

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

FACEBOOK, INC., INSTAGRAM, LLC, and WHATSAPP INC.,
Petitioners

v.

BLACKBERRY LIMITED,
Patent Owner

Case No. IPR2019-00923
Patent No. 7,372,961

DECLARATION OF MARKUS JAKOBSSON, PH.D.

TABLE OF CONTENTS

I.	INTRODUCTION AND SCOPE OF WORK	4
II.	QUALIFICATIONS	5
III.	MATERIALS CONSIDERED	9
IV.	PERSON OF ORDINARY SKILL IN THE ART	10
V.	OVERVIEW OF THE '961 PATENT	11
VI.	INTERPRETATION OF THE '961 PATENT CLAIMS AT ISSUE.....	14
VII.	OVERVIEW OF THE CITED REFERENCES	16
A.	DSS.....	16
B.	Rose	19
C.	Menezes.....	21
D.	Schneier	23
VIII.	THE PETITION FAILS TO ESTABLISH THAT THE CITED REFERENCES WOULD HAVE PROVIDED FOR REPETITION OF THE REQUISITE OPERATIONS RECITED IN INDEPENDENT CLAIMS 1, 15, AND 23.....	23
A.	The '961 Claims Require Repetition Of Operations For Both Random Seed Value Generation And Hashing Of The Seed Value If A Preceding Hashed Value Is Not Less Than Said Order Q	23
B.	The References Cited In Grounds 1 And 3 Do Not Disclose Repetition Of The Required Operations	26
C.	Grounds 2 And 4 Do Not Disclose Repetition Of The Required Operations	38
IX.	SUBSEQUENT DSS RELEASES DEMONSTRATE THAT THE PETITIONER'S THEORY WAS ROOTED IN HINDSIGHT, NOT THE REALITY AT THE TIME	48

A.	FIPS 186-2 (October 2001).....	49
B.	FIPS 186-3 (June 2009).....	51
X.	LEGAL STANDARDS	55
A.	Obviousness.....	55
XII.	ADDITIONAL REMARKS	60

I, Markus Jakobsson, Ph.D., of Portoloa Valley, California, declare that:

I. INTRODUCTION AND SCOPE OF WORK

1. I have been retained by Fish & Richardson P.C. as an expert witness on behalf of BlackBerry Limited (“Blackberry” or “Patent Owner”). I understand that Facebook, Inc., Instagram, LLC, and WhatsApp Inc. (“Petitioners”) filed a petition for *inter partes* review (“IPR”) of claims 1, 2, 5, 15, 16, 19, 23, 24, and 27 of U.S. Patent No. 7,372,961 (“the ’961 patent”), and the case was assigned case no. IPR2019-00923.

2. I have been asked to provide my independent analysis of the ’961 patent in light of the materials cited below and my knowledge and experience in this field during the relevant period. I have been asked to consider what a person of ordinary skill in the art at the time of the invention of the ’961 patent (a “POSITA”; refer to ¶¶15-16) would have understood from the teachings of the ’961 patent, including scientific and technical knowledge related to the ’961 patent. I have also been asked to consider whether the references cited in the petition anticipate or render obvious the inventions described by independent claims 1, 15, and 23 of the ’961 patent. I have been told that this is only a preliminary stage of this proceeding, and accordingly, I address at this stage only certain aspects of the petition and only some of my analysis of the cited grounds. I reserve the

opportunity to address other issues and provide further analysis at a later date should it become necessary.

3. I am being compensated according to my normal hourly rate for my time providing my independent analysis in this aforementioned IPR proceeding, but my compensation is not contingent in any way on the content of my analysis or the outcome of this proceeding. I am not, and never was, an employee or agent of BlackBerry Limited, the owner of the '961 patent.

II. QUALIFICATIONS

4. My findings, as explained below, are informed by my studies, experiences, and background as described below and set forth in greater detail in my Curriculum Vitae (attached as Appendix A). Based on my experience, I understand and know of the capabilities of persons of ordinary skill in the field of cryptography during the timeframe leading up to the earliest priority date of the '961 patent (December 27, 2000), and indeed, I have personal knowledge and experience in working directly with many such persons in the field during that timeframe. I have also relied on my review and analysis of the prior art cited in the petition, information provided to me in connection with this case, and information I have independently reviewed. *See* Section III.

5. I received a Ph.D. in Computer Science/Cryptography in 1997 from the University of California San Diego and a M.Sc. in Computer Engineering in

1994 from the Lund Institute of Technology, Sweden. The title of my doctoral thesis was “Privacy vs. Authenticity,” and the focus was on secure digital commerce.

6. In my professional capacity, I have worked extensively with technologies related to computer security, mobile security, malware detection, quantitative and qualitative fraud analysis, and disruptive security. I have studied and published on many topics including randomness, digital signatures, and security analysis.

7. I am currently the Chief of Security and Data Analytics at Amber Solutions, Inc. My work there is to develop and evaluate algorithms to process sensor outputs from distributed and mobile sensor networks, which generate predicates related to the underlying actions of users in the proximity of the sensor networks; to develop and evaluate algorithms to protect the privacy of such users; to develop and evaluate algorithms to protect the security of such users; and to develop and evaluate algorithms that assess the risk of attacks against the system. I also work as an independent security researcher and consultant in the fields of computer/mobile security, fraud detection/prevention, digital rights management, malware, phishing, crimeware, mobile malware, and cryptographic protocols. In addition, I am a visiting research fellow of the Anti-Phishing Working Group, an

international consortium focused on unifying the global response to cybercrime in the public and private sectors.

8. I have been materially involved in designing, developing, testing, and assessing technologies for computer and network security, digital rights management, trust establishment, randomness, cryptography, socio-technical fraud, malware detection, detection of malicious emails, user interfaces, authentication, and fraud detection for over twenty years. Between 2013 and 2015, I served as a Senior Director at Qualcomm where my work focused on mobile security systems and mobile malware detection.

9. I have authored or co-authored more than 100 publications including technical papers, textbooks, textbook chapters, and articles, and have delivered numerous presentations including keynotes at several international conferences and workshops, as well as the US Patent and Trademark Office. These publications of topics including, but not limited to: digital signatures, randomness, security analysis, crimeware, phishing, electronic identity theft, email security, endpoint security, network security, digital fraud, software based attestation, remote transaction security and malware detection. My peer reviewed publications have been cited at least 14,875 times, according to Google Scholar. I hold more than 100 patents.

10. Examples of textbooks I have authored or co-authored on the topic of Internet security include:

- “Crimeware: Understanding New Attacks and Defenses” (Symantec Press, 2008)
- “Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft” (Wiley, 2006)
- “The Death of the Internet” (Wiley, 2012)
- “Mobile Authentication: Problems and Solutions” (Springer, 2012)
- “Understanding Social Engineering Based Scams” (Springer, 2016)
- “Towards Trustworthy Elections: New Directions in Electronic Voting” (Springer, 2010); and
- “Mobile Authentication: Problems and Solutions” (Springer, 2012)

11. I have authored or provided inventive contributions to numerous publications regarding randomness and/or digital signatures. Relevant examples of such publications include:

- “A practical secure physical random bit generator”, CCS ’98 Proceedings of the 5th ACM conference on Computer and communications security (pp. 103-111)
- “How to turn loaded dice into fair coins”, IEEE Transactions on Information Theory (Vol. 46, Issue: 3, May 2000)
- U.S. Patent No. 6,317,499

12. In view of the foregoing, I believe I possess the expertise to testify from the perspective of a POSITA with respect to the technology at issue in this case.

III. MATERIALS CONSIDERED

13. In preparing this declaration, I have considered the claims, specification, and prosecution history of the '961 patent. I have also read and considered the petition for *inter partes* review in Case No. IPR2019-00923. As part of my analysis for this declaration, I have considered my own knowledge and experience, including my work and experience in the field of cryptography, and my experience in working with others in these fields. Some additional materials that I have reviewed in preparing this declaration include the following documents:

- **Ex. 1001:** U.S. Patent No. 7,372,961 (“the '961 Patent”)
- **Ex. 1002:** Declaration of Jonathan Katz, Ph.D. (“Katz Declaration”)
- **Ex. 1003:** Prosecution History for U.S. Patent Application No. 10/025,924 (“’961 file history”)
- **Ex. 1004:** Federal Information Processing (FIPS) Publication 196, *Digital Signature Standard* (“DSS”)
- **Ex. 1005:** Excerpts from Alfred J. Menezes et al., *Handbook of Applied Cryptography* (“Menezes”)
- **Ex. 1006:** USENET article by Greg Rose (“Rose”)
- **Ex. 1008:** Excerpts from Bruce Schneier, *Applied Cryptography* (2d ed. 1996) (“Schneier”)
- **Ex. 1018:** Excerpts from Thammavarapu R. N. Rao, *Error Coding for Arithmetic Processors* (1974) (“Rao”)
- **EX. 1019:** Excerpts from Nancy A. Floyd, *Essentials of Data Processing* (1987) (“Floyd”)

- **Ex. 2002:** FIPS PUB 186-2, Digital Signature Standard (DSS) Change Notice (October 5, 2001)
- **Ex. 2003:** FIPS PUB 186-3, Digital Signature Standard (DSS) (June 2009)
- **Ex. 2004:** Corrected Final Ruling on Claim Construction/*Markman* Hearing, *Blackberry Limited v. Snap Inc.*, Case Nos. CV 18-1844-GW & 18-2693-GW (C.D. Cal. April 5, 2019) (“*Markman* Order”)
- **Ex. 2005:** Serge Vaudenay, *Evaluation Report on DSA* (2003) (“Vaudenay”)

14. Further, in addition to the exhibits listed above, I have reviewed the remaining exhibits submitted with the petition in this proceeding—including Ex. 1009 through Ex. 1017 and Ex. 1020 through Ex. 1035, which were listed in the petition (at petition p. i-iv) but not expressly set forth in the listing of Grounds 1-4 (at petition p. 4).

IV. PERSON OF ORDINARY SKILL IN THE ART

15. I understand that the teaching of the prior art is viewed through the eyes of a POSITA. My analysis is thus based on the perspective of a POSITA having this level of knowledge and skill at the relevant time of the invention. For purposes of my analysis, I have been informed that the priority date of the ’961 patent is no later than the December 27, 2000 timeframe, and I have applied this timeframe as being the relevant time for the perspective of a POSITA. For purposes of assessing the level of ordinary skill in the art, I have considered the

types of problems encountered in the art, the prior solutions to those problems found in prior art references, the speed with which innovations were made at that time, the sophistication of the technology, and the level of education of active workers in the field. As previously described, I have reviewed and understand the '961 patent. Based on my above-described experience, I am familiar with and know of the capabilities of a POSITA in this field during the late 1990s and early 2000s.

16. Based upon my knowledge and experience in this area, I believe a POSITA at the time of the invention would have had a bachelor of science degree in Computer Science or the equivalent, and approximately three years of work or research experience in the field of cryptography or an equivalent subject matter; or an advanced degree (masters or doctorate) in Computer Science or the equivalent, and less work or research experience (dependent in part on the level of degrees achieved) in the field of cryptography or an equivalent subject matter. My analysis as to the level of ordinary skill in the art would remain the same regardless of whether the time of the invention is found to be in December 2000, or some later time up until and including the December 26, 2001 filing date of the '961 patent.

V. OVERVIEW OF THE '961 PATENT

17. U.S. Patent 7,372,961 ("the '961 patent") is titled "Method of Public Key Generation," and its Abstract provides that "[a] potential bias in the generation

[of] a private key is avoided by selecting the key and comparing it against the system parameters[,]” and “[i]f a predetermined condition is attained it is accepted” or “[i]f not it is rejected a new a new key is generated.” EX1001, Abstract; *see also id.*, 2:65-3:4, 3:64-4:17, FIG. 2.

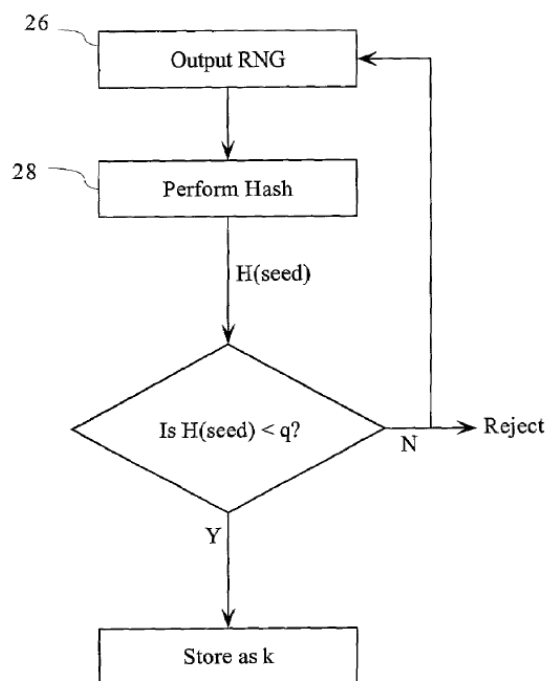
18. The background section of the '961 patent discloses that certain cryptographic functions such as the Digital Signature Algorithm (DSA) “utilize both long term and ephemeral keys to generate a signature of [a] message.” *Id.*, 1:54-56. A party that digitally signs a message typically employs a new ephemeral key k on a per-message or per-session basis in order to obscure the party’s permanent or long-term private key. *Id.*, 1:33-51.

19. According to the '961 patent, the selection of a key k should be randomized within a range of values. However, certain implementations of the DSA, including the Digital Signature Standard (DSS) that was promulgated by the National Institute of Standards and Technology (NIST) in the FIPS 186-2 standard, “inadvertently introduce a bias in to the selection of k .” *Id.*, 2:3-34. This bias made the selection of some values for the key k within the acceptable range more likely than others. *Id.* The '961 patent explained that the bias could be “exploited to extract a value of the [user’s long-term] private key d and thereafter render the security of the system vulnerable.” *Id.*, 2:33-36. For example, a security researcher, Dr. Daniel Bleichenbacher, determined that the method specified in

DSS for generating the ephemeral key k introduced sufficient bias such that an examination of 2^{22} signatures could allow a hacker to derive a user's long-term private key and effectively be able to pose as the user. *Id.*, 2:56-61. But, despite Dr. Bleichenbacher's recognition of a security vulnerability stemming from DSS's biased selection of ephemeral key k , a workable solution to the problem had not yet been resolved before the effective filing date of the '961 patent (December 27, 2000).

