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 IPR2019-00923
Patent 7,372,961

PATENT OWNER'S PRELIMINARY RESPONSE

TABLE OF CONTENTS

EXHIBITS

EX2001 Declaration of Markus Jakobsson, Ph.D.

EX2002 FIPS PUB 186-2, Digital Signature Standard (DSS) Change Notice (October 5, 2001)

EX2003 FIPS PUB 186-3, Digital Signature Standard (DSS) (June 2009)

EX2004 Corrected Final Ruling on Claim Construction in *BlackBerry Limited v. Facebook, Inc. et al* (Case No. CV 18-1844-GW) (C.D.Cal.) and *BlackBerry Limited v. Snap Inc.* (Case No. CV 18-2693-GW) (C.D.Cal.)

EX2005 Serge Vaudenay, *Evaluation Report on DSA* (2005) ("Vaudenay")

I. INTRODUCTION

Petitioners have failed to meet their burden of showing a reasonable likelihood that they would prevail with respect to any of the challenged claims of U.S. Patent No. 7,372,961 ("the '961 patent"). For the reasons detailed below, the petition suffers from both factual and legal errors, each of which provides an independent basis to deny institution. Collectively, the various deficiencies in the petition, along with the mature posture of concurrent litigation over the '961 patent that will reach a final resolution long before any final decision in this forum, provides overwhelming weight to deny institution of this flawed IPR petition.

As a threshold issue, the Board should exercise its discretion under §314(a) to deny institution. If instituted here, a PTAB trial would likely conclude in November 2020—more than a half year after the jury trial will have already resolved Petitioner's allegations of invalidity. In this case, a subsequent PTAB trial would unnecessarily multiply the proceedings, wasting the resources of both the parties and the Board.

Grounds 1-4 of the petition are also flawed on the merits. Grounds 1 and 3 cite Rose as allegedly providing motivation to modify the Digital Signature Standard ("DSS") to address the known security vulnerability associated with DSS's key generation algorithm. Yet, Rose itself fails to disclose multiple features from the solution claimed in the '961 patent, and the petition improperly attempts

to cure these deficiencies with modifications to Rose that are unsupported by any evidence submitted with the petition. Likewise, Grounds 2 and 4, which cite Menezes to address the security vulnerability in DSS, also fails to disclose several aspects of the '961 claims for which it is relied upon. In these Grounds, the petition again resorts to unsupported contentions and hindsight-based reasons to follow the roadmap gleaned from the '961 claims. Worse, in arguing that the claimed solution would have been "obvious to try," the petition ignores the critical fact that in three subsequent releases of DSS, none addressed the security vulnerability in Petitioner's hindsight matter contrived from Rose or Menezes. The evidence here confirms the petition's assertion (that only one or two solutions existed) was baseless.

For at least these reasons, Patent Owner respectfully requests that the Board deny institution of Grounds 1-4 of the petition, and decline to institute *inter partes* review of the '961 patent.

II. SUMMARY OF THE '961 PATENT

The '961 patent discloses a novel solution for generating a cryptographic key *k* in an unbiased manner so as to improve the security of a cryptographic function where the key *k* is applied. EX1001, 2:65-3:4, 3:64-4:17, FIG. 2; EX2001, ¶¶17-20; *see also id.*, ¶¶15-16.

As the background of the '961 patent explains, 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 uses 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.

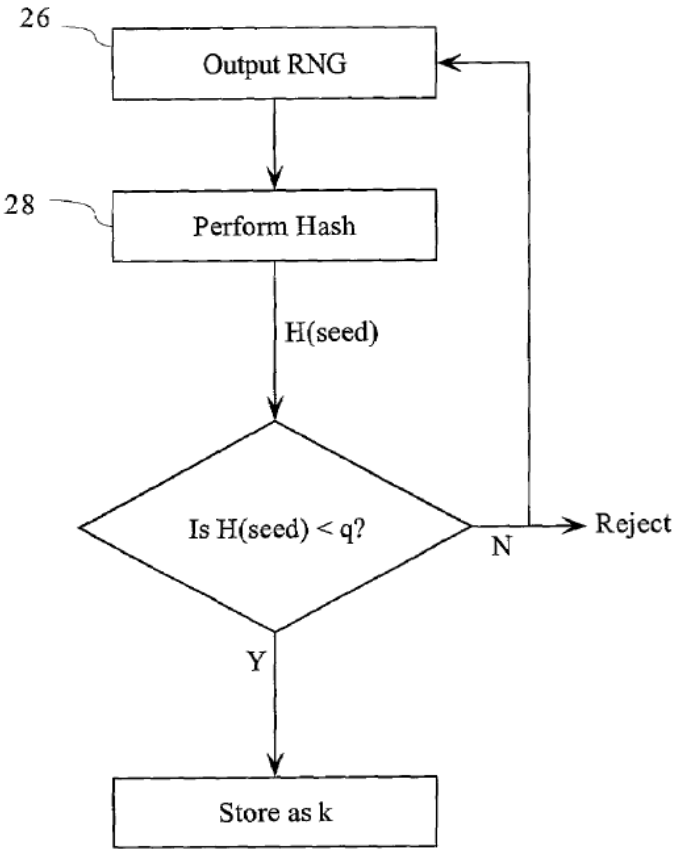
Ideally, the key *k* is chosen uniformly at random from a pre-defined range of values (possibly subject to known security requirements). However, certain implementations of the DSA—including the Digital Signature Standard (DSS) that was promulgated by the National Institute of Standards and Technology (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.* According to the '961 patent, 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, researcher 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, thereby enabling the hacker to sign messages as if he or she were the authorized user.. *Id.*, 2:56-61.

Despite 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). On this date, the inventors of the '961 patent filed a patent application disclosing their solution for addressing the security vulnerability identified by Bleichenbacher. Simply put, the reality in December 2000 was far different from Petitioner's current hindsight-based theory of what a person of ordinary skill in the art ("POSITA") would have been prompted to do.

The inventors' techniques 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* is to be employed (e.g., *q* in DSS). *Id.*, 3:64-4:17, FIG. 2. Specifically, the length *L* of the output of the hash function in terms of a number of binary bits may be at least as long as the length of the parameter *q* against which the hash output is compared. As such, the output of the hash function will 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). EX2001, ¶20. 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 Bleichenbacher was able to leverage to compromise the long-term private key of a user. *See* EX1004, p. 18; *infra*, Section III.A.



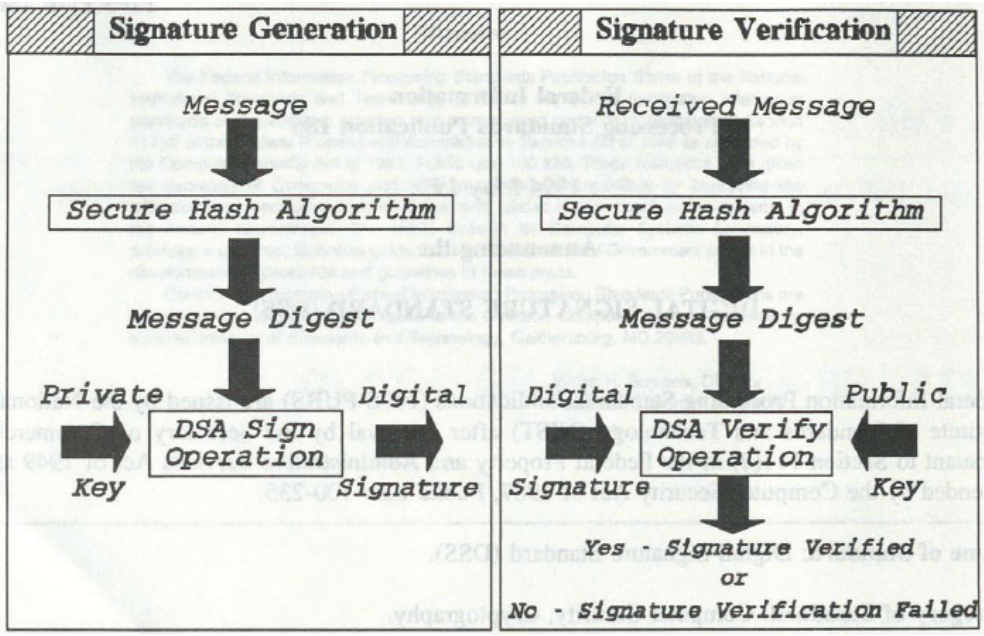
EX1001, FIG. 2.

