-track.html>, October 2011.
40. Brid-Aine Parnell. Securo-boffins call for 'self-aware' defensive technologies. The Register - http://www.theregister.co.uk/2011/09/14/self_aware_cyber_security_technologies_should_be_a_top_priority/, September 2011.
41. Clay Dillow. 'PinDr0p' Tech Uses Unique Noise Fingerprints to Trace Calls. Popular Science - <http://www.popsci.com/technology/article/2010-10/pindr0p-tech-tags-phone-calls-unique-fingerprints-trace-call-paths-across-networks>, October 2010.
42. Lewis Page. Voice-routing call fingerprint system fights vishing. The Register - http://www.theregister.co.uk/2010/10/06/voice_fingerprints, October 2010.
43. Science Daily. Voice Phishing: System to Trace Telephone Call Paths Across Multiple Networks Developed. <http://www.sciencedaily.com/releases/2010/10/101005121820.htm>, October 2010.
44. Brian Kalish. To Text or Not to Text During Emergencies. NextGov.com - http://www.nextgov.com/nextgov/ng_20100914_5986.php?oref=topnews, September 2010.
45. Ki Mae Heussner. 'Operation Chokehold': Fake Steve Jobs Rallies iPhone Users to Cripple AT&T Network. ABC News - <http://abcnews.go.com/Technology/GadgetGuide/fake-steve-jobs-rallies-iphone-users-cripple-att/story?id=9355447>, December 2009.
46. Bob Brown. Researchers Set Their Sights on iPhones, Mobile Malware. PC World Magazine - http://www.pcworld.com/article/182005/iphone_worms_mobile_malware.html?tk=rss, November 2009.
47. MacGregor Campbell. Botnets show their disruptive potential. New Scientist Magazine - <http://www.newscientist.com/article/mg20427347.000-mobile-botnets-show-their-disruptive-potential.html>, November 2009.
48. Angela Moscaritolo. Remote repair for infected phones in development. SC Magazine - <http://www.scmagazineus.com/remote-repair-for-infected-phones-in-development/article/157504/>, November 2009.
49. Bob Brown. iPhone worms, other smartphone malware in researchers' sights. Network World - <http://www.networkworld.com/news/2009/111109-smartphone-security-georgia-tech.html?hpg1=bn>, November 2009.

50. Urvaksh Karkaria. GT researchers work to secure cellphones. Atlanta Business Chronicle - <http://atlanta.bizjournals.com/atlanta/blog/atlantech/2009/11/cellphone.html>, November 2009.
51. Making Carriers Shoulder Smartphone Security. http://mobile.slashdot.org/story/09/11/11_2318247/Making-Carriers-Shoulder-Smartphone-Security?art_pos=31, November 2009.
52. Ben Meyer. Georgia Tech to Lead Fight Against Cell Phone Hackers. NBC 11 Atlanta - <http://www.11alive.com/news/local/story.aspx?storyid=132505&catid=3>, July 2009.
53. Illena Armstrong. Safeguarding your mobile networks. SC Magazine - <http://www.scmagazineus.com/Safeguarding-your-mobile-networks/article/138289/>, June 2009.
54. Kelli B. Grant. Four Free Cellphone Apps to Help Manage Your Money. SmartMoney Magazine - <http://www.smartmoney.com/Spending/Deals/4-Great-Free-Finance-Apps-for-Cellphones/>, June 2009.
55. Amanda Hoffstrom. Technology's limitations in alerting campus danger. UWire Magazine - <http://www.uwire.com/Article.aspx?id=3738798>, February 2009.
56. Laura Sydell. Compromise Allows Obama To Keep BlackBerry. National Public Radio - <http://www.npr.org/templates/story/story.php?storyId=99790788>, January 2009.
57. Dennis Carter. Questions abound as emergency alert flops Virginia Tech's text-message alert system failed when the sound of gunfire was heard on campus; officials scramble to understand why. eSchool News - http://www.eschoolnews.com/iphone/top-story/index.cfm?i=56122#_56122, November 2008.
58. Jessica Bauer. Study: Text alerts may fail in real emergency. Diamondback Online - <http://media.www.diamondbackonline.com/media/storage/paper873/news/2008/10/14/News/Study.Text.Alerts.May.Fail.In.Real.Emergency-3485509.shtml>, October 2008.
59. Associated Press. Hackers Expected To Start Targeting Cell Phones. <http://cbs5.com/watercooler/Cell.Phones.Hackers.2.840909.html>, 2008.
60. Associated Press. College alert systems unreliable, study says. http://www.ajc.com/search/content/metro/stories/2008/09/25/college_campus_alerts.html, 2008.
61. Lee Shearer. Study: Campus alerts unreliable. Athens Banner Herald http://www.onlineathens.com/stories/092508/uga_336494829.shtml, 2008.
62. Bill Ray. 3G Americas warns against text warning systems. The Register - http://www.theregister.co.uk/2008/09/18/emergency_text/, 2008.
63. 3G Americas. 3G Americas Highlights New Research Report on Use of Cellular Text Messaging for Emergency Alert Services. 3G Americas http://www.3gamericas.org/English/news_room/DisplayPressRelease.cfm?id=3400&s=ENG, 2008.
64. Evan Koblentz. Web Exclusive: From Messaging to Management Duty. Wireless Week - <http://www.wirelessweek.com/Messaging-to-Management-Duty.aspx>, 2008.
65. Christopher Beam. How Do You Intercept a Text Message? Turn your cell phone into a spy gadget. Slate Magazine <http://www.slate.com/id/2161402/>, 2007.

66. Jamming Cellphones with Text Messages. Slashdot <http://it.slashdot.org/it/05/10/05/1839217.shtml?tid=215&tid=172>, 2005.
67. Cell phone networks at risk? CNN http://money.cnn.com/2005/10/05/technology/hacker_cellphones/, 2005.
68. John Schwartz. Text Hackers Could Jam Cellphones, a Paper Says. The New York Times <http://www.nytimes.com/2005/10/05/technology/05phone.html?ex=1286164800&en=d917b9cd43dfaa31&ei=5090&partner=rssuserland&emc=rss>, 2005.