20. The solution proposed by the inventors of the '961 patent involved determining a key k by first generating a seed value from a random number generator, hashing the seed value using a suitable hash function (e.g., SHA), and comparing the output of the hash function to a parameter related to how the key k was to be employed (e.g., q in DSS). *Id.*, 3:64-4:17, FIG. 2. Because the length L of the output of the hash function in terms was at least as long as the length of the parameter q against which the hash output is compared, the output of the hash function would sometimes exceed the value of q and fall outside the acceptable range of values for the key k (e.g., since k must be less than q in DSS). The inventors proposed to ensure the selection of a compliant key k without introducing bias by either accepting the hashed seed value as the key k if the hashed value were less than the compared parameter (e.g., q), and rejecting the hashed value as the key k if the hashed value were greater than or equal to that parameter. EX1001,

3:64-4:17, FIG. 2 (reproduced below). If a hashed value was rejected for being too large, the method would repeat until an acceptable hash value was found that could be used as the key k . *Id.* This approach stands in contrast, for example, to DSS's method of generating key k where a modulo reduction was performed on the output of a hash function—thereby introducing the bias that Dr. Bleichenbacher was able to leverage to compromise the long-term private key of a user. *See* EX1004, p. 18.



EX1001, FIG. 2.

VI. INTERPRETATION OF THE '961 PATENT CLAIMS AT ISSUE

21. I understand that, for purposes of my analysis in this IPR proceeding, the terms appearing in the patent claims should be interpreted according to their plain and ordinary meaning under the *Phillips* standard. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2005) (*en banc*). According to *Phillips*, the

structure of the claims, the specification, and the patent prosecution history are used to construe a claim, and the “ordinary meaning” of a claim term is its meaning that would have been recognized by a POSITA after reading the entire patent. Moreover, *Phillips* provides that even treatises and dictionaries may be used, albeit under limited circumstances, to determine the meaning attributed by a POSITA to a claim term at the time of filing. For example, *Phillips* cautions against heavy reliance on the dictionary divorced from the intrinsic record of the patent, such as the patent’s specification. I followed this approach in my analysis.

22. I also understand that the words of the claims should be interpreted as they would have been interpreted by a POSITA at the time the invention was made (not today). For purposes of my analysis here, I have used the December 27, 2000 priority date of the ’961 patent as the date of invention. Without exception, however, my analysis of the proper meaning of the recited claim elements (under the *Phillips* standard) in this declaration would be correct if the date of invention was anywhere in the late 1990s or early 2000s.

23. I understand that the district court in a related proceeding involving the ’961 patent issued a Corrected Final Ruling on Claim Construction (“*Markman* Order”) on April 5, 2019. EX2004. I have reviewed the sections of the *Markman* Order that pertain to the ’961 patent. I understand that the claim construction standard under *Phillips* that applies in this *inter partes* review is the same standard

that the court applied in its Markman Order. For purposes of my analysis of the challenged claims in this IPR proceeding, I have employed the same constructions that were adopted by the court in the *Markman* Order. For example, I understand that the court construed the term “reducing mod q” in the *Markman* Order to mean “computing the remainder of dividing a value by q.” EX2004, 36. I have adopted the construction of this term for purposes of my analysis in this declaration. I note, however, that all of the shortcomings of Grounds 1-4 (as detailed below) would remain true even if the alternative interpretation of this claim phrase (as explained at p. 34-35 of the *Markman* Order) was implemented.

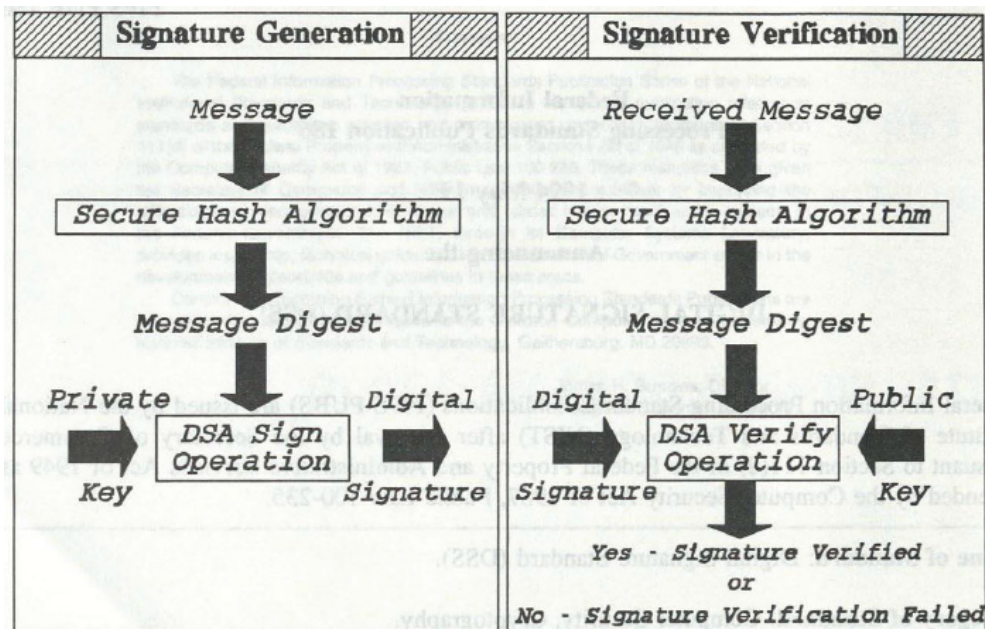
VII. OVERVIEW OF THE CITED REFERENCES¹

A. DSS

24. I understand that the primary reference cited in each ground of the petition is the Federal Information Processing Standards Publication (FIPS) 186,

¹ I understand the petition cites Rao (EX1018) and Floyd (EX1019) in combination with DSS, Rose, and Schneier in Ground 3, and in combination with DSS, Menezes, and Schneier in Ground 4. Pet., 57-63. The petition’s stated reasons for reliance on Rao and Floyd are not directly implicated by the particular claim element addressed in this declaration. Therefore, I did not include specific summaries of Rao and Floyd in this declaration.

entitled Digital Signature Standard (“DSS”). *See* EX1004. DSS describes a standardized version of the Digital Signature Algorithm. *Id.*, 5. This algorithm, which is also referenced in the background section of the ’961 patent (specifically, the ’961 patent refers to the FIPS 186-2 release of DSS), describes rules and parameters for computing a digital signature of an electronic message. *Id.* The digital signature allows “the identity of the signatory” of the message to be verified, and also allows the recipient of a message to verify its integrity. *Id.* For example, Figure 1 from DSS (reproduced below) depicts an overview of the signature generation and verification functions set forth in the standard:



EX1004, FIG. 1.

25. DSS provides for the use of two keys to generate a digital signature of a message: (1) a long-term secret (private) key x that belongs to the signing party

and is used across all messages signed by that party, and (2) a short-term (also referred to as “ephemeral”) key k that is typically used just once so that every message signed by a particular party is signed with a different short-term key k . As the background of the '961 patent explains, “[t]he ephemeral private key is generated at the start of each session between a pair of correspondents” (along with a “corresponding, ephemeral public key”),” and “[o]nce the session is terminated, the ephemeral key is securely discarded and a new ephemeral key generate for a new session.” EX1001, 1:38-51.

26. DSS describes the following algorithm for generating a batch of ephemeral keys k :

3.2. ALGORITHM FOR PRECOMPUTING ONE OR MORE k AND r VALUES

This algorithm can be used to precompute k , k^{-1} , and r for m messages at a time. Algorithm:

Step 1. Choose a secret initial value for the seed-key, KKEY.

Step 2. In hexadecimal notation let
 $t = \text{EFCDA B89 98BADCFE 10325476 C3D2E1F0 67452301}$.

This is a cyclic shift of the initial value for $H_0 \parallel H_1 \parallel H_2 \parallel H_3 \parallel H_4$ in the SHS.

Step 3. For $j = 0$ to $m - 1$ do

- a. $k = G(t, \text{KKEY}) \bmod q$.
- b. Compute $k_j^{-1} = k^{-1} \bmod q$.
- c. Compute $r_j = (g^k \bmod p) \bmod q$.
- d. $\text{KKEY} = (1 + \text{KKEY} + k) \bmod 2^b$.

EX1004, 18.

27. At step 1, the algorithm for generating key k involves first obtaining a seed-key, referred to as KKEY. *Id.* A one-way function $G()$ is then applied to KKEY at Step 3(a). *Id.* The one-way function $G()$ can be a hash or a block cipher such as those specified by the Secure Secure Hash Algorithm (SHA) or Data Encryption Standard (DES), respectively. *Id.*, 17. Because the value of key k must be less than the a pre-defined prime number q and the output of $G()$ may be larger than q , DSS provides for a reduction modulo q to be performed on the output of $G()$ to ensure that the value of key k is less than q . However, as explained in the background of the '961 patent, a bias is introduced by the modular reduction due to a greater likelihood of computing values in a first interval than a second. EX1001, 2:48-55.

28. DSS also allows for generation of a batch of keys k based by performing a random selection of KKEY just once (at step 1). EX1004, 18. After the algorithm computes the initial key k in the batch, subsequent keys k are generated by updating the value of the seed-key (KKEY) using the deterministic formula from step 3(d): $KKEY = (1 + KKEY + k) \bmod 2^b$. EX1004, 14.

B. Rose

29. The Rose reference is an article that was allegedly posted to a pair of USENET newsgroups in the 1990s. *See* EX1006. Rose describes a method for

“roll[ing] a random number in the interval [0, n),” and even provides computer code for an implementation of its method:

```
/*
 * Roll a random number [0..n-1].
 * Avoid the slight bias to low numbers.
 */
int
roll(int n)
{
    long        rval;
    long        biggest;
#define BIG 0x7FFFFFFFL /* biggest value returned by random() */

    if (n == 0)
        return 0;
    biggest = (BIG / n) * n;
    do {
        rval = random();
    } while (rval >= biggest);
    return rval % n;
}
```

EX1006, 2.

30. The roll() function specifies several operations for generating a random number within a desired range [0, n). First, Rose computes the largest multiple of n that is less than or equal to BIG, where BIG is “the biggest value returned by random ().” The computed multiple is assigned to the variable ‘biggest.’ *Id.* Rose then provides a do-while loop that repeatedly determines a random number using the random() function until the output of the random() function is less than the value of ‘biggest.’ *Id.* Because the final output of the random() function, which is assigned to ‘rval’, upon exiting the loop is compared to a multiple of n and thus may be greater than the maximum value of the specified

range $(n-1)$, Rose's `roll()` function computes 'rval' modulo n and returns the result of this operation as the output of the software routine. *Id.*

C. Menezes

31. Grounds 2 and 4 of the petition cite the "Handbook of Applied Cryptography" by Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone ("Menezes"). *See* EX1005. While Menezes covers a number of topics, the petition principally relies upon disclosure from the following paragraph on the subjects of random numbers and random bit generators:

5.2 Remark (*random bits vs. random numbers*) A random bit generator can be used to generate (uniformly distributed) random numbers. For example, a random integer in the interval $[0, n]$ can be obtained by generating a random bit sequence of length $\lfloor \lg n \rfloor + 1$, and converting it to an integer; if the resulting integer exceeds n , one option is to discard it and generate a new random bit sequence.

EX1005, 69.

32. In this description, Menezes discloses that a random bit sequence is converted to an integer. I understand the petition alleges that the operation of converting a random bit sequence to an integer corresponds to a hashing operation performed on a seed value, but based on my knowledge and experience in the field, I am confident a POSITA would have found this contention to be an error. A POSITA would have known as a basic computer science or engineering matter that to obtain an integer value from a random bit generator, all it takes would be to iterate the bit generation process the proper number of times (e.g., 160 times for a

160-bit integer using a pseudo-random bit generator that produces one bit at a time) and to concatenate the outputs to form an integer. This type of formatting is straightforward and can be achieved in a variety of ways, such as using string concatenation, bit-shift and addition, and more. This is completely different from the function performed by a hash function, and orders of magnitude less computationally demanding.

33. Moreover, Menezes describes a solution for obtaining a random integer in a fundamentally different than that provided by the '961 claims. Specifically, Menezes assumes the availability of a perfect random bit generator that could generate each bit in a sequence without any bias. But a POSITA would have recognized that such a bit generator that exhibited perfect randomness is largely theoretical and unavailable for broad adoption on a wide range of systems (such as the range of systems that typically implemented DSS). By starting with a perfect random bit generator, Menezes had no need to perform additional transformations on the random integer produced by the bit generator. In reality, however, Menezes' approach is impractical, and so the '961 patent provides a materially different solution for generating an unbiased output within a desired range by first obtaining a seed value from a random number generator, and then hashing the seed value, thereby removing or obscuring any bias that is often present in real-world random number generators.

D. Schneier

34. Each ground of the petition also relies upon the textbook *Applied Cryptography* (2nd Ed.) by Bruce Schneier (“Schneier”). See EX1008. The petition cites Schneier for its discussion of “various techniques for generating random and pseudorandom numbers” and its alleged description of the “benefit of [true random numbers] over pseudorandom numbers in cryptographic key generation.” Pet., 27 (citing EX1008, 146-153).

VIII. THE PETITION FAILS TO ESTABLISH THAT THE CITED REFERENCES WOULD HAVE PROVIDED FOR REPETITION OF THE REQUISITE OPERATIONS RECITED IN INDEPENDENT CLAIMS 1, 15, AND 23

A. The '961 Claims Require Repetition Of Operations For Both Random Seed Value Generation And Hashing Of The Seed Value If A Preceding Hashed Value Is Not Less Than Said Order Q

35. I understand that independent claim 1 of the '961 patent recites the following language:

generating a seed value SV from a random number generator;

performing a hash function $H(\)$ on said seed value SV to provide an output $H(SV)$;

determining whether said output $H(SV)$ is less than said order q prior to reducing mod q ;

accepting said output $H(SV)$ for use as said key k if the value of said output $H(SV)$ is less than said order q ;

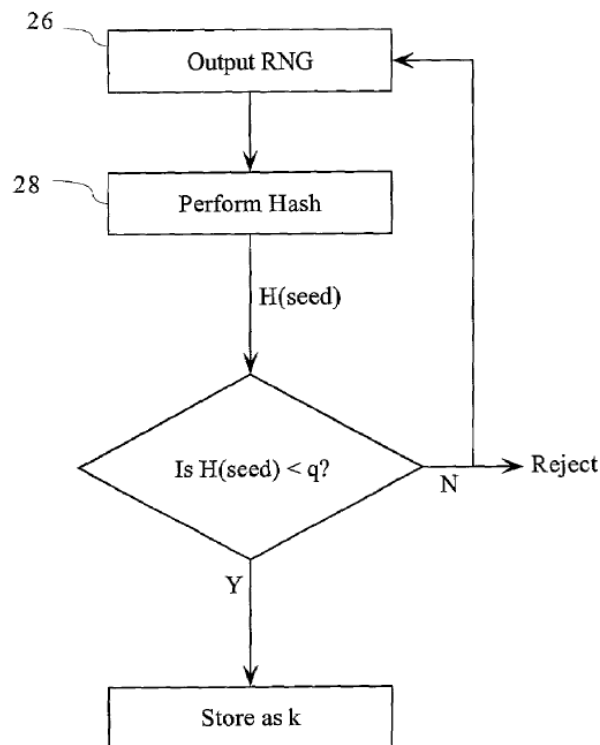
rejecting said output $H(SV)$ as said key if said value is not less than said order q ;

if said output $H(SV)$ is rejected, repeating said method;

EX1001, 5:36-51 (emphasis added).