III. SUMMARY OF THE CITED REFERENCES

A. DSS (EX1004)

The DSS reference submitted in this proceeding is a copy of the FIPS 186-1 Digital Signature Standard document.[1] *See* EX1004; EX2001, ¶¶24-28. As explained in the reference, DSS "specifies a Digital Signature Algorithm (DSA) appropriate for applications requiring a digital, rather than written, signature." *Id.*, 5. "The digital signature is computed using a set of rules (i.e., the DSA) and a set of parameters such that the identity of the signatory and integrity of the data can be verified." *Id.* Figure 1 shows an overview of digital signature generation and verification in DSS:

[1] Petitioners contend that the version of DSS submitted with the petition (i.e., FIPS 186-1 (1994) (EX1004)) prescribes the same key generation method as the version of the standard (i.e., FIPS 186-2) referenced in the background of the '961 patent. Pet., 20.



EX1004, FIG. 1.

The signature generation algorithm specified in DSS uses both a long-term private key *x* of the signer and a short-term (ephemeral) key *k* to generate a digital signature for a message. *Id.*, 10-11, 17-18. Of particular relevance to the instant proceeding, DSS describes an algorithm for pre-computing 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 = EFCDAB89 98BADCFE 10325476 C3D2E1F0 67452301.

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

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

a. k = G(t,KKEY) mod q.

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

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

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

EX1004, 18[2]; *see also* 17 (describing that the function G(t,KKEY) is a one-way function such as the Secure Hash Algorithm (SHA) or Data Encryption Standard (DES)).

This key generation algorithm from DSS is referenced in the background section of the ’961 patent, which notes that (according to Step 3(a) above), “DSS requires the reduction of the integer mod *q*, i.e., *k*=SHA(seed) mod *q*” and this

[2] For purposes of clarity, the image of text as it appears in the publication is shown; the words are included in total word count below as if they were presented as a block quotation of text.

operation introduces bias into the selection of key *k*. EX1001, 2:41-61.

Notably, DSS's algorithms allow for computation of a batch of keys *k* *without* randomly selecting a new seed-key (KKEY) each time a new key *k* is to be computed. EX1004, 18; EX2001, ¶28. The seed-key (KKEY) is independently selected (e.g., using a random number generator) only once—namely, when the algorithm commences to allow for computation of the initial key *k* in the batch. *See* EX1004, 18 ("Step 1. Choose a secret initial value for the seed-key, KKEY."); EX1001, 2:40-42 ("A seed value, SV, is generated from a random number generator which is then hashed by a SHA−1 hash function …"). After the algorithm computes the initial key *k* in the batch, subsequent keys *k* are computed simply by updating the value of the seed-key (KKEY) according to the formula set forth in Step 3(d): KKEY = (1 + KKEY + k) mod 2^b. EX1004, 14. DSS thus acknowledges that it is unnecessary to randomly re-select KKEY for each iteration, and expressly discloses a preference for applying a deterministic function to update KKEY based on its prior value rather than performing an independent, randomized re-selection of KKEY. EX2001, ¶28.

B. Rose (EX1006)

The Rose reference is an alleged newsgroup posting that describes a method for "roll[ing] a random number in the interval [0, n)." EX1006, 2; EX2001, ¶¶29-30. In particular, Rose's suggested method is explicitly defined by the following

computer code:

```
/*
 * 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[3].

As shown in the computer code, Rose's method determines a random number within a specified range [0, n-1] by first computing the largest multiple of n that is less than or equal to BIG (i.e., "the biggest value returned by random ()") and assigns this value to the variable 'biggest.' EX1006, 2. The method then enters 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' (i.e., until the output of the random() function is less than the largest computed multiple of n). *Id.* Because the final output of the random() function

[3] *Supra*, footnote 2.

(assigned to the variable '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 method ends by computing 'rval' modulo n and returns the result of this operation as the output of the software routine. *Id.*

While Rose purports to "[a]void [a] slight bias to low numbers" for the basic task of random number generation, Rose never explains how its techniques could be applied to a key generation algorithm, never seeks to avoid bias in a value other than the output of a random number generator (e.g., never seeks to avoid bias in a *hash* of the output of the random number generator within a desired range that is smaller than the range of the hash function itself), and never repeats the pair of operations required by the '961 claims when the result of a particular iteration is rejected. EX2001, ¶30. Rose thus provides a meaningfully different solution to a different problem than that addressed by the '961 patent. *Id.*

C. Menezes (EX1005)

Menezes is textbook on the subject of "applied cryptography" that spans hundreds of pages, but the petition relies almost exclusively on a single paragraph of disclosure regarding random number generation:

> **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[4]; EX2001, ¶¶31-33.

Unlike Rose, Menezes does not explicitly offer source code for the algorithm disclosed in the cited paragraph. Nonetheless, Menezes's algorithm shares with Rose many of the same points of departure from the features recited in the Challenged Claims of the '961 patent. EX2001, ¶33. For example, as explained in further detail below (Section VII), Menezes's algorithm determines whether to accept or reject an integer of a bit sequence output by a random bit generator rather than operating on the *hash* of a randomly generated number. These different contexts are material because Menezes assumes the availability of a perfect random bit generator that could generate each bit in a sequence without any bias. EX2001, ¶33. 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 fundamentally different solution for

[4] *Supra*, footnote 2.

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. EX2001, ¶33.

These, and other deficiencies, highlight the flaws in Petitioner's hindsight-based proposals that fall short of the innovation described in the '961 patent.

D. Schneier (EX1008)

Schneier is another textbook on the subject of applied cryptography covering a wide range of topics. *See* EX1008; EX2001, ¶34. 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).

E. Rao (EX1018) / Floyd (EX1019)

The petition relies on Rao and Floyd for their alleged descriptions of "arithmetic processor[s]" and "arithmetic/logic unit[s] (ALU[s])," respectively. Pet., 60-62. The arguments presented in this Preliminary Response are not specifically directed to the citations of Rao and Floyd, but Patent Owner reserves its right to raise additional arguments in this regard at appropriate times in this proceeding in the future.

IV. CLAIM CONSTRUCTION

Patent Owner submits that all claim terms should be construed according to the *Phillips* standard. *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005); 37 C.F.R. §42.100; EX2001, ¶¶15-16 (level of ordinary skill for a "POSITA"). Unlike the broadest reasonable interpretation standard, the *Phillips* standard seeks "the correct construction—the construction that most accurately delineates the scope of the claim[ed] invention." *PPC Broadband, Inc. v. Corning Optical Comm'ns RF, LLC,* 815 F.3d 734, 740-42 (Fed. Cir. 2016). Under the *Phillips* standard, the purpose of claim construction is "to understand and explain, but not to change, the scope of the claims." *Embrex, Inc. v. Serv. Eng'g Corp.*, 216 F.3d 1343, 1347 (Fed. Cir. 2000).

On April 5, 2019, the district court issued a claim construction order in the concurrent litigation, and a copy of the order is provided here. *See* EX2004. Regarding the claim phrase "**reducing mod q**," the district court concluded that this phrase means "computing the remainder of dividing a value by q." EX2004, 31-35. The district court cited to column 2:32-55 of the '961 patent as being consistent with this construction. *Id.*, 34. Here, Grounds 1-4 are fatally defective for the reasons detailed below regardless of whether the district court's formal constructions for the terms of the '961 patent are adopted. Thus, Patent Owner recognizes the Board may determine that no formal claim constructions are

necessary at institution because "claim terms need only be construed to the extent necessary to resolve the controversy." *Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011); EX2001, ¶¶21-23.

V. STANDARD FOR GRANTING *INTER PARTES* REVIEW

The Board may grant a petition for *inter partes* review only where "the information presented in the petition … shows that there is a reasonable likelihood that the Petitioners would prevail with respect to at least 1 of the claims challenged in the petition." 35 U.S.C. §314(a); 37 C.F.R. §42.108(c). Petitioners bear the burden of showing that this statutory threshold has been met. *See* Office Patent Trial Practice Guide, 77 Fed. Reg. 48,756, 48,756 (Aug. 14, 2012). Critically, Petitioners must fulfill this burden based on "information presented in the petition" (35 U.S.C. §314(a)), and the law forbids Petitioners from subsequently adding theories/arguments that should have been part of their initial petition. *Intelligent Bio-Systems, Inc. v. Illumina Cambridge, Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016) (citing to 35 U.S.C. § 312) ("It is of the utmost importance that petitioners in the IPR proceedings adhere to the requirement that the initial petition identify 'with particularity' the 'evidence that supports the grounds for the challenge to each claim.'"); *see also Cuozzo Speed Techs., LLC v. Lee*, 136 S.Ct. 2131, 2154 (2016) (Alito, J. concurring in part and dissenting in part) ("Thus, if a petition fails to state its challenge with particularity—or if the Patent Office institutes review on

claims or grounds not raised in the petition—the patent owner is forced to shoot into the dark. The potential for unfairness is obvious.").

As such, the original petition must establish a *prima facie* case of obviousness with regard to its proposed combinations of references. It is well settled that "rejections on obviousness cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." *KSR Intn'l Co. v. Teleflex Inc.*, 550 U.S. 398, 418 (2007) (quoting *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)); *PersonalWeb Techs., LLC v. Apple, Inc.*, 848 F.3d 987, 993 (Fed. Cir. 2017) (explaining that it is insufficient to allege "that a skilled artisan, once presented with the two references, would have understood that they could be combined."). As shown below, Petitioners have not established a *prima facie* case of obviousness with respect to Grounds 1-4, and the Board should decline to institute review of the '961 patent.