36. Independent claims 15 and 23 recite corresponding language to the language quoted above from claim 1. *Id.*, 6:62-64 (claim 15), 7:31-8:6 (claim 23).

37. Notably, each of claims 1, 15, and 23 provides that “if said output $H(SV)$ is rejected,” then the claimed method is to be repeated. The method that is to be repeated entails multiple operations as expressly recited in the claims, including “generating a seed value SV from a random number generator” and “performing a hash function $H()$ on said seed value SV to provide an output $H(SV)$.” *Id.*, 5:37-40. In other words, if in a first iteration of the method, the output $H(SV)$ is rejected due to the value of $H(SV)$ not being less than the order q , the ’961 claims provide that the method must repeat such that, in a second iteration, a new seed value SV would be generated from a random number generator and the hash function $H()$ would be performed on the new seed value SV to provide another output $H(SV)$. This pattern of operations is confirmed by Figure 2 of the ’961 patent, in which, after the rejection of $H(seed)$, the flowchart returns to box 26 where a new seed value is obtained from the output of the random number generator (RNG):



EX1001, FIG. 2; *see also id.*, 4:11-17 (“The resultant output $H(\text{seed})$ is tested against the value of q and a decision made based on the relative values. If $H(\text{seed}) < q$ then it is accepted for use as k . If not, the value is rejected and the random number generator is conditioned to generate a new value which is again hashed by the function 28 and tested. This loop continues until a satisfactory value is obtained.”).

38. Despite the plain requirements of claims 1, 15, and 23, as I explain below, the petition has failed to establish that it would have been obvious to provide for repetition of both operations for “generating a seed value SV from a random number generator” and “performing a hash function $H()$ on said seed value SV to provide an output $H(SV)$.”

B. The References Cited In Grounds 1 And 3 Do Not Disclose Repetition Of The Required Operations

39. Based on my review of Grounds 1 and 3, I believe the petition has failed to show that the relied upon combinations of prior art disclose the recited features of claims 1, 15, and 23 of “if said output $H(SV)$ is rejected, repeating said method,” including repeating the steps of “generating a seed value SV from a random number generator” and “performing a hash function $H()$ on said seed value SV to provide an output $H(SV)$.” *See* EX1001, 5:36-48; *supra*, Section VIII.A. Moreover, the petition’s assertion that it would have been obvious to provide such repetition in combinations based on DSS, Rose, and Schneier is unsupported, and I do not believe the evidence submitted with the petition establishes that the cited combinations would have realized these features as provided in claims 1, 15, and 23 of the ’961 patent.

40. To start, DSS fails to disclose the repetition of a method for key generation in a manner analogous to the features recited in the ’961 claims. This is even admitted in the petition, which points out that the algorithm for key generation in DSS employs modular reduction (Step 3A) to obtain a value for k that is less than q —rather than, for example, employing an iterative technique to search for a value within the allowed range. EX1004, 18; Pet., 35-36.

3.2. ALGORITHM FOR PRECOMPUTING ONE OR MORE k AND r VALUES

This algorithm can be used to precompute k , k^{-1} , and r for m messages at a time. Algorithm:

Step 1. Choose a secret initial value for the seed-key, KKEY.

Step 2. In hexadecimal notation let

$t = \text{EFCDAB89 98BADCFE 10325476 C3D2E1F0 67452301}$.

This is a cyclic shift of the initial value for $H_0 \parallel H_1 \parallel H_2 \parallel H_3 \parallel H_4$ in the SHS.

Step 3. For $j = 0$ to $m - 1$ do

a. $k = G(t, \text{KKEY}) \bmod q$.

b. Compute $k_j^{-1} = k^{-1} \bmod q$.

c. Compute $r_j = (g^k \bmod p) \bmod q$.

d. $\text{KKEY} = (1 + \text{KKEY} + k) \bmod 2^b$.

EX1004, 18 (highlighting added).

41. The petition then turns to Rose as allegedly providing inspiration for an iterative technique for identifying a value within a desired range. Pet. 36-44.

For reference, the computer code disclosed in Rose for implementing his technique is reproduced below:

```

/*
 * Roll a random number [0..n-1].
 * Avoid the slight bias to low numbers.
 */
int
roll(int n)
{
    long    rval;
    long    biggest;
#define BIG 0x7FFFFFFFL /* biggest value returned by random() */

    if (n == 0)
        return 0;
    biggest = (BIG / n) * n;
    do {
        rval = random();
    } while (rval >= biggest);
    return rval % n;
}

```

EX1006, 2.

42. As shown above, Rose's roll() function was concerned with returning a random value within the range $[0, n-1]$, where n was provided as an input parameter to the function. Rose never purported to apply his technique to cryptographic key generation, and certainly never suggested that his technique would be applicable for the purpose of determining a value for the key k in DSS. There is a straightforward reason for this. Namely, a POSITA would have recognized the fact that Rose's function would have been ill-suited for use in a cryptographic system (e.g., a system implementing key generation for DSS) based at least on the nature of the particular random() function employed in the code. In the context of Rose, random() represents the randomness function that was provided at the time of Rose by programming languages such as C++, which was

an insecure random generator that did not involve the use of a one-way function and, much less, a hash function. Moreover, Rose's function does not iteratively perform the same combination of operations recited by claims 1, 15, and 23 of the '961 patent (i.e., operations that are also not iteratively performed in DSS) because Rose merely repeats, through the do-while loop, evaluation of the random() function until an acceptable value is obtained. In contrast to the '961 claims, Rose does not disclose a hashing operation that follows generation of a seed value from a random number generator, let alone performing both of those operations in a method that is repeated when the result of a hashing operation is not less than a specific value (e.g., order q). The petition does not rely on Schneier (Ground 1), or Schneier, Rao or Floyd (Ground 3) for the repetition aspects of the challenged claims, and based on my review, the cited portions of these references also do not disclose these features.

43. Despite none of the cited references in Grounds 1 or 3 disclosing the series of operations which are subject to repetitive performance as recited in the '961 claims, I understand the petition alleges that the teachings of DSS, Schneier, and Rose (Ground 1) or DSS, Schneier, Rose, Rao, and Floyd (Ground 3), when considered in their respective combinations, nonetheless would have provided these features. Pet., 40-44. For example, the petition contends that it

would have been obvious to modify DSS in view of Rose and Schneier to implement the following operations:

(1) Generate a random number for KKEY in accordance with DSS alone or in combination with Schneier (the claimed “**seed value SV**”), as explained for claim 1[a];

(2) perform a hash of KKEY, $G(t, KKEY)$, according to DSS (the claimed “**output H(SV)**”), as explained for claim 1[b];

(3) determine whether $G(t, KKEY)$ (“**output H(SV)**”) is less than **order q** (*i.e.* the value q), according to the teachings of Rose;

(4) accept $G(t, KKEY)$ (“**output H(SV)**”) if $G(t, KKEY)$ **is** less than q , according to the teachings of Rose;

(5) reject $G(t, KKEY)$ if it is **not** less than q , by repeating the method and going back to step (1), according to the teachings of Rose; and

(6) if $G(t, KKEY)$ (“**output H(SV)**”) was accepted in step (4), perform a ‘mod q ’ operation on $G(t, KKEY)$, pursuant to the teachings of Rose, and provide it as a key **k** for use in performing a cryptographic function, in accordance with DSS.

Pet., 41-42.

44. The petition also cites the declaration of Dr. Katz, which provides a pseudo-code representation of the petition’s proposed DSS-Schneier-Rose combination based on a newly theorized modification of Rose’s computer code. I have copied this pseudo-code below (with annotations), for reference:

```

generate_k (q)          // q is the max value
{
    do {

        // Get a number from real random number generator
        KKEY = [Output of random number generator];

        // Hash the random number
        H_KKEY = G(t, KKEY);

    } while (H_KKEY >= q); // Roll again unless
                          // H(KKEY) < q

    return H_KKEY mod q;  // Value for k
}

```

Petition fails to justify inclusion of both functions within the do-while loop

EX1002, ¶158 (highlighting and annotations added).

45. As shown above, the Petitioner and Dr. Katz assume that applying teachings from Rose to DSS would have resulted in a system in which, for each iteration until an acceptable value is determined, operations are performed for (1) generating KKEY (allegedly a “seed value”) from the output of a random number generator and (2) hashing KKEY to produce an output H_KKEY (allegedly “output H(SV)”). This assumption is flawed and contrary to Rose’s actual teaching.

46. Based on my review, the petition fails to support its claim that the resulting DSS-Schneier-Rose combination would have achieved the specific iterative approach recited in claims 1, 15, and 23 of the ’961 patent. As I explained above, none of the cited references in Ground 1 individually disclose such an approach. While Rose discloses an iterative process, it only provides for iterative evaluation of a random() function—but *never* describes a distinct hashing

operation that follows generation of a seed value from a random number generator, as claimed. While the petition imports a hashing operation into the method based on DSS (e.g., by applying the SHA hash function to evaluate $G(t, KKEY)$), the petition fails to address with any explanation or evidence why a POSITA allegedly would have chosen to both (i) evaluate a random number generator to obtain a seed value and (ii) hash the seed value with each iteration of the method. *See, e.g., Pet.*, 40-44. Based on my knowledge and experience in the field, and my review of the material identified above (*see* Section III), this aspect of the petition's proposed combination is supported nowhere in the references themselves, and worse, a POSITA would have recognized that it was in fact unnecessary to repeat both operations. Moreover, teachings from both DSS and Menezes indicate an implicit rejection of the petition's proposed combination.

47. For example, due to well-known properties of cryptographic hash functions such as SHA, a POSITA would have recognized that it would not have been required to obtain a new random input to the hash function each time the function was evaluated while searching for an output of the hash function that falls below a specified value (e.g., q). Instead, a POSITA would have known that even by updating the random input using an arithmetic transformation, such as adding 1 to the random input, would have been sufficient. Even with a simple arithmetic transformation, the resulting hash function output would be statistically

uncorrelated with the previous hash function output. This follows from the uniformity of the hash function and the observation that the probability of any input to the hash function mapping to each of the possible outputs of the hash function being approximately equal. Thus, consider an example where a first randomly generated seed-key $KKEY_1$ is provided as input to the SHA hash function from DSS, and the result of the hash is a value that is not less than said order q . The probability of identifying a hashed value that is less than said order q in a next iteration would be approximately the same regardless of whether (i) the input to the hash function was a new seed-key $KKEY_2$ obtained from a random number generator, (ii) the input to the hash function was derived by applying a deterministic function to update $KKEY_1$ (e.g., incrementing $KKEY_1$ by a fixed amount (even simply incrementing by one)), (iii) the input to the hash function was the preceding output of the hash function (e.g., $G(t, KKEY_1)$) or was derived deterministically from the preceding output of the hash function, or (iv) the input to the hash function was derived deterministically from the preceding output of the hash function and the preceding input $KKEY_1$ to the hash function. In view of these multiple possible avenues (and others) for updating the input to the hash function with each iteration, the petition errs to the extent it argues that generating a seed value with a random number generator with each iteration would necessarily have been present in the combination.

48. The petition's proposed combination also fails to follow the suggested manner of updating the input or seed to a repeatedly evaluated hash function by deterministically revising the input based at least on the preceding value of the input. For example, DSS describes in Section 3.2 an algorithm for precomputing a batch of keys k :

3.2. ALGORITHM FOR PRECOMPUTING ONE OR MORE k AND r VALUES

This algorithm can be used to precompute k , k^{-1} , and r for m messages at a time. Algorithm:

Step 1. Choose a secret initial value for the seed-key, KKEY. ← **Independently initialize KKEY once (e.g., using a random number generator)**

Step 2. In hexadecimal notation let
 $t = \text{EFC DAB89 98BADC FE 10325476 C3D2E1F0 67452301}.$

This is a cyclic shift of the initial value for $H_0 \parallel H_1 \parallel H_2 \parallel H_3 \parallel H_4$ in the SHS.

Step 3. For $j = 0$ to $m - 1$ do

- a. $k = G(t, \text{KKEY}) \bmod q.$
- b. Compute $k_j^{-1} = k^{-1} \bmod q.$
- c. Compute $r_j = (g^k \bmod p) \bmod q.$
- d. **$\text{KKEY} = (1 + \text{KKEY} + k) \bmod 2^b.$** ← **Deterministically update the value of KKEY based on its previous value for each subsequent key**

EX1004, 18 (highlighting and annotations added).

49. Through this algorithm, DSS expressed a preference for deterministically updating the seed-key KKEY that is the input to the hash function represented by $G(t, \text{KKEY})$ (e.g., the SHA function). At Step 1, the seed-key KKEY is independently initialized only once. Thereafter, at Step 3d, for each key k that is to be generated in the batch, KKEY is updated based on the preceding values of KKEY and k according to the function $\text{KKEY} = (1 + \text{KKEY} + k) \bmod 2^b$.

Although DSS's algorithm here updates KKEY in a different context than the context of the petition's proposed DSS-Rose-Schneier combination so as to generate new KKEY values for each key k in the batch rather than generating new KKEY values to find a hash of KKEY that is less than q , a POSITA would have recognized that the effect of deterministically updating KKEY rather than returning to Step 1 to independently re-generate KKEY is largely the same. Indeed, an even simpler deterministic function could be applied to update KKEY in each iteration of the alleged DSS-Rose-Schneier combination (e.g., incrementing KKEY by fixed value) since the new value is not being applied for generation of another key in the batch.

50. Similarly, Menezes (which the petition cites in Grounds 2 and 4), teaches in Section 5.3 that a one-way hash function such as SHA can be repeatedly evaluated to obtain a series of pseudorandom numbers with low correlation between them simply by incrementing the input to the function by one—an approach that certainly would have been suitable for the purpose of identifying a hashed value that fell below a threshold q (even if some correlation existed).

5.3 Pseudorandom bit generation

A one-way function f (Definition 1.12) can be utilized to generate pseudorandom bit sequences (Definition 5.3) by first selecting a random seed s , and then applying the function to the sequence of values $s, s+1, s+2, \dots$; the output sequence is $f(s), f(s+1), f(s+2), \dots$. Depending on the properties of the one-way function used, it may be necessary to only keep a few bits of the output values $f(s+i)$ in order to remove possible correlations between successive values. Examples of suitable one-way functions f include a cryptographic hash function such as SHA-1 (Algorithm 9.53), or a block cipher such as DES (§7.4) with secret key k .

EX1005, 72 (highlighting added).

51. Based on my review, the petition fails to explain why its proposed combination chose to repeatedly re-generate KKEY using a random number generator at each iteration rather than applying other preferable choices including those described in both DSS and Menezes. Furthermore, the petition failed to explain why its approach disregards certain advantages that a POSITA would have recognized would have been achieved through use of other techniques (e.g., simple deterministic functions) for updating the seed value in a DSS-Schneier-Rose combination rather than through use of a random number generator—especially a true or otherwise robust random number generator such as that described in Schneier (and applied in the petition’s combination). True random number generators produce randomness that is derived from chaotic physical phenomena (such as air turbulence, which cannot be predicted). However, typical true random generators often require significant time to produce high-quality output (e.g., Schneier describes a random-number generator that uses the computer’s disk drive

as a source of randomness, but it was only “able to collect about 100 bits per minute” (EX1008, 149)). For these reasons, true random number generators were often used to generate seed values, where these seed values were later arithmetically manipulated and input to a hash function or other one-way function. The petition erroneously assumes a POSITA would have been motivated to re-run a true random generator for each key he or she wanted to produce, but such an assumption is unfounded and contrary to the reality facing a POSITA at the time. Similarly, the petition erroneously assumed a POSITA would have been motivated to generate a random number from a cryptographic pseudo-random generator and then process this using a hash function or another one-way function, but this assumption is also flawed. There simply would have been no reason to do both, and a POSITA would have readily understood this fact. Since the cost of generating a sufficiently long output with a high-quality pseudo-random function would have been at least the same as performing a hash function, and significantly higher than an arithmetic manipulation of the seed value of the kind performed in DSS, a POSITA would have recognized the avoidable redundancy/problem of trying to re-run the pseudo-random function because it was unnecessary.

52. For each of the reasons described above, I do not believe that the DSS-Schneier-Rose combination provides all features of claims 1, 15, and 23 of the '961 patent, especially where the petition fails to establish that a POSITA

would have implemented repetitive re-generation of a seed key using a random number generator in the manner recited in claims 1, 15, and 23.

C. Grounds 2 And 4 Do Not Disclose Repetition Of The Required Operations

53. Based on my review of Grounds 2 and 4, I believe the petition has failed to show that the relied upon combinations of prior art disclose the recited features of claims 1, 15, and 23 of “if said output $H(SV)$ is rejected, repeating said method,” including repeating the steps of “generating a seed value SV from a random number generator” and “performing a hash function $H()$ on said seed value SV to provide an output $H(SV)$.” *See* EX1001, 5:36-48; *supra*, Section VIII.A. Moreover, the petition’s assertion that it would have been obvious to provide such repetition in combinations based on DSS, Menezes, and Schneier is unsupported, and I do not believe the evidence submitted with the petition establishes that the cited combinations would have realized these features as provided in claims 1, 15, and 23 of the ’961 patent.

54. To start, as I previously explained, DSS fails to disclose the repetition of a method for key generation in a manner analogous to the features recited in the ’961 claims. This is even admitted in the petition, which points out that the algorithm for key generation in DSS employs modular reduction (Step 3A) to obtain a value for k that is less than q —rather than, for example, employing an

iterative technique to search for a value within the allowed range. EX1004, 18; Pet., 35-36.

3.2. ALGORITHM FOR PRECOMPUTING ONE OR MORE k AND r VALUES

This algorithm can be used to precompute k , k^{-1} , and r for m messages at a time. Algorithm:

Step 1. Choose a secret initial value for the seed-key, KKEY.

Step 2. In hexadecimal notation let
 $t = \text{EFCDA B89 98BADCFE 10325476 C3D2E1F0 67452301}$.

This is a cyclic shift of the initial value for $H_0 \parallel H_1 \parallel H_2 \parallel H_3 \parallel H_4$ in the SHS.

Step 3. For $j = 0$ to $m - 1$ do

- a. $k = G(t, \text{KKEY}) \bmod q$.
- b. Compute $k_j^{-1} = k^{-1} \bmod q$.
- c. Compute $r_j = (g^k \bmod p) \bmod q$.
- d. $\text{KKEY} = (1 + \text{KKEY} + k) \bmod 2^b$.

EX1004, 18 (highlighting added).

55. The petition then turns to Menezes as allegedly providing inspiration for an iterative technique for identifying a value within a desired range. Pet., 50-56. Recall that Menezes disclosed that “[a] random bit generat can be used to generate (uniformly distributed) random numbers” and “[f]or example, a random integer in the interval $[0, n]$ can be obtained by generating a random bit sequence of length $\lg n + 1$, and converting it to an integer; if the resulting integer exceeds n , one option is to discard it and generate a new random bit sequence.” EX1005, 59.

56. Thus, similar to Rose, Menezes' cited disclosure was concerned with using a random bit generator to select a random number within a desired range. Yet, in contrast to the '961 claims, the relied upon portion of Menezes never discusses a hashing operation that follows generation of a seed value from a random number generator, let alone performing both of those operations in a method that is repeated when the result of a hashing operation is not less than a specific value (e.g., order q). The petition does not rely on Schneier (Ground 2), or Schneier, Rao or Floyd (Ground 4) for the repetition aspects of the challenged claims, and based on my review, the cited portions of these references also do not disclose these features.

57. Despite none of the cited references in Grounds 2 or 4 disclosing the series of operations which are subject to repetitive performance as recited in the '961 claims, I understand the petition alleges that the teachings of DSS, Menezes, and Rose (Ground 2) or DSS, Schneier, Menezes, Rao, and Floyd (Ground 4), when considered in their respective combinations, nonetheless would have provided these features. Pet., 50-56. For example, the petition contends that it would have been obvious to modify DSS in view of Menezes and Schneier to implement the following operations:

(1) Generate a random number for KKEY according to DSS alone or in combination with Schneier (the claimed "seed value SV"), as explained for claim 1[a];

(2) compute a hash of KKEY, $G(t, KKEY)$, according to DSS (the claimed “**output $H(SV)$** ”), as explained for claim 1[b];

(3) determine whether $G(t, KKEY)$ (“**output $H(SV)$** ”) is less than **order q** , according to the teachings of Menezes;

(4) accept $G(t, KKEY)$ (“**output $H(SV)$** ”) if $G(t, KKEY)$ **is** less than q , according to the teachings of Menezes; and

(5) reject $G(t, KKEY)$ if it is **not** less than q , by repeating the method and going back to step (1), according to the teachings of Menezes.

Pet., 53-54 (emphases in original) (citing EX1002, ¶185).

58. As shown above, the petition alleges that a predictable application of teachings from Menezes to DSS would have resulted in a system in which, for each iteration until an acceptable value is determined, operations are performed for (1) generating KKEY (allegedly a “seed value SV”) from the output of a random number generator and (2) hashing KKEY to produce an output H_KKEY (allegedly “output $H(SV)$ ”). I disagree.

59. Based on my review, the petition fails to support its claim that the resulting DSS-Schneier-Menezes combination would have achieved the specific iterative approach recited in claims 1, 15, and 23 of the ’961 patent. As I explained above, none of the cited references in Grounds 2 or 4 individually disclose such an approach. While Menezes discloses an iterative process, it only provides for iterative evaluation of an integer converted from a random bit sequence—but *never*

describes a distinct hashing operation that follows generation of a seed value from a random number generator, as claimed. While the petition imports a hashing operation into the method based on DSS (e.g., by applying the SHA hash function to evaluate $G(t, KKEY)$), the petition fails to address with any explanation or evidence why a POSITA allegedly would have chosen to both (i) evaluate a random number generator to obtain a seed value and (ii) hash the seed value with each iteration of the method. *See, e.g., Pet.*, 50-56. Based on my knowledge and experience in the field, and my review of the material identified above (*see* Section III), this aspect of the petition's proposed combination is supported nowhere in the references themselves, and worse, a POSITA would have recognized that it was in fact unnecessary to repeat both operations. Moreover, teachings from both DSS and Menezes indicate an implicit rejection of the petition's proposed combination.

60. For example, due to well-known properties of cryptographic hash functions such as SHA, a POSITA would have recognized that it would not have been required to obtain a new random input to the hash function each time the function was evaluated while searching for an output of the hash function that falls below a specified value (e.g., q). Instead, a POSITA would have known that even by updating the random input using an arithmetic transformation, such as adding 1 to the random input, would have been sufficient. Even with a simple arithmetic transformation, the resulting hash function output would be statistically

uncorrelated with the previous hash function output. This follows from the uniformity of the hash function and the observation that the probability of any input to the hash function mapping to each of the possible outputs of the hash function being approximately equal. Thus, consider an example where a first randomly generated seed-key $KKEY_1$ is provided as input to the SHA hash function from DSS, and the result of the hash is a value that is not less than said order q . The probability of identifying a hashed value that *is* less than said order q in a next iteration would be approximately the same regardless of whether (i) the input to the hash function was a new seed-key $KKEY_2$ obtained from a random number generator, (ii) the input to the hash function was derived by applying a deterministic function to update $KKEY_1$ (e.g., incrementing $KKEY_1$ by a fixed amount (even simply incrementing by one)), (iii) the input to the hash function was the preceding output of the hash function (e.g., $G(t, KKEY_1)$) or was derived deterministically from the preceding output of the hash function, or (iv) the input to the hash function was derived deterministically from the preceding output of the hash function and the preceding input $KKEY_1$ to the hash function. In view of these multiple possible avenues (and others) for updating the input to the hash function with each iteration, the petition errs to the extent it argues that generating a seed value with a random number generator with each iteration would necessarily have been present in the combination.

61. The petition's proposed combination also fails to follow the suggested manner of updating the input to a repeatedly evaluated hash function by deterministically revising the input based at least on the preceding value of the input. For example, DSS describes in Section 3.2 an algorithm for precomputing a batch of keys k :

3.2. ALGORITHM FOR PRECOMPUTING ONE OR MORE k AND r VALUES

This algorithm can be used to precompute k , k^{-1} , and r for m messages at a time. Algorithm:

Step 1. Choose a secret initial value for the seed-key, KKEY. ← **Independently initialize KKEY once (e.g., using a random number generator)**

Step 2. In hexadecimal notation let
 $t = \text{EFCDA B89 98BADCFE 10325476 C3D2E1F0 67452301}.$

This is a cyclic shift of the initial value for $H_0 \parallel H_1 \parallel H_2 \parallel H_3 \parallel H_4$ in the SHS.

Step 3. For $j = 0$ to $m - 1$ do

- a. $k = G(t, \text{KKEY}) \bmod q.$
- b. Compute $k_j^{-1} = k^{-1} \bmod q.$
- c. Compute $r_j = (g^k \bmod p) \bmod q.$
- d. $\text{KKEY} = (1 + \text{KKEY} + k) \bmod 2^b.$** ← **Deterministically update the value of KKEY based on its previous value for each subsequent key**

EX1004, 18 (highlighting and annotations added).

62. Through this algorithm, DSS expressed a preference for deterministically updating the seed-key KKEY that is the input to the hash function represented by $G(t, \text{KKEY})$ (e.g., the SHA function). At Step 1, the seed-key KKEY is independently initialized only once. Thereafter, at Step 3d, for each key k that is to be generated in the batch, KKEY is updated based on the preceding values of KKEY and k according to the function $\text{KKEY} = (1 + \text{KKEY} + k) \bmod 2^b$.

Although DSS's algorithm here updates KKEY in a different context than the context of the petition's proposed DSS-Menezes-Schneier combination so as to generate new KKEY values for each key k in the batch rather than generating new KKEY values to find a hash of KKEY that is less than q , a POSITA would have recognized that the effect of deterministically updating KKEY rather than returning to Step 1 to independently re-generate KKEY is largely the same. Indeed, an even simpler deterministic function could be applied to update KKEY in each iteration of the alleged DSS-Menezes-Schneier combination (e.g., incrementing KKEY by fixed value) since the new value is not being applied for generation of another key in the batch.

63. Similarly, Menezes teaches in Section 5.3 that a one-way hash function such as SHA can be repeatedly evaluated to obtain a series of pseudorandom numbers with low correlation between them simply by incrementing the input to the function by one—an approach that certainly would have been suitable for the purpose of identifying a hashed value that fell below a threshold q (even if some correlation existed).

5.3 Pseudorandom bit generation

A one-way function f (Definition 1.12) can be utilized to generate pseudorandom bit sequences (Definition 5.3) by first selecting a random seed s , and then applying the function to the sequence of values $s, s+1, s+2, \dots$; the output sequence is $f(s), f(s+1), f(s+2), \dots$. Depending on the properties of the one-way function used, it may be necessary to only keep a few bits of the output values $f(s+i)$ in order to remove possible correlations between successive values. Examples of suitable one-way functions f include a cryptographic hash function such as SHA-1 (Algorithm 9.53), or a block cipher such as DES (§7.4) with secret key k .

EX1005, 72 (highlighting added).

64. Based on my review, the petition fails to explain why its proposed combination chose to repeatedly re-generate KKEY using a random number generator at each iteration rather than applying other preferable choices including those described in both DSS and Menezes. Furthermore, the petition failed to explain why its approach disregards certain advantages that a POSITA would have recognized would have been achieved through use of other techniques (e.g., simple deterministic functions) for updating the seed value in a DSS-Schneier-Menezes combination rather than through use of a random number generator—especially a true and high-quality random number generator such as that described in Schneier (and applied in the petition’s combination). True random number generators produce randomness that is derived from chaotic physical phenomena (such as air turbulence, which cannot be predicted). However, typical true random generators often require significant time to produce high-quality output (e.g., Schneier describes a random-number generator that uses the computer’s disk drive as a

source of randomness, but it was only “able to collect about 100 bits per minute” (EX1008, 149)). For these reasons, true random number generators were often used to generate seed values, where these seed values were later arithmetically manipulated and input to a hash function or other one-way function. Again, the petition erroneously assumes a POSITA would have been motivated to re-run a true random generator for each key he or she wanted to produce, but such an assumption is unfounded and contrary to the reality facing a POSITA at that time. Similarly, the petition erroneously assumes a POSITA would have been motivated to generate a random number from a cryptographic pseudo-random generator and then process this using a hash function or another one-way function, but this assumption is also flawed. There simply would have been no reason to do both, and POSITA would have readily understood this fact. Since the cost of generating a sufficiently long output with a high-quality pseudo-random function would have been at least the same as performing a hash function, and significantly higher than an arithmetic manipulation of the seed value of the kind performed in DSS, a POSITA would have recognized the avoidable redundancy/problem of trying to re-run the pseudo-random function because it was unnecessary.

65. For each of the reasons described above, I do not believe that the DSS-Schneier-Menezes combination provides all features of claims 1, 15, and 23 of the '961 patent, especially where the petition fails to establish that a POSITA

would have implemented repetitive re-generation of a seed key using a random number generator in the manner recited in claims 1, 15, and 23.