VI. GROUND 1 BASED ON DSS IN VIEW OF SCHNEIER AND ROSE IS DEFICIENT

A. The Petition's Proposed Modification Of DSS In View Of Rose Is Flawed And Improperly Rooted In Hindsight

The combination cited in Ground 1 starts with the digital signature algorithm disclosed in DSS and alleges that it would have been obvious for a POSITA to

modify DSS based on Rose to arrive at the inventions claimed in the '961 patent.[5] But as explained in detail below, Petitioners' contentions here are false and unsupported by the evidence. EX2001, ¶¶39-54. DSS merely discloses the same key generation technique that was already referenced in the background of the '961 patent, which suffers from a bias that renders its signature scheme subject to compromise. EX1004, 17-18; EX1001, 1:52-2:61. Rose then discloses an algorithm for generating a random number within a specified range, but the algorithm differs in multiple respects from the features that are actually recited in the Challenged Claims. EX1006. Despite these differences, the petition proceeds

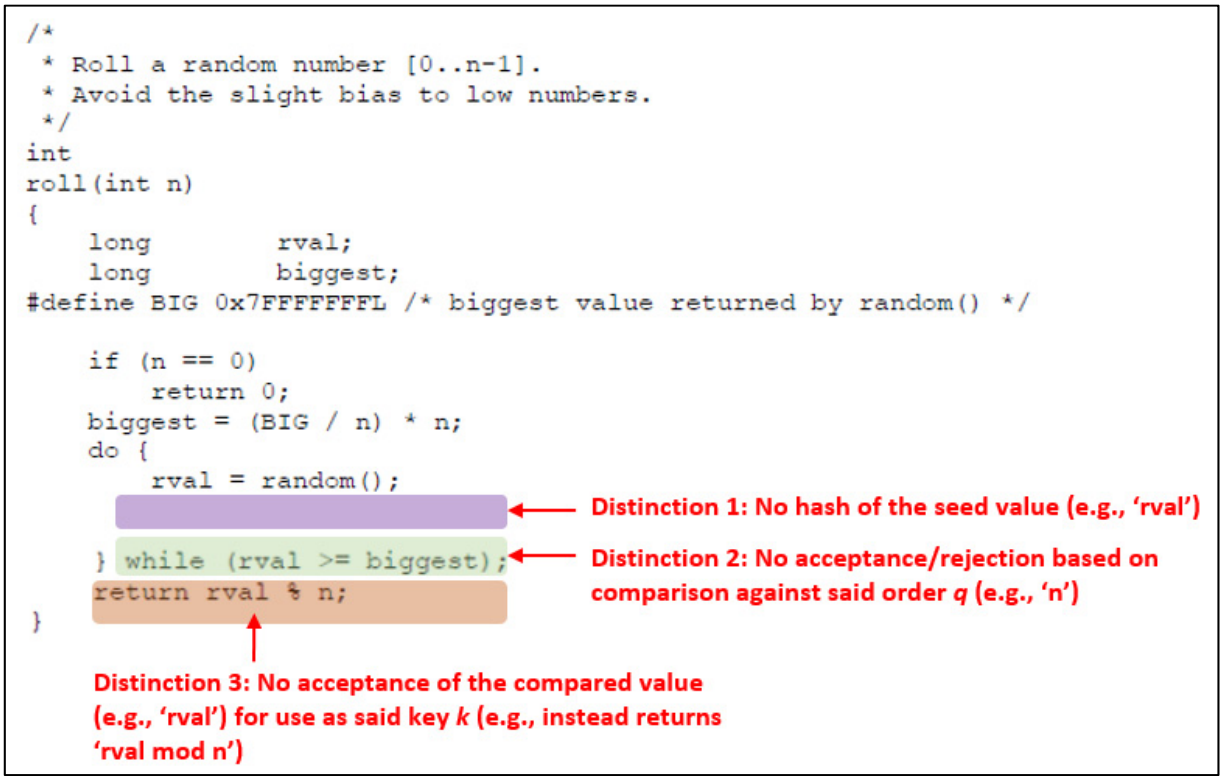
[5] The petition relies on Schneier for its teaching of a true random number generator to the extent that such a generator is not expressly disclosed in DSS. Pet., 42. However, this brief focuses on the flaws in Rose (Grounds 1 and 3) and Menezes (Grounds 2 and 4), as well as the flaws in the petition's modification of DSS in view of Rose or Menezes, respectively. For simplicity, Patent Owner at times refers to the DSS-Rose or DSS-Menezes combinations and modifications to DSS in view of Rose or Menezes without explicit reference to Schneier. Nonetheless, it is recognized that Schneier is also part of these Grounds, and Patent Owner will address the deficiencies of Schneier later (if necessary) in this proceeding.

to argue that the proposed combination of DSS and Rose somehow would have provided *all* the features of independent claim 1. But Petitioners offer no evidence or explanation to justify material changes that were required to be made to Rose's algorithm to allegedly map the DSS-Rose combination to claim 1. Lacking such evidence or other corroboration, Petitioners' DSS-Rose combination in Ground 1 is improperly rooted in hindsight and is critically flawed.

1. Rose Fails To Disclose The Missing Claim Features From DSS

DSS discloses the same, insecure algorithm that introduces bias in the selection of key *k* as described in the background of the '961 patent. EX2001, ¶40. As such, the petition relies exclusively on Rose to argue that the inventors' claimed solution to the bias problem would have been obvious. Pet., 42 (FN 11). In these circumstances, it might be expected that Rose would disclose the limitations of claim 1 that are missing from DSS. But this is not the case. Rose instead describes a random number generator that never would have been acceptable for use in a cryptographic function such as DSS. EX2001, ¶¶41-42. Worse, Rose actually fails to disclose several of the limitations that are also missing from DSS. *Id.*

For context, recall that Rose discloses the following software code to illustrate his algorithm for rolling a random number in the range [0, n-1]:



EX1006, p. 2 (highlighting and annotations added)[6].

Even a cursory review of Rose's code reveals three critical differences from the techniques described and claimed in the '961 patent. First, Rose merely updates 'rval' in each iteration of the do-while loop by calling the random() function and assigning its output to 'rval'. In contrast, claim 1, which recites the following operations, requires more:

> generating a seed value SV from a random number generator;
>
> **performing a hash function H( ) on said seed value**

[6] *Supra*, footnote 2.

> **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, 5:37-48 (claim 1) (emphasis added).[7] Thus, claim 1 provides for the method to be repeated after each rejection, and that the repeated method includes "performing a hash function H( ) on said seed value SV to provide an output H(SV)." Yet, Rose's loop only evaluates the random() function to generate an output 'rval' (allegedly corresponding to the "seed value SV" of claim 1). Rose never performs the additional operation required by claim 1 of performing a hash function on the seed value (e.g., performing a hash function on 'rval') either in the loop or before the loop is entered. EX2001, ¶¶41-42 (explaining that the

[7] Independent claim 15 recites corresponding language to claim 1, and the petition's challenges against claim 15 are deficient for at least each of the same reasons that are explained herein with respect to claim 1.

"random()" function represented in Rose invoked a common random number generator based on a physical source of randomness rather than a hash or other one-way function).

A second difference stems from the condition that Rose employs to exit the do-while loop. Claim 1 recites that the output H(SV) is accepted if its value is less than "said order q" and rejected if its value is not less than "said order q." EX1001, 5:37-48. That is, acceptance or rejection of the output H(SV) turns on whether its value is less than the parameter q. The petition (pp. 37-39) alleges correspondence between 'n' (Rose) and 'q' ('961 claims), but even under Petitioners' mapping, Rose does not condition its acceptance or rejection of 'rval' in an analogous manner to the '961 patent. Specifically—as Petitioners' own declarant, Dr. Katz, admits—Rose compares 'rval' not to 'n' itself but instead to the value 'biggest', where 'biggest' is a multiple of 'n' (and potentially much greater than 'n' itself). Pet. 37 ("biggest is assigned to the largest multiple of 'n' that is less than or equal to BIG") (citing EX1002, ¶132).

Third, further Rose departs from the '961 claims as a result of the modulo operation performed on 'rval'. In particular, considering the output of Rose's roll() function as corresponding to the claimed "key *k*", Rose differs from claim 1 in that the output of roll() is not the value that was accepted to terminate repetition of the method (e.g., the do-while loop), but is instead a remainder of the value that was

accepted to terminate repetition of the method. Put another way, under Petitioners' mapping, Rose would need to return the value 'rval'—i.e., the value that is accepted to exit the do-while loop—as the output of the roll() function (corresponding to "key *k*") to align in an analogous way with claim 1. *See* EX1001, 5:49-51 (claim 1 reciting "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"). But Rose does not adopt this approach, instead reducing 'rval' by modulus 'n' before returning the reduced value as the output of the roll() function.

Each of these differences demonstrates that Rose provides a flawed starting point for addressing the security vulnerability in DSS. To prove obviousness, Petitioners were required to explain why each of these deficiencies in Rose alone would be resolved in the combination with DSS, but as explained in the following section, Petitioners failed to do so.

2. The Petition Uses The '961 Claims As A Roadmap To Rewrite Rose's Algorithm Without Justification

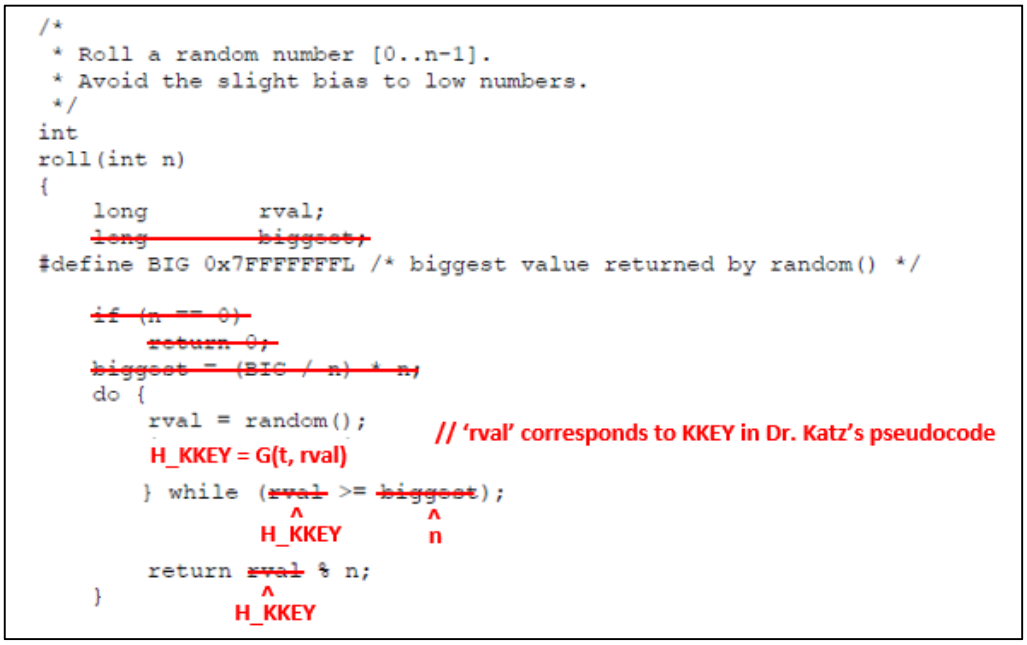
In spite of Rose's deficiencies as outlined above (Section VI.A.1), the petition pushes forward with its combination with DSS. Petitioners' declarant, Dr. Katz, represents the proposed combination through a modified version of Rose's computer code (modified with the luxury of hindsight):