IX. SUBSEQUENT DSS RELEASES DEMONSTRATE THAT THE PETITIONER'S THEORY WAS ROOTED IN HINDSIGHT, NOT THE REALITY AT THE TIME

66. I understand the petition alleges that Rose's and Menezes' techniques would have been "obvious to try," and that Dr. Katz assumes that there were only two solutions to the problem of obtaining random numbers within a desired range that is smaller than the range of the random number generator itself: "(1) perform a modular reduction directly on the generated random number (which Rose criticizes), or (2) reject any generated number that falls outside the desired range (which Rose encourages)." Pet. 44; EX1002, ¶158. To the extent Petitioner and Dr. Katz assume that only one viable solution exists to obtaining a random number in a desired range without introducing bias in the selection of that number, this assumption is certainly flawed. Based on my knowledge and experience in the field, it is clear that the modulo bias problem in fact could have been addressed using a variety of options not contemplated by the framing of the petition's limited choices, including solutions within either category (e.g., modular reduction, and trial-and-error), both categories, or neither.

67. My analysis in this regard is confirmed by the fact that subsequent releases of DSS were updated to address the modular bias problem identified by

Dr. Bleichenbacher, but none of them adopted solutions similar to either Rose's or Menezes' specific approaches. For example, in the following sub-sections, I describe three approaches for mitigating modular bias in the generation of the ephemeral key k taken by subsequent DSS releases—and note their differences with the Rose, Menezes, and the petition's proposed combinations.

A. FIPS 186-2 (October 2001)

68. First, I understand that in October 2000—approximately ten months after the effective filing date of the '961 patent—the National Institute of Standards and Technology released a change notice to the FIPS 186-2 DSS standard. *See* EX2002. The change notice constituted an updated release of DSS, and this release prescribed a new algorithm for generating the user's per-message secret key k , as shown below:

2. Revised Algorithm for Precomputing one or More k and r Values (Appendix 3.2 of FIPS 186-2)

This algorithm can be used to precompute k , k^{-1} , and r for m messages at a time. Note that implementation of the DSA with precomputation may be covered by U.S. and foreign patents.

Step 1. Choose a secret initial value for the seed-key, $KKEY$.

Step 2. In hexadecimal notation let

$t = \text{EFCDAB89 98BADCFE 10325476 C3D2E1F0 67452301.}$

This is a cyclic shift of the initial value for $H_0 \parallel H_1 \parallel H_2 \parallel H_3 \parallel H_4$ in the SHS.

Step 3. For $j = 0$ to $m - 1$ do

3.1 For $i = 0$ to 1 do

a. $w_i = G(t, KKEY)$

b. $KKEY = (1 + KKEY + w_i) \bmod 2^b$

3.2 $k = (w_0 \parallel w_1) \bmod q$

EX2002, 73 (highlighting added).

69. The algorithm prescribed by the FIPS 186-2 October 2001 change notice sought to mitigate the security vulnerability identified by Dr. Bleichenbacher from the FIPS 186-1 and original 186-2 releases by doubling the length of the dividend that was subject to the modular reduction operation. That is, this subsequent release of DSS *did not* abandon the modular reduction approach altogether as proposed by the Petitioner and Dr. Katz, but instead modified its parameters to reduce the effect of any bias that it could impose. In particular, the dividend for computing k in the earlier versions of DSS was simply the output of $G(t, KKEY)$, which was 160 bits long and the same length as q itself. *See*

EX1004, 17-18. But in the October 2001 revision of FIPS 186-2, the dividend length was doubled to 320 bits by concatenating two computations of $G(t, KKEY)$ (i.e., w_0 and w_1), thereby increasing the range of possible values for the dividend from 2^{160} to 2^{320} . EX2002, 73. The result of increasing the dividend in this manner was to substantially decrease the impact of bias from the modular selection to a statistically insignificant level that could not practically be used to compromise the user's long-term private key. For example, in a 2003 publication, Vaudenay estimates that the resulting bias in DSS changes from $q/(\pi N) \sin(\pi N/q) e^{i\pi (N-1)/q}$ to $q/(\pi N^2) \sin(\pi N^2/q) e^{i\pi (N^2-1)/q}$. Since q and N are of the same approximate size, this changes the bias from being on the order of $1/\pi$ to being on the order of $1/(\pi N)$. Given that N is a 160-bit number, this is a truly dramatic reduction of bias representing a reduction by billions and billions. See EX2005, pp. 4-5. Thus, the October 2001 revision of FIPS 186-2 shows that solutions to mitigating bias that were categorically different from the techniques cited from Rose and Menzes were available despite never being acknowledge by either the petition or Dr. Katz.

B. FIPS 186-3 (June 2009)

70. Next, I understand that the DSS standard was subsequently updated in June 2009 with the FIPS 186-3 release. See EX2003. FIPS 186-3 again updated the prescribed methods for generating a user's per-message private key k , and specifically provided two options for generating such keys. EX2003, 48-50. The

first method addressed Bleichenbacher's vulnerability in a similar manner to the FIPS 186-2 October 2001 release, namely by increasing the length of the dividend prior to its modular reduction to obtain the key k :

Process:

1. $N = \text{len}(q); L = \text{len}(p).$

Comment: Check that the (L, N) pair is specified in Section 4.2.

2. If the (L, N) pair is invalid, then return an **ERROR** indication, *Invalid_k*, and *Invalid_k_inverse*.

3. *requested_security_strength* = the security strength associated with the (L, N) pair; see SP 800-57.

4. Obtain a string of $N+64$ *returned_bits* from an **RBG** with a security strength of *requested_security_strength* or more. If an **ERROR** indication is returned, then return an **ERROR** indication, *Invalid_k*, and *Invalid_k_inverse*.

5. Convert *returned_bits* to the (non-negative) integer c (see Appendix C.2.1).

6. $k = (c \bmod (q-1)) + 1.$

7. $(\text{status}, k^{-1}) = \text{inverse}(k, q).$

8. Return *status*, k , and k^{-1} .

EX2003, 49 (highlighting added).

71. As shown in the algorithm above, an integer c is formed from a random bit generator (RBG) that is 64 bits longer than the length of q . *Id.* When c is then applied as the dividend in the modular reduction (shown at step 6), the relative lengths of c and q ensure that any bias introduced by the modulo operation is sufficiently small to prevent any practical security vulnerability. EX2003, 48 (“In this method, 64 more bits are requested from the RBG than are needed for k so that bias produced by the mod function in step 6 is not readily apparent.”). This

method is again similar to the original DSS approach and categorically different from either Rose or Menezes.

72. In the method presented by the second option in FIPS 186-3, “a random number is obtained and tested to determine that it will produce a value of k in the correct range,” and “[i]f k is out-of-range, another random number is obtained” so that “the process is iterated until an acceptable value of k is obtained.” *Id.*, 49.

Process:

1. $N = \text{len}(q); L = \text{len}(p).$

Comment: Check that the (L, N) pair is specified in Section 4.2).

2. If the (L, N) pair is invalid, then return an **ERROR** indication, *Invalid_k*, and *Invalid_k inverse*.

3. *requested_security_strength* = the security strength associated with the (L, N) pair; see SP 800-57.

4. Obtain a string of N *returned_bits* from an **RBG** with a security strength of *requested_security_strength* or more. If an **ERROR** indication is returned, then return an **ERROR** indication, *Invalid_k*, and *Invalid_k inverse*.

5. Convert *returned_bits* to the (non-negative) integer c (see Appendix C.2.1).

6. If $(c > q-2)$, then go to step 4.

7. $k = c + 1.$

8. $(\text{status}, k^{-1}) = \text{inverse}(k, q).$

9. Return *status*, k , and k^{-1} .

EX2003, 50².

² *Supra*, footnote 2.

73. Here, this second option prescribed by FIPS 186-3 does use an iterative trial-and-error approach, but its specific solution was quite different from the petition's proposed DSS-Schneier-Rose or DSS-Schneier-Menezes combinations. Specifically, the FIPS 186-3 algorithm demonstrates that when a sufficiently secure random bit generator is employed to generate a random number of appropriate length (e.g., 160 bits), it was unnecessary to also process the output of the generator with a one-way function $G()$ (e.g., a hash function such as SHA), let alone perform both operations repeatedly, as proposed in the petition's combination. The FIPS 186-3 approach thus would have been preferable when a sufficiently secure random number generator is employed because it eliminates the need to perform an additional hash operation.

74. For each of these reasons, and as confirmed by the approaches adopted by subsequent releases of DSS, the petition's "obvious to try" theory was based upon flawed assumptions. The petition apparently assumed that there were only two solutions to the problem of obtaining random numbers within a desired range that is smaller than the range of the random number generator itself (petition at p. 44 and EX1002, ¶158), but that assumption is unfounded and not consistent with the reality facing a POSITA at the time.

X. LEGAL STANDARDS

75. I understand that Petitioners contend that certain claims of the '961 patent are unpatentable because they are alleged to be rendered obvious in view of combinations of references. In preparing my analysis, I have applied the legal standards described below, which were provided to me by counsel for the Patent Owner.

76. It is my understanding that assessing the validity of a U.S. patent based on a prior art analysis requires two steps. First, one must construe the terms of the patent claims to understand what meaning one of ordinary skill in the art would have given the terms. Second, after the claim terms have been construed, one may then assess validity by comparing a patent claim to the "prior art." I understand that the teaching of the prior art is viewed through the eyes of a person of ordinary skill in the art at the time the invention was made. My analysis as to what constitutes a relevant person of ordinary skill in the art is set forth above. For purposes of my independent analysis at this stage, I have assumed that the references cited by the Petitioners in this proceeding are "prior art" even though some may not qualify as such under the law.

A. Obviousness

77. I understand that Petitioners contend that the challenged claims are invalid as "obvious." My understanding is that a patent claim is invalid as obvious

only if the subject matter “as a whole” of the claimed invention would have been obvious to a person of ordinary skill in the field at the time the invention was made. In determining the differences between a prior art reference (or a proposed combination of prior art references) and the claims, the question of obviousness is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious. Also, obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with rational underpinning to support the legal conclusion of obviousness.

78. I understand that a patent claim composed of several elements is not proved obvious merely by demonstrating that each of its elements was independently known in the prior art. In evaluating whether any claim of the '961 patent would have been obvious, I considered whether the Petition or any evidence submitted in this proceeding presented an articulated reason with a rational basis that would have prompted a person of ordinary skill in the field to combine the elements or concepts from the prior art in the same way as in the claimed invention.

79. I understand that there is no single way to define the line between true inventiveness on one hand (which is patentable) and the application of common sense and ordinary skill to solve a problem on the other hand (which is not

patentable). For example, market forces or other design incentives may be what produced a change, rather than true inventiveness.

80. It is my understanding that the decision-maker may consider whether the change was merely the predictable result of using prior art elements according to their known functions, or whether it was the result of true inventiveness. And, the decision-maker may also consider whether there is some teaching or suggestion in the prior art to make the modification or combination of elements recited in the claim at issue. Also, the decision-maker may consider whether the innovation applies a known technique that had been used to improve a similar device or method in a similar way. The decision-maker may also consider whether the claimed invention would have been obvious to try, meaning that the claimed innovation was one of a relatively small number of possible approaches to the problem with a reasonable expectation of success by those skilled in the art.

81. I have been instructed by counsel that if any of these considerations are relied upon to reach a conclusion of obviousness, the law requires that the analysis of such a relied-upon consideration must be made explicit. I understand that the decision-maker must be careful not to determine obviousness using the benefit of hindsight and that many true inventions might seem obvious after the fact. I understand that the decision-maker should consider obviousness from the position of a person of ordinary skill in the relevant field at the time the claimed

invention was made and that the decision-maker should not consider what is known today or what is learned from the teaching of the patent.

82. I understand that in order to determine whether a patent claim is obvious, one must make certain factual findings regarding the claimed invention and the prior art. Specifically, I understand that the following factors must be evaluated to determine whether a claim is obvious: (1) the scope and content of the prior art; (2) the difference or differences, if any, between the claim of the patent and the prior art; (3) the level of ordinary skill in the art at the time the claimed invention was made; and (4) the objective indicia of non-obviousness (if available), also known as “secondary considerations.”

83. I understand that the secondary considerations include:

- commercial success of a product due to the merits of the claimed invention;
- a long felt need for the solution provided by the claimed invention;
- unsuccessful attempts by others to find the solution provided by the claimed invention;
- copying of the claimed invention by others;
- unexpected and superior results from the claimed invention;
- acceptance by others of the claimed invention as shown by praise from others in the field or from the licensing of the claimed invention;
- teaching away from the conventional wisdom in the art at the time of the invention;

- independent invention of the claimed invention by others before or at about the same time as the named inventor thought of it; and
- other evidence tending to show obviousness.


84. It is my understanding that, in order to establish a secondary consideration, the evidence must demonstrate a nexus between that secondary consideration and the claimed invention.

XII. ADDITIONAL REMARKS

85. 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.

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

Dated: Aug 9 '19

By: 

Markus Jakobsson, Ph.D.
Portola Valley, California

APPENDIX A

CV and Research Statement

Markus Jakobsson

www.linkedin.com/in/markusjakobsson

www.markus-jakobsson.com

1 At a Glance

- **Focus.** *Identification of security problems, trends and solution along four axes – computational, structural, physical and social; quantitative and qualitative fraud analysis; development of disruptive security technologies.*
- **Education.** *PhD* (Computer Science/Cryptography, University of California at San Diego, 1997); *MSc* (Computer Engineering, Lund Institute of Technology, Sweden, 1994).
- **Large research labs.** *San Diego Supercomputer Center* (Researcher, 1996-1997); *Bell Labs* (Member of Technical Staff, 1997-2001); *RSA Labs* (Principal Research Scientist, 2001-2004); *Xerox PARC* (Principal Scientist, 2008-2010); PayPal (Principal Scientist of Consumer Security, Director, 2010-2013); Qualcomm (Senior Director, 2013-2015); Agari (Chief Scientist, 2016-2018); Amber Solutions Inc (Chief of Security and Data Analytics, 2018 – current)
- **Academia.** *New York University* (Adjunct Associate Professor, 2002-2004); *Indiana University* (Associate Professor & Associate Director, 2004-2008; Adjunct Associate Professor, 2008-2016).
- **Entrepreneurial activity.** *ZapFraud* (Anti-scam technology; CTO and founder, 2012-); *RavenWhite Security* (Authentication solutions; CTO and founder, 2005-); *RightQuestion* (Consulting; Founder, 2007-); *FatSkunk* (Malware detection; CTO and founder, 2009-2013 – FatSkunk was acquired by Qualcomm); *LifeLock* (Id theft protection; Member of fraud advisory board, 2009-2013); *CellFony* (Mobile security; Member of technical advisory board, 2009-2013); *PopGiro* (User Reputation; Member of technical advisory board, 2012-2013); *MobiSocial* (Social networking, Member of technical advisory board, 2013); *Stealth Security* (Anti-fraud, Member of technical advisory board, 2013–current)
- **Anti-fraud consulting.** *KommuneData* [Danish govt. entity] (1996); *J.P. Morgan Chase* (2006-2007); *PayPal* (2007-2011); *Boku* (2009-2010); *Western Union* (2009-2010).

- **Intellectual Property, Testifying Expert Witness.** *Inventor of 100+ patents; expert witness in several patent litigation cases* (McDermott, Will & Emery; Bereskin & Parr; WilmerHale; Hunton & Williams; Quinn Emanuel Urquhart & Sullivan; Freed & Weiss; Berry & Domer; Fish & Richardson; DLA Piper; Cipher Law Group; Kecker & Van Nest). Details and references upon request.
- **Publications.** Books: *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft* (Wiley, 2006); *Crimeware: Understanding New Attacks and Defenses* (Symantec Press, 2008); *The Death of the Internet* (Wiley, 2012); *Towards Trustworthy Elections: New Directions in Electronic Voting* (Springer Verlag, 2010); *Understanding Social Engineering* (Springer Verlag, 2016); *100+ peer-reviewed publications*

2 Summary

I am one of the more prominent computer scientists studying fraud and fraud prevention. I have performed and published novel research on fraud and authentication since 1993, with a focus on the payments industry since 1995. In 1999, I posited that what later became known as phishing would become a big problem. As a Principal Scientist at RSA Laboratories in 2001, my mandate was to determine the impact of future fraud scenarios on commerce and authentication, and developing intellectual property to address such problems. In 2004, I built a research group around online fraud and countermeasures, resulting in more than 50 publications and two books (“Phishing and Countermeasures”, Wiley; “Crimeware”, Symantec Press.) I co-founded the first company to address consumer security education, and am a pioneer in that area. I also co-founded an RSA Security spinoff (RavenWhite Security), and a company to address mobile malware (FatSkunk), and have overseen their intellectual property creation. FatSkunk was acquired by Qualcomm in 2013. I also founded ZapFraud, a company addressing Business Email Compromise. I am currently the Chief Scientist at Agari, a company addressing email-based fraud.

I have recruited and supervised junior colleagues, developers and PhD/Masters students for fifteen years. I have been in charge with building research groups at Bell Laboratories, RSA Laboratories and Indiana University. I was the most senior security researcher at Indiana University, and was hired to Xerox PARC to provide thought leadership to their security group. My former advisees have prominent roles at RSA Laboratories, Mozilla, Google, and top universities such as MIT and ETH Zurich MIT. I played a prominent role in defining the intellectual property efforts at PayPal/eBay, and contributed significantly to their portfolio. I founded and built FatSkunk, bringing a new security paradigm to the marketplace.

3 Recent Focus

My work primarily involves identifying trends in fraud and computing before they affect the market, and to develop and test countermeasures – whether technical, or based on user interaction or education. I am the inventor of more than 100 patents. At PayPal, I developed and tested a technology that allows the automatic creation of PINs from passwords [46], with direct applications to improved mobile security and simplified user experience. I also studied liar buyer fraud [39] and developed improved authentication and fraud detection methods. At FatSkunk, I developed a new Anti-Virus paradigm (see, e.g., [42]); protected the intellectual property; built a team to build the technology; and worked towards commercializing the technology. After the acquisition of FatSkunk, this work was continued at Qualcomm, where I also worked on IoT, wearable authentication methods [41], anti-theft technology and privacy technology aimed at automatically detecting and block attempts to track users. My work at ZapFraud focused on understanding and blocking email scams [40], with a focus on business email compromise, and building a foundational patent portfolio. My work at Agari addressed enterprise-facing scams such as Business Email Compromise, Ransomware, and other abuse based on social engineering and identity deception. My work at Amber Solutions involve protocol design for defending against attacks on consumer and enterprise sensor networks.

My PhD is in theoretical computer science, but my later emphasis has been on applied security, including authentication, click-fraud [29], mobile malware detection [42], detection of business email compromise, and the development of metrics to detect new types of fraud.

4 Publication List

Books (1-6); book chapters, journals, conference publications and other scientific publications (7-147), issued /published U.S. patents (148-234). For an updated list, and for international patents, please see www.markus-jakobsson.com/publications and appropriate patent search engines.

References

- [1] M. Jakobsson, *Mobile Authentication: Problems and Solutions*, ISBN 1461448778, 125 pages, Springer, 2013.
- [2] M. Jakobsson, (editor) *The Death of the Internet*, ASIN B009CN2JVE, 359 pages, IEEE Computer Society Press, 2012.
- [3] D. Chaum, M. Jakobsson, R. L. Rivest, P. Y. Ryan, J. Benaloh, and M. Kutyłowski, (editors), *Towards Trustworthy Elections: New Directions in Electronic Voting*, 411 pages, (Vol. 6000), Springer, 2010.

- [4] M. Jakobsson and Z. Ramzan (editors), *Crimeware: Trends in Attacks and Countermeasures*, ISBN 0321501950, Hardcover, 582 pages, Symantec Press / Addison Wesley, 2008.
- [5] M. Jakobsson and S. A. Myers (editors), *Phishing and Countermeasures: Understanding the Increasing Problem of Electronic Identity Theft*, ISBN 0-471-78245-9, Hardcover, 739 pages, Wiley, 2006.
- [6] M. Jakobsson, M. Yung, J. Zhou, *Applied Cryptography and Network Security: Second International Conference, Yellow Mountain, China, 2004*, 511 pages, Lecture Notes in Computer Science (Book 3089), 2004.
- [7] N. Sae-Bae, M. Jakobsson, *Hand Authentication on Multi-Touch Tablets*, HotMobile 2014
- [8] Y. Park, J. Jones, D. McCoy, E. Shi, M. Jakobsson, *Scambaiter: Understanding Targeted Nigerian Scams on Craigslist*, NDSS 2014
- [9] D. Balfanz, R. Chow, O. Eisen, M. Jakobsson, S. Kirsch, S. Matsumoto, J. Molina, and P. van Oorschot, "The future of authentication," *Security & Privacy*, IEEE, 10(1), 22-27, 2012.
- [10] M. Jakobsson, and H. Siadati, *Improved Visual Preference Authentication: Socio-Technical Aspects in Security and Trust*, (STAST), 2012 Workshop on IEEE, 27-34, 2012.
- [11] M. Jakobsson, R. I. Chow, and J. Molina, "Authentication-Are We Doing Well Enough?[Guest Editors' Introduction]" *Security & Privacy*, IEEE, 10(1), 19-21, 2012.
- [12] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, "Implicit authentication through learning user behavior," *Information Security*, 99-113, Springer Berlin Heidelberg, 2011.
- [13] M. Jakobsson and K. Johansson, "Practical and Secure Software-Based Attestation," *Lightweight Security & Privacy: Devices, Protocols and Applications (LightSec)*, 1-9, 2011.
- [14] A. Juels, D. Catalano, and M. Jakobsson, *Coercion-resistant electronic elections: Towards Trustworthy Elections*, 37-63, Springer Berlin Heidelberg, 2010.
- [15] M. Jakobsson and F. Menczer, "Web Forms and Untraceable DDoS Attacks," in *Network Security*, Huang, S., MacCallum, D., and Du, D. Z., Eds., 77-95, Springer, 2010.
- [16] R. Chow, M. Jakobsson, R. Masuoka, J. Molina, Y. Niu, E. Shi, and Z. Song, "Authentication in the Clouds: A Framework and its Application to Mobile Users," 2010.

- [17] X. Wang, P. Golle, M. Jakobsson, and A. Tsow, "Deterring voluntary trace disclosure in re-encryption mix-networks," *ACM Trans. Inf. Syst. Secur.*, 13(2), 1-24, 2010.
- [18] X. Wang, P. Golle, M. Jakobsson, A. Tsow, "Deterring voluntary trace disclosure in re-encryption mix-networks," *ACM Trans. Inf. Syst. Secur.* 13(2): (2010)
- [19] M. Jakobsson, and C. Soghoian, "Social Engineering in Phishing," *Information Assurance, Security and Privacy Services*, 4, 2009.
- [20] M. Jakobsson, C. Soghoian and S. Stamm, "Phishing," *Handbook of Financial Cryptography* (CRC press, 2008)
- [21] M. Jakobsson and A. Tsow, "Identity Theft," In John R. Vacca, Editor, "Computer And Information Security Handbook" (Morgan Kaufmann, 2008)
- [22] S. Srikwan and M. Jakobsson, "Using Cartoons to Teach Internet Security," *Cryptologia*, vol. 32, no. 2, 2008
- [23] M. Jakobsson, N. Johnson and P. Finn, "Why and How to Perform Fraud Experiments," *IEEE Security and Privacy*, March/April 2008 (Vol. 6, No. 2) pp. 66-68
- [24] M. Jakobsson and S. Myers, "Delayed Password Disclosure," *International Journal of Applied Cryptography*, 2008, pp. 47-59.
- [25] M. Jakobsson and S. Stamm, "Web Camouflage: Protecting Your Clients from Browser Sniffing Attacks," *IEEE Security & Privacy Magazine*. November/December 2007
- [26] P. Finn and M. Jakobsson, "Designing and Conducting Phishing Experiments," *IEEE Technology and Society Magazine*, Special Issue on Usability and Security
- [27] T. Jagatic, N. Johnson, M. Jakobsson and F. Menczer. "Social Phishing," *The Communications of the ACM*, October 2007
- [28] A. Tsow, M. Jakobsson, L. Yang and S. Wetzel, "Warkitting: the Drive-by Subversion of Wireless Home Routers," *Anti-Phishing and Online Fraud, Part II Journal of Digital Forensic Practice*, Volume 1, Special Issue 3, November 2006
- [29] M. Gandhi, M. Jakobsson and J. Ratkiewicz, "Badvertisements: Stealthy Click-Fraud with Unwitting Accessories," *Anti-Phishing and Online Fraud, Part I Journal of Digital Forensic Practice*, Volume 1, Special Issue 2, November 2006

- [30] N. Ben Salem, J.-P. Hubaux and M. Jakobsson. "Reputation-based Wi-Fi Deployment," *Mobile Computing and Communications Review*, Volume 9, Number 3 (Best papers of WMASH 2004)
- [31] N. Ben Salem, J. P. Hubaux, and M. Jakobsson. "Node Cooperation in Hybrid Ad hoc Networks," *IEEE Transactions on Mobile Computing*, Vol. 5, No. 4, April 2006.
- [32] P. MacKenzie, T. Shrimpton, and M. Jakobsson. "Threshold Password-Authenticated Key Exchange," *Journal of Cryptology*, 2005
- [33] A. Juels, M. Jakobsson, E. Shriver, and B. Hillyer. "How To Turn Loaded Dice Into Fair Coins." *IEEE Transactions on Information Theory*, vol. 46(3). May 2000. pp. 911–921.
- [34] M. Jakobsson, P. MacKenzie, and J.P. Stern. "Secure and Lightweight Advertising on the Web," *Journal of Computer Networks*, vol. 31, issue 11–16, Elsevier North-Holland, Inc., 1999. pp. 1101–1109.
- [35] M. Jakobsson, "Cryptographic Protocols," Chapter from *The Handbook of Information Security*. Hossein Bidgoli, Editor-in-Chief. Copyright John Wiley & Sons, Inc., 2005, Hoboken, N.J.
- [36] M. Jakobsson, "Cryptographic Privacy Protection Techniques," Chapter from *The Handbook of Information Security*. Hossein Bidgoli, Editor-in-Chief. Copyright John Wiley & Sons, Inc., 2005, Hoboken, N.J.
- [37] M. Jakobsson, E. Shi, P. Golle, R. Chow, "Implicit authentication for mobile devices," 4th USENIX Workshop on Hot Topics in Security (HotSec '09); 2009 August 11; Montreal, Canada.
- [38] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, J. Molina, "Controlling data in the cloud: outsourcing computation without outsourcing control," *Proceedings of the 2009 ACM Workshop on Cloud Computing Security (CCSW 2009)*; 2009 November 13; Chicago, IL. NY: ACM; 2009; pp. 85–90.
- [39] M. Jakobsson, H. Siadati, M. Dhiman, "Liar Buyer Fraud, and How to Curb It," *NDSS*, 2015
- [40] M. Jakobsson, T.-F. Yen, "How Vulnerable Are We To Scams?," *BlackHat*, 2015
- [41] M. Jakobsson, "How to Wear Your Password," *BlackHat*, 2014
- [42] M. Jakobsson and G. Stewart, "Mobile Malware: Why the Traditional AV Paradigm is Doomed, and How to Use Physics to Detect Undesirable Routines," in *BlackHat*, 2013.

- [43] M. Jakobsson, and H. Siadati, "SpoofKiller: You Can Teach People How to Pay, but Not How to Pay Attention" in Socio-Technical Aspects in Security and Trust (STAST), 2012 Workshop on, 3-10, 2012.
- [44] M. Jakobsson, and M. Dhiman, "The benefits of understanding passwords," in Proceedings of the 7th USENIX conference on Hot Topics in Security, Berkeley, CA, USA, 2012.
- [45] M. Jakobsson, and S. Taveau, "The Case for Replacing Passwords with Biometrics," Mobile Security Technologies, 2012.
- [46] M. Jakobsson and D. Liu, "Bootstrapping mobile PINs using passwords," W2SP, 2011.
- [47] M. Jakobsson and R. Akavipat, "Rethinking passwords to adapt to constrained keyboards," 2011.
- [48] Y. Niu, E. Shi, R. Chow, P. Golle, and M. Jakobsson, "One Experience Collecting Sensitive Mobile Data," In USER Workshop of SOUPS, 2010.
- [49] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, "Implicit Authentication through Learning User Behavior," 2010.
- [50] M. Jakobsson and K. Johansson, Assured Detection of Malware With Applications to Mobile Platforms, 2010.
- [51] M. Jakobsson and K. Johansson, "Retroactive Detection of Malware With Applications to Mobile Platforms," in HotSec 2010, Washington, DC, 2010.
- [52] M. Jakobsson, A Central Nervous System for Automatically Detecting Malware, 2009.
- [53] R. Chow, P. Golle, M. Jakobsson, R. Masuoka, J. Molina, E. Shi, and J. Staddon, "Controlling data in the cloud: outsourcing computation without outsourcing control," ACM workshop on Cloud computing security (CCSW), 2009.
- [54] M. Jakobsson and A. Juels, "Server-Side Detection of Malware Infection," in New Security Paradigms Workshop (NSPW), Oxford, UK, 2009.
- [55] M. Jakobsson, "Captcha-free throttling," Proceedings of the 2nd ACM workshop on Security and artificial intelligence, 15-22, 2009.
- [56] M. Jakobsson, E. Shi, P. Golle, and R. Chow, "Implicit authentication for mobile devices," Proceedings of the 4th USENIX conference on Hot topics in security, 9-9, 2009.
- [57] C. Soghoian, O. Friedrichs and M. Jakobsson, "The Threat of Political Phishing," International Symposium on Human Aspects of Information Security & Assurance (HAISA 2008)