```
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
}
```

EX1002, ¶152[8].

While Dr. Katz's pseudocode retains a familiar form to Rose (e.g., implementation of a do-while loop), the layers and degree of changes injected here (to follow the roadmap of the '961 patent claims) is striking—especially given that the original code was barely a dozen lines long to begin with. The following annotations on Rose's original computer code shows the specific revisions implemented in Dr. Katz's code:

[8] *Supra*, footnote 2.



EX1002, ¶152 (annotated)[9].

Among the layers of revisions adopted by Dr. Katz relative to Rose's initial proposal (revisions which the petition fully embraces) are:

(1) elimination of the computation of 'biggest' as a multiple of 'n';

(2) addition of an operation for computing H_KKEY within the do-while loop based on the output of the one-way function G() (e.g., a hash function such as SHA as provided in DSS);

(3) changing of the loop-exit condition from 'rval' (or 'KKEY' as labeled in Dr. Katz's code) being less than the 'biggest' multiple of 'n' to H_KKEY being less than 'n' itself; and

[9] *Supra*, footnote 2.

(4) replacing the value returned by the roll() function with 'H_KKEY modulo n' rather than 'rval modulo n.'

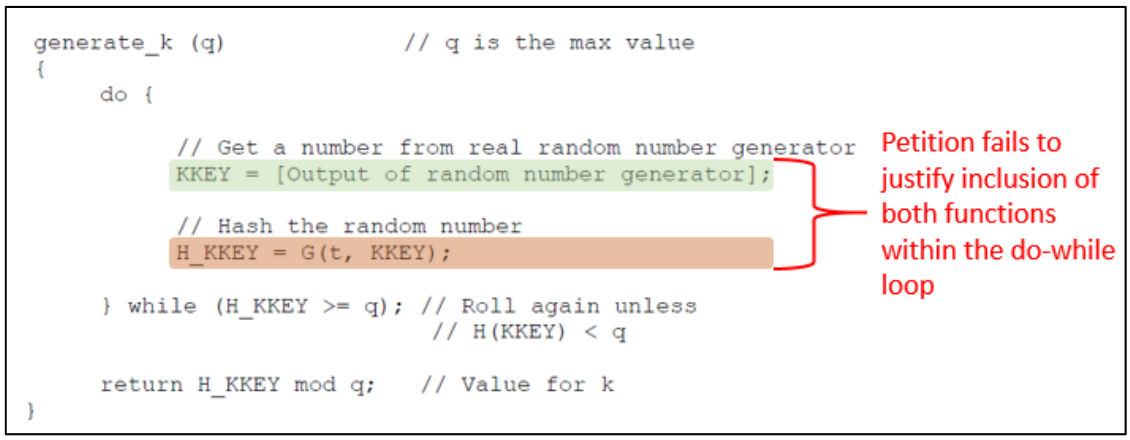
EX2001, ¶¶43-45.

While the petition and Dr. Katz attempt to justify *some* of the departures from Rose's original algorithm—such as the elimination of 'biggest'—as an allegedly natural consequence of applying Rose's teaching to DSS, the petition conveniently ignores at least two subtle, but significant modifications proposed by Dr. Katz. *See* Pet., 40 (describing why 'biggest' would allegedly be unnecessary in the DSS combination).

In particular, the petition first fails to justify its revision in the proposed DSS-Rose combination for conditioning loop termination on the output of the one-way function G() (i.e., 'H_KKEY') rather than on the output of the random() function (i.e., 'rval' or 'KKEY') as disclosed in Rose. EX2001, ¶¶35-38 (explaining recursive technique described by claim 1), ¶¶39-54 (describing deficiencies in cited DSS-Rose combination's recursive aspect). None of the cited references determine whether to accept/reject a *hashed* random number rather than the random number itself, and yet the petition assumes—without corroboration or a specific explanation from Dr. Katz—that a POSITA would have discarded Rose's stated preference and instead achieved a solution taught only in the '961 patent. For example, Rose only teaches a technique for accepting/rejecting the *output of a*

random number generator, not the output of a *hashed* random number. EX1006, 2. Even if inclusion of the one-way function G() (e.g., the SHA hash function) in the loop could have been explained in view of DSS, the petition presents no evidence that Rose's alleged suggestion (for mitigating bias through selection of a random number that is less than a threshold value (e.g., a threshold based on 'n')) would be transferable to also mitigate bias in the selection of an output of the one-way function G() after the random number has been transformed. Only with the benefit of hindsight, the petition instead injects this proposed change in Rose's code for purposes of following the roadmap drawn by the '961 patent claims. *Arkie Lures, Inc. v. Gene Larew Tackle, Inc.*, 119 F.3d 953, 956 (Fed. Cir. 1997) ("Good ideas may well appear 'obvious' after they have been disclosed."). The Board has consistently rejected such flawed obviousness theories, and it should do so here. *Cook Incorporated et al v. Medtronic Vascular, Inc.*, IPR2018-01571, Paper 7, 22 (PTAB March 4, 2019) ("[W]e are mindful not to read into [the secondary reference] that which is not disclosed. Such hindsight bias has no role in an obviousness analysis."); *Freebit AS v. Bose Corporation*, IPR2018-00509, Paper 7, 24 (PTAB June 24, 2018) ("Petitioner provides insufficient reasons why a person of ordinary skill in the art would have made the asserted modifications to the teachings of [the primary reference other than hindsight improperly gleaned from the challenged claims").

Moreover, a second flaw in the proposed DSS-Rose combination stems from the petition's failure to address why a POSITA allegedly would have opted to evaluate *both* the random() function *and* the one-way function within the do-while loop together:

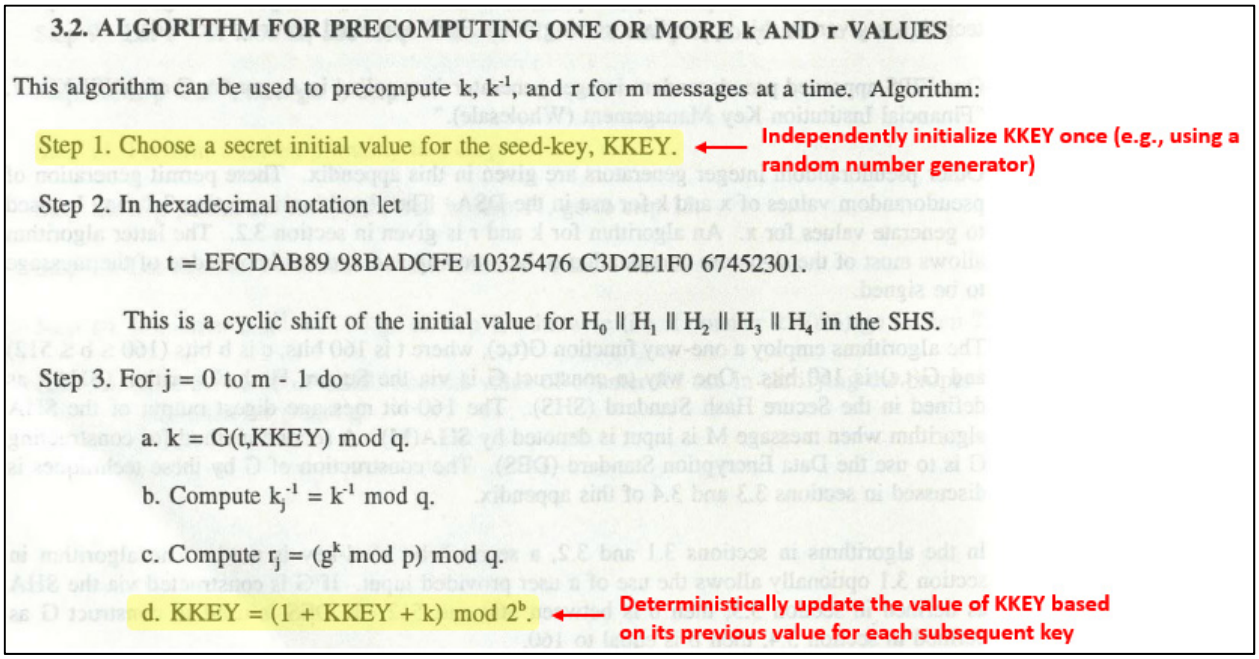


EX2001, ¶44 (annotating Dr. Katz's hindsight example at ¶152)[10].

As a consequence of Petitioners' hindsight proposal, each time the loop repeats (i.e., whenever the value of H_KKEY is not less than q), the random number generator must be accessed to obtain a new value for KKEY and that value must thereafter be transformed (e.g., hashed) according to the function G(). Pet., 40-42. But, this approach is unnecessary, unsupported, and even in tension with DSS's express teaching to update KKEY using a simple deterministic function rather than a random number generator. EX2001, ¶¶46-54.

[10] *Supra*, footnote 2.

For example, Petitioners' approach is unnecessary because the one-way functions disclosed in DSS (e.g., the SHA hash function and the DES block cipher) already produce an unpredictable output that obscures the input to the function. EX2001, ¶47; EX1004, 17-18. In other words, after an iteration in which H_KKEY is deemed too large, simply incrementing KKEY or otherwise updating the preceding value of KKEY with a deterministic function would have a comparable likelihood of producing a new H_KKEY value that is less than the threshold as if KKEY were independently updated without reference to its previous value using the random number generator. EX2001, ¶47. Neither DSS nor Rose disclose a comparable scenario where a random number generator and one-way function G() are repeatedly employed together in an iterative fashion. This is because it is typically sufficient for the input to the one-way function G() to be randomly selected just once when generating a given key. EX2001, ¶47. In fact, in an analogous scenario, DSS's prescribed algorithm for batch key generation independently initializes KKEY *only once* (at the start of the batch key generation) and thereafter applies a deterministic function to update KKEY for each subsequent key generated:



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 = EFCDAB89 98BADCFE 10325476 C3D2E1F0 67452301.

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

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

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

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

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

d. $KKEY = (1 + KKEY + k) \bmod 2^b$. ← Deterministically update the value of KKEY based on its previous value for each subsequent key

EX2001, ¶¶48-49 (annotating DSS's p. 18)[11]. It was error for the petition to ignore the stated preference of the primary reference; the Board should not do so. *Polaris Indus., Inc. v. Arctic Cat, Inc.*, 882 F.3d 1056, 1069 (Fed. Cir. 2018) (a primary reference's "statements regarding preferences are relevant to a finding regarding whether a skilled artisan would be motivated to combine that reference with another reference").

The petition's proposal to repeat the independent determination of KKEY using a random number generator thus represents an arbitrary, hindsight-driven selection not justified by any evidence, or even a purported benefit of doing so. It

[11] *Supra*, footnote 2.