- [58] R. Chow, P. Golle, M. Jakobsson, L. Wang and X. Wang, "Making CAPTCHAs Clickable," In proc. of HotMobile 2008.
- [59] M. Jakobsson, A. Juels, and J. Ratkiewicz, "Privacy-Preserving History Mining for Web Browsers," Web 2.0 Security and Privacy, 2008.
- [60] M. Jakobsson, E. Stolterman, S. Wetzel, L. Yang, "Love and Authentication," (Notes) ACM Computer/Human Interaction Conference (CHI), 2008. Also see www.I-forgot-my-password.com
- [61] M. Jakobsson and S. Myers, "Delayed Password Disclosure," Proceedings of the 2007 ACM workshop on Digital Identity Management
- [62] M. Jakobsson, S. Stamm, Z. Ramzan, "JavaScript Breaks Free," W2SP '07
- [63] A. Juels, S. Stamm, M. Jakobsson, "Combatting Click Fraud via Premium Clicks," USENIX Security 2007
- [64] R. Chow, P. Golle, M. Jakobsson, X. Wang, "Clickable CAPTCHAs," Ad-Fraud '07 Workshop; 2007 September 14; Stanford, CA, USA
- [65] S. Stamm, Z. Ramzan, and M. Jakobsson, "Drive-by Pharming," In Proceedings of Information and Communications Security, 9th International Conference, ICICS 2007
- [66] M. Jakobsson, A. Tsow, A. Shah, E. Blevis, Y.-K. Lim, "What Instills Trust? A Qualitative Study of Phishing," USEC '07.
- [67] R. Akavipat, V. Anandpara, A. Dingman, C. Liu, D. Liu, K. Pongsanon, H. Roinestad and M. Jakobsson, "Phishing IQ Tests Measure Fear, not Ability," USEC '07.
- [68] M. Jakobsson, "The Human Factor in Phishing," American Conference Institute's Forum on Privacy & Security of Consumer Information, 2007
- [69] S. Srikwan, M. Jakobsson, A. Albrecht and M. Dalkilic, "Trust Establishment in Data Sharing: An Incentive Model for Biodiversity Information Systems," TrustCol 2006
- [70] J.Y. Choi, P. Golle, M. Jakobsson, "Tamper-Evident Digital Signatures: Protecting Certification Authorities Against Malware," DACS '06
- [71] L. Yang, M. Jakobsson, S. Wetzel, "Discount Anonymous On Demand Routing for Mobile Ad hoc Networks," SECURECOMM '06
- [72] P. Golle, X. Wang, M. Jakobsson, A. Tsow, "Deterring Voluntary Trace Disclosure in Re-encryption Mix Networks." IEEE S&P '06
- [73] M. Jakobsson, A. Juels, T. Jagatic, "Cache Cookies for Browser Authentication (Extended Abstract)," IEEE S&P '06

- [74] M. Jakobsson and J. Ratkiewicz, “Designing Ethical Phishing Experiments: A study of (ROT13) rOnl auction query features.”, WWW ’06
- [75] M. Jakobsson and S. Stamm. “Invasive Browser Sniffing and Countermeasures,” WWW ’06
- [76] J.Y. Choi, P. Golle and M. Jakobsson. “Auditable Privacy: On Tamper-Evident Mix Networks,” Financial Crypto ’06
- [77] A. Juels, D. Catalano and M. Jakobsson. “Coercion-Resistant Electronic Elections,” WPES ’05
- [78] V. Griffith and M. Jakobsson. “Messin’ with Texas, Deriving Mother’s Maiden Names Using Public Records,” ACNS ’05, 2005.
- [79] M. Jakobsson and L. Yang. “Quantifying Security in Hybrid Cellular Networks,” ACNS ’05, 2005
- [80] Y.-C. Hu, M. Jakobsson, and A. Perrig. “Efficient Constructions for One-way Hash Chains,” ACNS ’05, 2005
- [81] M. Jakobsson. “Modeling and Preventing Phishing Attacks,” Phishing Panel in Financial Cryptography ’05. 2005, abstract in proceedings.
- [82] N. Ben Salem, J.-P. Hubaux, and M. Jakobsson. “Reputation-based Wi-Fi Deployment Protocols and Security Analysis,” In WMASH ’04. ACM Press, 2004. pp. 29–40.
- [83] M. Jakobsson and S. Wetzel. “Efficient Attribute Authentication with Applications to Ad Hoc Networks,” In VANET ’04. ACM Press, 2004. pp. 38–46.
- [84] M. Jakobsson, X. Wang, and S. Wetzel. “Stealth Attacks in Vehicular Technologies,” Invited paper. In Proceedings of IEEE Vehicular Technology Conference 2004 Fall (VTC-Fall 2004). IEEE, 2004.
- [85] A. Ambainis, H. Lipmaa, and M. Jakobsson. “Cryptographic Randomized Response Technique,” In PKC ’04. LNCS 2947. Springer-Verlag, 2004. pp. 425–438.
- [86] P. Golle, M. Jakobsson, A. Juels, and P. Syverson. “Universal Re-encryption for Mixnets,” In CT-RSA ’04. LNCS 2964. Springer-Verlag, 2004. pp. 163–178.
- [87] P. Golle and M. Jakobsson. “Reusable Anonymous Return Channels,” In WPES ’03. ACM Press, 2003. pp. 94–100.
- [88] M. Jakobsson, S. Wetzel, B. Yener. “Stealth Attacks on Ad-Hoc Wireless Networks,” In IEEE VTC ’03, 2003.

- [89] N. Ben Salem, L. Buttyan, J.-P. Hubaux, and M. Jakobsson. “A Charging and Rewarding Scheme for Packet Forwarding in Multi-hop Cellular Networks,” In ACM MobiHoc ’03. ACM Press, 2003. pp. 13–24.
- [90] M. Jakobsson, J.-P. Hubaux and L. Buttyan. “A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks,” In FC ’03. LNCS 2742. Springer-Verlag, 2003. pp. 15–33.
- [91] M. Jakobsson, T. Leighton, S. Micali and M. Szydło. “Fractal Merkle Tree Representation and Traversal,” In RSA-CT ’03 2003.
- [92] A. Boldyreva and M. Jakobsson. “Theft protected proprietary certificates,” In DRM ’02. LNCS 2696, 2002. pp. 208–220.
- [93] P. Golle, S. Zhong, M. Jakobsson, A. Juels, and D. Boneh. “Optimistic Mixing for Exit-Polls,” In Asiacrypt ’02. LNCS 2501. Springer-Verlag, 2002. pp. 451–465.
- [94] P. MacKenzie, T. Shrimpton, and M. Jakobsson. “Threshold Password-Authenticated Key Exchange,” In CRYPTO ’02. LNCS 2442. Springer-Verlag, 2002. pp. 385–400.
- [95] M. Jakobsson. “Fractal Hash Sequence Representation and Traversal,” In Proceedings of the 2002 IEEE International Symposium on Information Theory (ISIT ’02). 2002. pp. 437–444.
- [96] M. Jakobsson, A. Juels, and R. Rivest. “Making Mix Nets Robust For Electronic Voting By Randomized Partial Checking,” In Proceedings of the 11th USENIX Security Symposium. USENIX Association, 2002. pp. 339–353.
- [97] D. Coppersmith and M. Jakobsson. “Almost Optimal Hash Sequence Traversal,” In Financial Crypto ’02. 2002.
- [98] M. Jakobsson. “Financial Instruments in Recommendation Mechanisms,” In Financial Crypto ’02. 2002.
- [99] J. Garay, and M. Jakobsson. “Timed Release of Standard Digital Signatures,” In Financial Crypto ’02. 2002.
- [100] F. Menczer, N. Street, N. Vishwakarma, A. Monge, and M. Jakobsson. “Intellishopper: A Proactive, Personal, Private Shopping Assistant,” In AAMAS ’02. ACM Press, 2002. pp. 1001–1008.
- [101] M. Jakobsson, A. Juels, and P. Nguyen. “Proprietary Certificates,” In CT-RSA ’02. LNCS 2271. Springer-Verlag, 2002. pp. 164–181.
- [102] M. Jakobsson and A. Juels. “An Optimally Robust Hybrid Mix Network,” In PODC ’01. ACM Press. 2001. pp. 284–292.

- [103] M. Jakobsson and M. Reiter. “Discouraging Software Piracy Using Software Aging,” In DRM '01. LNCS 2320. Springer-Verlag, 2002. pp. 1–12.
- [104] M. Jakobsson and S. Wetzel. “Security Weaknesses in Bluetooth,” In CT–RSA '01. LNCS 2020. Springer-Verlag, 2001. pp. 176–191.
- [105] M. Jakobsson and D. Pointcheval. “Mutual Authentication for Low-Power Mobile Devices,” In Financial Crypto '01. LNCS 2339. Springer-Verlag, 2001. pp. 178–195.
- [106] M. Jakobsson, D. Pointcheval, and A. Young. “Secure Mobile Gambling,” In CT–RSA '01. LNCS 2020. Springer-Verlag, 2001. pp. 110–125.
- [107] M. Jakobsson and S. Wetzel. “Secure Server-Aided Signature Generation,” In PKC '01. LNCS 1992. Springer-Verlag, 2001. pp. 383–401.
- [108] M. Jakobsson and A. Juels. “Addition of ElGamal Plaintexts,” In T. Okamoto, ed., ASIACRYPT '00. LNCS 1976. Springer-Verlag, 2000. pp. 346–358.
- [109] M. Jakobsson, and A. Juels. “Mix and Match: Secure Function Evaluation via Ciphertexts,” In ASIACRYPT '00. LNCS 1976. Springer-Verlag, 2000. pp. 162–177.
- [110] R. Arlein, B. Jai, M. Jakobsson, F. Monrose, and M. Reiter. “Privacy-Preserving Global Customization,” In ACM E-Commerce '00. ACM Press, 2000. pp. 176–184.
- [111] C.-P. Schnorr and M. Jakobsson. “Security of Signed ElGamal Encryption,” In ASIACRYPT '00. LNCS 1976. Springer-Verlag, 2000. pp. 73–89.
- [112] P. Bohannon, M. Jakobsson, and S. Srikwan. “Cryptographic Approaches to Privacy in Forensic DNA Databases,” In Public Key Cryptography '00. LNCS 1751. Springer-Verlag, 2000, pp. 373–390.
- [113] J. Garay, M. Jakobsson, and P. MacKenzie. “Abuse-free Optimistic Contract Signing,” In CRYPTO '99. LNCS 1666. Springer-Verlag, 1999. pp. 449–466.
- [114] M. Jakobsson. “Flash Mixing,” In PODC '99. ACM Press, 1999. pp. 83–89.
- [115] G. Di Crescenzo, N. Ferguson, R. Impagliazzo, and M. Jakobsson. “How To Forget a Secret,” In STACS '99. LNCS 1563. Springer-Verlag, 1999. pp. 500–509.
- [116] M. Jakobsson, D. M'Raihi, Y. Tsiounis, and M. Yung. “Electronic Payments: Where Do We Go from Here?,” In CQRE (Secure) '99. LNCS 1740. Springer-Verlag, 1999. pp. 43–63.

- [117] C.P. Schnorr and M. Jakobsson. “Security Of Discrete Log Cryptosystems in the Random Oracle + Generic Model,” In Conference on The Mathematics of Public-Key Cryptography. 1999.
- [118] M. Jakobsson and A. Juels “Proofs of Work and Breadpudding Protocols,” In CMS ’99. IFIP Conference Proceedings, Vol. 152. Kluwer, B.V., 1999. pp. 252 – 272.
- [119] M. Jakobsson and C-P Schnorr. “Efficient Oblivious Proofs of Correct Exponentiation,” In CMS ’99. IFIP Conference Proceedings, Vol. 152. Kluwer, B.V., 1999. pp. 71–86.
- [120] M. Jakobsson, P. MacKenzie, and J.P. Stern. “Secure and Lightweight Advertising on the Web,” In World Wide Web ’99
- [121] M. Jakobsson, J.P. Stern, and M. Yung. “Scramble All, Encrypt Small,” In Fast Software Encryption ’99. LNCS 1636. Springer-Verlag, 1999. pp. 95–111.
- [122] M. Jakobsson and J. Mueller. “Improved Magic Ink Signatures Using Hints,” In Financial Cryptography ’99. LNCS 1648. Springer-Verlag, 1999. pp. 253–268.
- [123] M. Jakobsson. “Mini-Cash: A Minimalistic Approach to E-Commerce,” In Public Key Cryptography ’99. LNCS 1560. Springer-Verlag, 1999. pp. 122–135.
- [124] M. Jakobsson. “On Quorum Controlled Asymmetric Proxy Re-encryption,” In Public Key Cryptography ’99. LNCS 1560. Springer-Verlag, 1999. pp. 112–121.
- [125] M. Jakobsson and A. Juels. “X-Cash: Executable Digital Cash,” In Financial Cryptography ’98. LNCS 1465. Springer-Verlag, 1998. pp. 16–27.
- [126] M. Jakobsson and D. M’Raihi. “Mix-based Electronic Payments,” In Proceedings of the Selected Areas in Cryptography. LNCS 1556. Springer-Verlag, 1998. pp. 157–173.
- [127] M. Jakobsson, E. Shriver, B. Hillyer, and A. Juels. “A Practical Secure Physical Random Bit Generator,” In CCS ’98: Proceedings of the 5th ACM conference on Computer and communications security. ACM Press, 1998. pp. 103–111.
- [128] M. Jakobsson. “A Practical Mix,” In Advances in Cryptology – EuroCrypt ’98. LNCS 1403. Springer-Verlag, 1998. pp. 448–461.
- [129] M. Jakobsson and M. Yung. “On Assurance Structures for WWW Commerce,” In Financial Cryptography ’98. LNCS 1465. Springer-Verlag, 1998. pp. 141–157.