is of no moment that Rose's original computer code repeatedly re-evaluated the random() function until an acceptable value was identified, especially because Rose's original code and Petitioners' proposed DSS-Rose algorithm operate in fundamentally different contexts. Rose's original code did not perform a one-way transformation G() (e.g., a hash function) on the value obtained from the random number generator, but the DSS-Rose algorithm does. The evidence here shows a POSITA would have recognized that addition of the one-way function G() obviates the need to repeatedly access the random number generator, and yet the petition's hindsight proposal does exactly that for no stated reason. EX2001, ¶¶46-52.

Indeed, a POSITA would have recognized that, depending on the specific system at issue, high-quality random number generators were typically more time-consuming to run than basic arithmetic operations performed on a stored value such as KKEY. EX2001, ¶51. This is especially the case where, as in DSS, the length of KKEY is not less than the output of the function to which it is seeding (i.e., the one-way function G()). EX1004, 17-18; *see also In re SuongHyu Hyon*, 679 F.3d 1363, 1366 ("It is impermissible within the framework of section 103 to pick and choose from any one reference only as much of it as will support a given position, to the exclusion of other parts necessary to the full appreciation of what such reference fairly suggests to one of ordinary skill in the art."); *Plas-Pak Indus., Inc. v. Sulzer Mixpac AG*, 600 Fed. Appx. 755, 759 (Fed. Cir. 2015) ("Such a

change in a reference's 'principle of operation' is unlikely to motivate a person of ordinary skill to pursue a combination with that reference").

Considering these advantages in efficiency of a deterministic approach to updating KKEY, and in view of DSS's prescription for a deterministic approach in the context of batch key generation, the evidence here shows that a POSITA would not have been prompted to pair together both the random number generator and the one-way function G() in the do-while loop. At a minimum, it was Petitioners' burden to explain why a POSITA would have paired the functions together in the manner proposed, but it failed to do so. Accordingly, the "information presented in the petition" is simply not enough to satisfy Petitioner's burden under §314(a)—especially where Petitioners cannot subsequently add theories/arguments that should have been part of the petition. *Intelligent Bio-Systems, Inc. v. Illumina Cambridge, Ltd.*, 821 F.3d 1359, 1369 (Fed. Cir. 2016); *In re Magnum Oil Tools Int'l*, 829 F.3d 1364, 1380 (Fed. Cir. 2016). This failure, along with the others described above and below, are fatal to Ground 1, and institution should be denied.

B. The Petition Falsely Contends That A Limited Number Of Solutions To Mitigating Modulo Bias Would Have Rendered Rose's Approach Obvious To Try, As Evidenced By The Fact That Subsequent Versions Of DSS Addressed The Modulo Bias Problem Using Multiple Approaches Fundamentally Different From Rose

The petition contends that Rose's technique would have been "obvious to

try" because, as assumed by its declarant Dr. Katz, 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. Implicit in Petitioners' statement is that there exists only one viable solution to obtaining a random number in a desired range without introducing bias in the selection of that number. But this statement is not only demonstrably false, it is also misleading.

While the invention of the '961 patent is now (today) recognized as being the most elegant solution with benefits of simplified implementation, Petitioners fatally ignore the fact that in the past (and without hindsight) the modulo bias problem could have been addressed in a number of ways never contemplated by the petition's false dichotomy of limited choices, including solutions within either category (e.g., modular reduction, and trial-and-error), both categories, or neither. EX2001, ¶66; *generally id.*, ¶¶66-74. The Petitioners thus failed to meet their burden because, as held by the Federal Circuit, an invention is only "obvious to try" when the challenger can actually demonstrate that "there [were] a finite number of identified, predictable solutions" to a known problem. *Sanofi-Aventis Deutschland GmbH v. Glenmark Pharms., Inc.*, 748 F.3d 1354, 1360 (Fed. Cir.

2014). As the *Sanofi* court explained, the alleged "obvious to try" path:

> must "present a finite (and small in the context of the art) number of options easily traversed to show obviousness." *Ortho-McNeil Pharm., Inc. v. Mylan Labs., Inc.*, 520 F. 3d 1358, 1364 (Fed. Cir. 2008). As illustrated in *In re O'Farrell*, 853 F.2d 894, 903 (Fed. Cir. 1988), it would not be "obvious to try" when "the prior art gave either no indication of which parameters were critical or no direction as to which of many possible choices is likely to be successful."

Sanofi, 748 F.3d at 1360. Similarly here, the petition wrongly assumes the number of options at the time, and the prior art cited in the petition "gave either no indication of which parameters were critical or no direction as to which of many possible choices is likely to be successful." *Id.*

In more detail, Petitioners' assumption that Rose's technique was somehow the only alternative available to DSS's original key generation technique is belied by the fact that subsequent versions of the DSS standard were updated to address the modular bias vulnerability in several ways—and *none* of the revised standards adopted Rose's specific approach. EX2001, ¶67. For example, some 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. In this release, the prescribed algorithm for generating the user's per-message secret key *k* was updated according to the operations 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 = EFCDAB89 98BADCFE 10325476 C3D2E1F0 67452301.

This is a cyclic shift of the initial value for $H_0 \| H_1 \| H_2 \| H_3 \| H_4$ 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 $k = (w_0 \| w_1) \bmod q$

EX2002, 73 (highlighting added)[12].

Contrary to the false dichotomy framed by Petitioners and Dr. Katz, the FIPS 186-2 revision was able to mitigate the security vulnerability identified by Bleichenbacher not by abandoning modular reduction altogether, but instead by doubling the length of the dividend that was subject to the modular reduction. EX2002, ¶¶68-69. In particular, the dividend for computing *k* in earlier versions of DSS was simply the output of G(t, KKEY), which was 160 bits long—the same length as *q* itself. EX1004, 17. But in the October 2001 revision of FIPS 186-2,

[12] *Supra*, footnote 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; EX2001, ¶69. The result of increasing the dividend in this manner is to substantially decrease the impact of bias from the modular selection to a statistically insignificant level that could not practically be leveraged to compromise the user's long-term private key. EX2001, ¶69 ("Given that N is a 160-bit number, this is a truly dramatic reduction of bias representing a reduction by billions and billions.") (citing EX2005). Thus, despite presumable knowledge of techniques like that disclosed in Rose, skilled artisans serving as stewards of the DSS standard were not prompted to adopt Rose's approach during that time.

Later, in June 2009, the DSS standard was updated yet again in the FIPS 186-3 release. *See* EX2003, ¶¶70-74. FIPS 186-3 revised the prescribed method for generating a user's per-message private key *k*, and specifically provided two distinct 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 = **len**(q); L = **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. ($status$, k^{-1}) = **inverse** (k, q).
8. Return *status*, k, and k^{-1}.

EX2003, 49 (highlighting added)[13].

In this method, 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."); EX2001, ¶71. This method is

[13] *Supra*, footnote 2.

again similar to the original DSS approach and categorically different from Rose.

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 = **len**(q); L = **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. ($status$, k^{-1}) = **inverse**(k, q).
9. Return *status*, k, and k^{-1}.

EX2003, 50[14].

While this method employs an iterative trial-and-error approach, it is profoundly different from the Petitioners' proposed DSS-Rose combination and

[14] *Supra*, footnote 2.

demonstrates that the petition's new theory of applying Rose to DSS was not the result of realistic engineering/design concerns at the relevant time, but was instead a hindsight-driven attempt that used the claims of the '961 patent as a roadmap. EX2001, ¶73. For instance, FIPS 186-3 recognized 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 number with a one-way function G() (e.g., a hash function)—let alone perform both operations repeatedly—as proposed in the Petitioners' combination (which purports to employ a true random number generator according to the teachings of Schneier). EX2001, ¶73; *supra*, Section A.1. To do otherwise would have been inefficient and unnecessary. *Id.*

In sum, the several methods for key generation prescribed by subsequent releases of the DSS standard show that Petitioners' obviousness arguments are based upon a flawed assumption and legal error. *Sanofi*, 748 F.3d at 1360. Not only do these subsequent releases provide a range of historical solutions beyond those that the petition contemplated as allegedly being "obvious to try," but also none of the subsequent releases tracked either Rose's approach or the Petitioners' approach as set forth in its proposed DSS-Rose combination. *See also STIHL Inc. v. ElectroJet Technologies, Inc.*, IPR2018-00022, Paper 13, 20 (PTAB April 11, 2018) ("a critical part of the 'obvious to try' test as, showing 'a finite number of

identified, predictable potential solutions' to the problem at hand"). Accordingly, while the petition failed to justify multiple aspects of its proposed modification of DSS in view of Rose (*supra*, Section VI.A), subsequent releases of the DSS standard demonstrate that Petitioners' proposed combination was far from compelled—and was in fact rejected by skilled artisans of the past. The Petitioners thus failed to meet their burden of demonstrating that were in fact "a finite number of identified, predictable solutions" to the bias problem. *Sanofi-Aventis*, 748 at 1360. For these additional reasons, Ground 1 is deeply flawed and institution should be denied.

VII. GROUND 2 BASED ON DSS IN VIEW OF SCHNEIER AND MENEZES IS DEFICIENT

Petitioners' second ground offers no discernible improvement over the first, and in fact suffers from many of the same deficiencies as those described above for Ground 1. As with Ground 1, Ground 2 relies on DSS for the framework of a cryptographic key generation algorithm, and cites Schneier for its disclosure of a random number generator that could allegedly be employed to generate a seed value (e.g., to generate KKEY in DSS). Pet., 47 ("Ground 2 is similar to Ground 1 except that, instead of Rose, Ground 2 relies on the rejection sampling technique of Menezes …"). The only formal difference in the second ground is the substitution of Menezes in place of Rose, although the underlying teachings of these references

are largely the same.

As such, the proposed combination of DSS and Menezes[15] rests on similarly unsupported premises as the DSS-Rose combination, including the flawed assumption that any of the cited references would have led a POSITA to independently re-generate a seed value (e.g., KKEY in DSS) for a hash function (e.g., G(t, KKEY) in DSS) *each time* a previously generated hash value falls outside a desired range. EX2001, ¶¶53-65. Worse, Menezes never connects its disclosure to the bias problem at the heart of the Bleichenbacher's DSS vulnerability, and the petition fails to establish that one of ordinary skill would have had any basis to recognize Bleichenbacher's teaching as even an (allegedly) possible solution to it. *Id*. As described in further detail below, the errors in Ground 2 prevent Petitioners from demonstrating a reasonable likelihood of prevailing against any of the Challenged Claims.

A. Menezes Does Not Disclose Or Suggest The Claimed Solution, And A POSITA Would Not Have Looked To Menezes To Apply In Combination With DSS

As noted above (Section III.C), Menezes is a textbook on the subject of applied cryptography, but the petition relies in substance on just a narrow slice of disclosure from the following paragraph:

[15] *Supra*, footnote 5.

> **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[16]; Pet., 48-49.

Ground 2 cites this disclosure in combination with DSS, but since DSS merely discloses the same known key generation algorithm that the inventors of the '961 patent specifically improved upon, the petition effectively argues that the above two sentences from Menezes disclose at least ***five*** limitations from claim 1 as recited in elements 1[c], 1[d], 1[e], 1[f], and 1[g].[17] Pet., 50-53, 55-56. But this is simply not the case. Petitioners' arguments stretch this limited disclosure from Menezes well beyond its text and well beyond any inspiration that a POSITA allegedly would have taken from it. EX2001, ¶55-56; *see also*, ¶33.

In particular, Menezes' disclosure is remarkable not for what it *does* say but rather for what it *does not*—namely, anything at all about ensuring unbiased selections of *hashes* of random numbers in a manner analogous to claim 1. EX2001, ¶33. The petition fails to explain why Menezes' technique for accepting or rejecting a "random integer" based on whether it lies within a specific range of

[16] *Supra*, footnote 2.

[17] *See supra*, Footnote 6.

values (e.g., [0, *n*]) would have been applicable to the techniques in the '961 patent for accepting or rejecting the result of a *hash* of a randomly generated seed value (e.g., H(SV)). *See* EX1001, 5:41-42 (claim 1: "determining whether said output H(SV) is less than said order q"). Claim 1 specifically provides distinct steps for generating a seed value from a random number generator, hashing the seed value, and then determining whether to accept or reject the hash of the seed value as key *k*. *Id.*, 5:33-51. Yet, the petition does not apply Menezes' teaching against the step in claim 1 where a random number is actually generated (i.e., "generating a seed value *from a random number generator*"); instead, Petitioners map Menezes to the subsequent steps in claim 1 for accepting or rejecting the *hash* of the randomly generated seed value. Pet., 50-54.

For Menezes to be applied in this different context, Petitioners were required to show that a POSITA would have understood Menezes' teaching as being transferable from the context of using a random bit generator for creating a random integer to the distinct context of evaluating hashes of random numbers output by a random number generator. Furthermore, it was Petitioners' burden to show that one of ordinary skill would have been motivated (without the benefit of hindsight from the '961 patent) to apply Menezes' teaching in this different context. *Power Integrations, Inc. v. Fairchild Semiconductor Int'l, Inc.*, 711 F.3d 1348, 1368 (Fed. Cir. 2013) ("the prejudice of hindsight bias" often overlooks that the "genius of