- [130] E. Gabber, M. Jakobsson, Y. Matias, and A. Mayer. “Curbing Junk E-Mail via Secure Classification,” In *Financial Cryptography '98*. LNCS 1465. Springer-Verlag, 1998. pp. 198–213.
- [131] M. Jakobsson and M. Yung. “Distributed ‘Magic Ink’ Signatures,” In *Advances in Cryptology – EuroCrypt '97*. LNCS 1233. Springer-Verlag, 1997. pp. 450–464.
- [132] M. Jakobsson and M. Yung. “Applying Anti-Trust Policies to Increase Trust in a Versatile E-Money System,” In *Financial Cryptography '97*. LNCS 1318. Springer-Verlag, 1997. pp. 217–238.
- [133] A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, and M. Yung. “Proactive public-key and signature schemes,” In *Proceedings of the 4th Annual Conference on Computer Communications Security*. ACM Press, 1997. pp. 100–110.
- [134] M. Bellare, M. Jakobsson, and M. Yung. “Round-Optimal Zero-Knowledge Arguments Based on any One-Way Function,” In *Advances in Cryptology – EuroCrypt '97*. LNCS 1233. Springer-Verlag, 1997. pp. 280–305.
- [135] M. Jakobsson and M. Yung. “Proving Without Knowing,” In *Crypto '96*. LNCS 1109. Springer-Verlag, 1996. pp. 186–200.
- [136] M. Jakobsson, K. Sako, and R. Impagliazzo. “Designated Verifier Proofs and Their Applications,” In *Advances in Cryptology – EuroCrypt '96*. LNCS 1070. Springer-Verlag, 1996. pp. 143–154.
- [137] M. Jakobsson and M. Yung. “Revokable and Versatile Electronic Money,” In *CCS '96: Proceedings of the 3rd ACM conference on Computer and communications security*. ACM Press, 1996. pp. 76–87.
- [138] M. Jakobsson. “Ripping Coins for a Fair Exchange,” In *Advances in Cryptology – EuroCrypt '95*. LNCS 921. Springer-Verlag, 1995. pp. 220–230.
- [139] M. Jakobsson. “Blackmailing using Undeniable Signatures,” In *Advances in Cryptology EuroCrypt '94*. LNCS 950. Springer-Verlag, 1994. pp. 425–427.
- [140] M. Jakobsson. “Reducing costs in identification protocols,” *Crypto '92*, 1992.
- [141] M. Jakobsson. “Machine-Generated Music with Themes,” In *International Conference on Artificial Neural Networks '92*. Vol 2. Amsterdam: Elsevier, 1992. pp. 1645–1646
- [142] M. Jakobsson, “Social Engineering 2.0: What’s Next,” *McAfee Security Journal*, Fall 2008

- [143] M. Jakobsson and S. Myers, "Delayed Password Disclosure," ACM SIGACT News archive, Volume 38, Issue 3 (September 2007), pp. 56 - 75
- [144] V. Griffith and M. Jakobsson. "Messin' with Texas, Deriving Mother's Maiden Names Using Public Records," CryptoBytes, 2007.
- [145] M. Jakobsson, A. Juels, J. Ratkiewicz, "Remote-Harm Detection," Beta-version available at rhd.ravenwhitedevelopment.com/
- [146] S. Stamm, M. Jakobsson, "Social Malware," Experimental results available at [www.indiana.edu/ phishing/verybigad/](http://www.indiana.edu/phishing/verybigad/)
- [147] M. Jakobsson. "Privacy vs. Authenticity," Ph.D. Thesis, University of California at San Diego, 1997
- [148] Markus Jakobsson, Automatic PIN creation using passwords, US20130125214 A1, 2012.
- [149] Markus Jakobsson, Systems and methods for creating a user credential and authentication using the created user credential, US 20130111571 A1, 2012.
- [150] Markus Jakobsson, Password check by decomposing password, US 20120284783 A1, 2012.
- [151] Markus Jakobsson, William Leddy, System and methods for protecting users from malicious content, US 20120192277 A1, 2011.
- [152] Markus Jakobsson, Karl-Anders R. Johansson, Auditing a device, US8370935 B1, 2011.
- [153] Markus Jakobsson, Methods and Apparatus for Efficient Computation of One-Way Chains in Cryptographic Applications, US20120303969 A1, 2011.
- [154] Markus Jakobsson, Automatic PIN creation using password, US 20120110634 A1, 2011.
- [155] Markus Jakobsson, Jim Roy Palmer, Gustavo Maldonado, Interactive CAPTCHA, US20130007875 A1, 2011.
- [156] Markus Jakobsson, System access determination based on classification of stimuli, US 20110314559 A1, 2011.
- [157] Markus Jakobsson, System access determination based on classification of stimuli, WO 2011159356 A1, 2011.
- [158] Markus Jakobsson, Method, medium, and system for reducing fraud by increasing guilt during a purchase transaction, US8458041 B1, 2011.
- [159] Markus Jakobsson, Visualization of Access Information, US 20120233314 A1, 2011.

- [160] Markus Jakobsson, Richard Chow,Runting Shi, Implicit authentication, US20120137340 A1, 2010.
- [161] Markus Jakobsson, Karl-Anders R. Johansson, Auditing a device, EP2467793 A1, 2010.
- [162] Markus Jakobsson, Event log authentication using secure components, US 20110314297 A1, 2010.
- [163] Markus Jakobsson, Philippe J.P. Golle, Risk-based alerts, US 20110314426 A1, 2010.
- [164] Markus Jakobsson, Karl-Anders R. Johansson, Auditing a device, US 20110041178 A1, 2010.
- [165] M. Jakobsson, A. Juels, J. Kaliski Jr, S. Burton and others, Identity authentication system and method, US Patent 7,502,933, 2009.
- [166] Markus Jakobsson, Method and system for facilitating throttling of interpolation-based authentication, US8219810 B2, 2009.
- [167] Markus Jakobsson, Karl-Anders R. Johansson, Auditing a device, US8375442 B2, 2009.
- [168] Markus Jakobsson, Karl-Anders R. Johansson, Auditing a device, US 20110041180 A1, 2009.
- [169] Markus Jakobsson, Pattern-based application classification, US 20110055925 A1, 2009.
- [170] Markus Jakobsson, Method and apparatus for detecting cyber threats, US8286225 B2, 2009.
- [171] Markus Jakobsson, Method and apparatus for detecting cyber threats, US20110035784 A1, 2009.
- [172] Markus Jakobsson, CAPTCHA-free throttling, US8312073 B2, 2009.
- [173] Markus Jakobsson, Captcha-free throttling, US 20110035505 A1, 2009.
- [174] Markus Jakobsson, Christopher Soghoian, Method and apparatus for throttling access using small payments, US 20100153275 A1, 2008.
- [175] Markus Jakobsson, Christopher Soghoian, Method and apparatus for mutual authentication using small payments, US 20100153274 A1, 2008.
- [176] Philippe J.P. Golle, Markus Jakobsson,Richard Chow, Resetting a forgotten password using the password itself as authentication, US 20100125906 A1, 2008.
- [177] Philippe J. P. Golle, Markus Jakobsson,Richard Chow, Authenticating users with memorable personal questions, US8161534 B2, 2008.

- [178] Richard Chow, Philippe J.P. Golle, Markus Jakobsson, Jessica N. Stad-don, Authentication based on user behavior, US20100122329 A1, 2008.
- [179] Richard Chow, Philippe J.P. Golle, Markus Jakobsson, Jessica N. Stad-don, Enterprise password reset, US 20100122340 A1, 2008.
- [180] Markus Jakobsson, Methods and apparatus for efficient computation of one-way chains in cryptographic applications, US8086866 B2, 2008.
- [181] Richard Chow, Philippe J. P. Golle, Markus Jakobsson, Selectable captchas, US8307407 B2, 2008.
- [182] Markus Jakobsson, Ari Juels, Sidney Louis Stamm, Method and apparatus for combatting click fraud, US 20080162227 A1, 2007.
- [183] Jakobsson, Method and apparatus for evaluating actions performed on a client device, US 20080037791 A1, 2007.
- [184] Jakobsson, Ari Juels, Method and apparatus for storing information in a browser storage area of a client device, US 20070106748 A1, 2006.
- [185] Markus Jakobsson, Steven Andrew Myers, Anti-phising logon authentica-tion object oriented system and method, WO 2006062838 A1, 2005.
- [186] Jakobsson, Jean-Pierre Hubaux,Levente Buttyan, Micro-payment scheme encouraging collaboration in multi-hop cellular networks, US 20050165696 A1, 2004.
- [187] Andrew Nanopoulos, Karl Ackerman, Piers Bowness, William Duane, Markus Jakobsson, Burt Kaliski, Dmitri Pal, Shane D. Rice,Less , Sys-tem and method providing disconnected authentication, WO2005029746 A3, 2004.
- [188] Andrew Nanopoulos, Karl Ackerman, Piers Bowness, William Duane, Markus Jakobsson, Burt Kaliski, Dmitri Pal, Shane Rice, Ronald Rivest, Less , System and method providing disconnected authentication, US 20050166263 A1, 2004.
- [189] Andrew Nanopoulos, Karl Ackerman, Piers Bowness, William Duane, Markus Jakobsson, Burt Kaliski, Dmitri Pal, Shane D. Rice,Less , Sys-teme et procede d'authentification deconnectee, WO 2005029746 A2, 2004.
- [190] Markus Jakobsson, Ari Juels, Burton S. Kaliski Jr., Identity authentica-tion system and method, WO2004051585 A3, 2003.
- [191] Markus Jakobsson, Ari Juels, Burton S. Kaliski, Jr., Identity authentica-tion system and method, US 7502933 B2, 2003.
- [192] Markus Jakobsson, Ari Juels, Burton S. Kaliski Jr., Systeme et procede de validation d'identite, WO 2004051585 A2, 2003.

- [193] Markus Jakobsson, Burton S. Kaliski, Jr., Method and apparatus for graph-based partition of cryptographic functionality, US7730518 B2, 2003.
- [194] Markus Jakobsson, Phong Q. Nguyen, Methods and apparatus for private certificates in public key cryptography, US7404078 B2, 2002.
- [195] Jakobsson, Philip MacKenzie, Thomas Shrimpton, Method and apparatus for performing multi-server threshold password-authenticated key exchange, US20030221102 A1, 2002.
- [196] Markus Jakobsson, Philip D MacKenzie, Method and apparatus for distributing shares of a password for use in multi-server password authentication, US 7073068 B2, 2002.
- [197] Markus Jakobsson, Methods and apparatus for efficient computation of one-way chains in cryptographic applications, EP 1389376 A1 (text from WO2002084944A1), 2002.
- [198] Markus Jakobsson, Adam Lucas Young, Method and apparatus for identification tagging documents in a computer system, US 7356845 B2, 2002.
- [199] Juan A. Garay, Markus Jakobsson, Methods and apparatus for computationally-efficient generation of secure digital signatures, US 7366911 B2, 2001.
- [200] Markus Jakobsson, Methods and apparatus for efficient computation of one-way chains in cryptographic applications, US 7404080 B2, 2001.
- [201] Markus Jakobsson, Susanne Gudrun Wetzel, Securing the identity of a bluetooth master device (bd addr) against eavesdropping by preventing the association of a detected channel access code (cac) with the identity of a particular bluetooth device, WO2002019641 A3, 2001.
- [202] Markus Jakobsson, Susanne Gudrun Wetzel, Procédé et appareil permettant d'assurer la sécurité d'utilisateurs de dispositifs à capacités bluetooth, WO 2002019641 A2, 2001.
- [203] Robert M. Arlein, Ben Jai, Markus Jakobsson, Fabian Monroe, Michael Kendrick Reiter, Less , Methods and apparatus for providing privacy-preserving global customization, US 7107269 B2, 2001.
- [204] Markus Jakobsson, Susanne Gudrun Wetzel, Secure distributed computation in cryptographic applications, US6950937 B2, 2001.
- [205] Markus Jakobsson, Susanne Gudrun Wetzel, Method and apparatus for ensuring security of users of short range wireless enable devices, US6981157 B2, 2001.
- [206] Markus Jakobsson, Susanne Gudrun Wetzel, Method and apparatus for ensuring security of users of bluetooth TM-enabled devices, US 6574455 B2, 2001.

- [207] Garay; Juan A. (West New York, NJ), Jakobsson; M. (Hoboken, NJ), Kristol; David M. (Summit, NJ), Mizikovsky; Semyon B.(Morganville, NJ), Cryptographic key processing and storage , 7023998, 2001.
- [208] Markus Jakobsson, Method, apparatus, and article of manufacture for generating secure recommendations from market-based financial instrument prices, US 6970839 B2, 2001.
- [209] Markus Jakobsson, Encryption method and apparatus with escrow guarantees, US7035403 B2, 2001.
- [210] Jakobsson, Call originator access control through user-specified pricing mechanism in a communication network, US20020099670 A1, 2001.
- [211] Markus Jakobsson, Claus Peter Schnorr, Tagged private information retrieval, US7013295 B2, 2000.
- [212] Markus Jakobsson, Secure enclosure for key exchange, US 7065655 B1, 2000.
- [213] Markus Jakobsson, Michael Kendrick Reiter, Software aging method and apparatus for discouraging software piracy, US 7003110 B1, 2000.
- [214] Markus Jakobsson, Probabilistic theft deterrence, US6501380 B1, 2000.
- [215] Markus Jakobsson, Michael Kendrick Reiter, Abraham Silberschatz, Anonymous and secure electronic commerce, EP 1150227 A1, 2000.
- [216] Markus Jakobsson, Ari Juels, Proofs of work and bread pudding protocols, US 7356696 B1, 2000.
- [217] Markus Jakobsson, Joy Colette Mueller, Methods of protecting against spam electronic mail, US7644274 B1, 2000.
- [218] Philip L. Bohannon, Markus Jakobsson, Fabian Monroe, Michael Kendrick Reiter, Susanne Gudrun Wetzels, Less , Generation of repeatable cryptographic key based on varying parameters, EP1043862 B1, 2000.
- [219] Markus Jakobsson, Ari Juels, Mix and match: a new approach to secure multiparty computation, US6772339 B1, 2000.
- [220] Markus Jakobsson, Ari Juels, Mixing in small batches, US6813354 B1, 2000.
- [221] Philip L. Bohannon, Markus Jakobsson, Fabian Monroe, Michael Kendrick Reiter, Susanne Gudrun Wetzels, Less , Generation of repeatable cryptographic key based on varying parameters, US 6901145 B1, 36566
- [222] Markus Jakobsson, Claus Peter Schnorr, Non malleable encryption method and apparatus using key-encryption keys and digital signature, US6931126 B1, 2000.

- [223] Markus Jakobsson, Claus Peter Schnorr, Non malleable encryption method and apparatus using key-encryption keys and digital signature, US 6931126 B1, 2000.
- [224] Markus Jakobsson, Flash mixing apparatus and method, US 6598163 B1, 1999.
- [225] Markus Jakobsson, Minimalistic electronic commerce system, US 6529884 B1, 1999.
- [226] Markus Jakobsson, Method and system for providing translation certificates, US 6687822 B1, 1999.
- [227] Markus Jakobsson, Verification of correct exponentiation or other operations in cryptographic applications, US6978372 B1, 1999.
- [228] Markus Jakobsson, Non malleable encryption apparatus and method, US 6507656 B1, 1999.
- [229] Markus Jakobsson, Method and system for quorum controlled asymmetric proxy encryption, US 6587946 B1, 1998.
- [230] Markus Jakobsson, Practical mix-based election scheme, US 6317833 B1, 1998.
- [231] Markus Jakobsson, Ari Juels, Method and apparatus for extracting unbiased random bits from a potentially biased source of randomness, US 6393447 B1, 1998.
- [232] Markus Jakobsson, Ari Juels, Executable digital cash for electronic commerce, US6157920 A, 1998.
- [233] Bruce Kenneth Hillyer, Markus Jakobsson, Elizabeth Shriver, Storage device random bit generator, US 6317499 B1, 1998.
- [234] Markus Jakobsson, Method and apparatus for encrypting, decrypting, and providing privacy for data values, US 6049613 A, 1998.