invention is often a combination of known elements which in hindsight seems preordained"). Yet, the petition failed to meet its burden in either regard. The petition merely advances a conclusory assertion that "Menezes, in combination with DSS, discloses 'determining whether said output H(SV) is less than said order q prior to reducing mod q" and "[u]nder the combination with DSS, the 'resulting integer in the passage of Menezes quoted above corresponds to 'said output H(SV),' and the interval from 0 to *n* corresponds to the interval from 0 to q-1 specified by the 'order q.'" Pet., 51. But critically, the petition never grapples with the different contexts at issue: a random bit generator for creating a random integer in the absence of hashing (Menezes) and hashes of seed values output by a random number generator ('961 claims). Indeed, the petition never establishes that the reasons allegedly motivating use of Menezes' techniques (e.g., to obtain a random integer within a desired range of values from a random bit generator) would have similar force in the context of *hashes* of random numbers that were output by a random number generator. And, as explained above (Section III.C), Menezes' perfect random bit generator ensured that no hashing would be required to reduce or eliminate bias in its output. But the solution claimed in the '961 patent does not assume that a perfect generator is available, and therefore takes a fundamentally different approach to mitigating bias by generating a seed value with a random number generator and then hashing the seed value before evaluating

whether the output falls within a desired range. EX2001, ¶33.

Beyond the unaddressed shifts in context, the petition commits several additional errors as well. For example, the petition alleges that Menezes' step of converting a random bit sequence to an integer somehow corresponds to a hashing operation as claimed. Pet., 51 ("the 'resulting integer' in the passage of Menezes quoted above corresponds to 'said output H(SV)"). But this is simply an absurd contention. The conversion of a bit sequence from one data type (e.g., a bit sequence) to an integer type would have been understood as a low-level computer operation for formatting the bit sequence in an integer format—an operation that bears no resemblance in function or purpose to a hash function (e.g., SHA in DSS). EX2001, ¶32.

Moreover, the petition errs in alleging that a POSITA "would [] have been motivated to use the technique in Menezes to solve a particular problem in implementing DSS—ensuring that the resulting *k* value falls within the desired range of 0 to q-1." Pet., 54. But a POSITA would have had no reason in the first place to look for ways to ensure that "the resulting *k* value falls within the desired range," especially since DSS's operation for reducing G(t, KKEY) mod q already ensured this was the case (i.e., that the resulting *k* value falls within the desired range). Indeed, even the petition never argues that there was any problem with DSS's ability to find a value for key *k* within the desired range (e.g., [0, q-]) for

which Menezes's technique would have been needed. *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 567 F.3d 1314, 1326 (Fed. Cir. 2009) ("An inference of nonobviousness is especially strong where the prior art's teachings undermine the very reason being proffered as to why a person of ordinary skill would have combined the known elements.").

For each of these reasons, Menezes is a flawed reference that fails to align with recited features of the '961 claims and that a POSITA would never have been prompted to combine with DSS before the effective filing date of the '961 patent (December 27, 2000). These flaws, along with the additional problems described below (Sections VII.B, VII.C) render Ground 2 critically deficient. Institution should be denied.

B. Petitioners Fail To Support Their Modifications In The DSS-Menezes Combination, Including The Modification For Repeatedly Determining A Random KKEY Value And Hashing The KKEY Value

As with the cited combination in Ground 1, the references relied upon in Ground 2 fail to disclose "if said output H(SV) is rejected, repeating said method," and in particular fails to disclose that the repetition of the method would include both "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)." EX2001, ¶¶57-65; *see also supra*, Section VI.A.2. There is no dispute

that DSS lacked these features, especially where its approach to resolving a key *k* within the desired range [0, q-1] involved modular reduction rather than an iterative technique. EX1004, 17-18. Menezes also did not disclose repetition of the operations for generating a seed value and hashing the seed value since Menezes was merely concerned with generating a random number rather than performing an extra hashing operation and determining whether to accept or reject the hashed value. *See* EX1006, 69 ("[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."). Schneier was not relied upon for these features, and also fails to remedy the shortcomings of DSS and Menezes.

Despite the failure of any reference cited in Ground 2 to teach repetition of a method that includes randomly generating a seed value and hashing that value in a corresponding manner to claim 1, the petition nevertheless asserts that these features would have been provided in the combination. For example, the petition argues that the DSS-Schneier-Menezes combination "would have predictably resulted in the technique of generating *k* in DSS with the following steps:"

> (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.

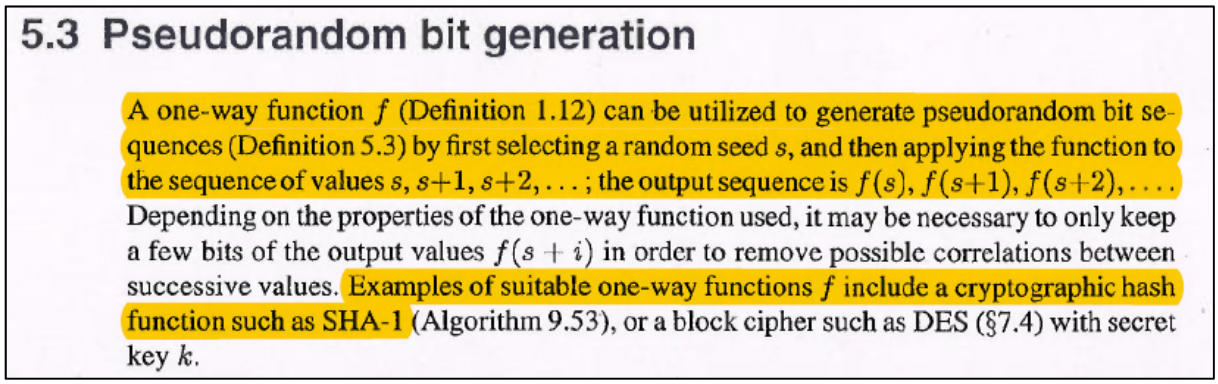
Pet., 53-54 (emphases in original) (citing EX1002, ¶185).

It should be noted that the proposed combination involves a subtle, but necessary, sleight of hand in order for the petition to map to all the features of claim 1. Specifically, at step (5), the petition alleges that it would have been obvious according to the teachings of Menezes to return to step (1) if G(t,KKEY) is rejected. *Id.* But as noted above, neither Menezes nor DSS nor Schneier teach that the random generation of a seed value and hashing of that seed value should be performed together in an iterative fashion. EX2001, ¶¶57-59. At most, Menezes teaches that—*in the absence of a subsequent hashing operation*—a new random number can be repeatedly obtained until an acceptable value is identified within

the desired range. EX1006, 69. But a POSITA would have recognized that the addition of a hashing operation represented a material change to Menezes' algorithm, and in this context it would not have been obvious to both re-generate the seed value using a random number generator and hash the seed value in each iteration after the result of a preceding iteration was rejected. EX2001, ¶¶33, 59-60. Critically, the petition never explains, and presents no specific evidence to justify, its conclusory assertion that a POSITA would have combined the teachings of DSS, Schneier, and Menezes in a manner that would repeat both of these operations in a manner corresponding to claim 1.

Worse, the petition's proposal to repeat both random seed value generation and hashing was not only unnecessary to achieve an operable combination, but also contradicts express teachings in DSS and Menezes. It is unnecessary because the hash function disclosed in DSS (i.e., SHA) already produces an unpredictable output that obscures the input to the function. EX2001, ¶¶61-62; EX1004, 17-18. In other words, after an iteration in which G(t,KKEY) is rejected for being outside the desired range of values for *k*, simply incrementing KKEY or otherwise updating the preceding value of KKEY with a deterministic function would have a comparable likelihood of producing a new G(t,KKEY) within the desired range as if KKEY were independently updated without reference to its previous value using a random number generator. EX2001, ¶62.

The petition's proposal then contradicts express preferences in both DSS and Menezes for employing deterministic functions to update the input to an iteratively executed hash function. For example, Menezes explains that a one-way function such as the SHA-1 hash function (i.e., the same function prescribed for hashing in DSS) can be repeatedly evaluated to obtain a pseudorandom bit sequence simply by incrementing the input s to the function by one unit in each iteration:



5.3 Pseudorandom bit generation

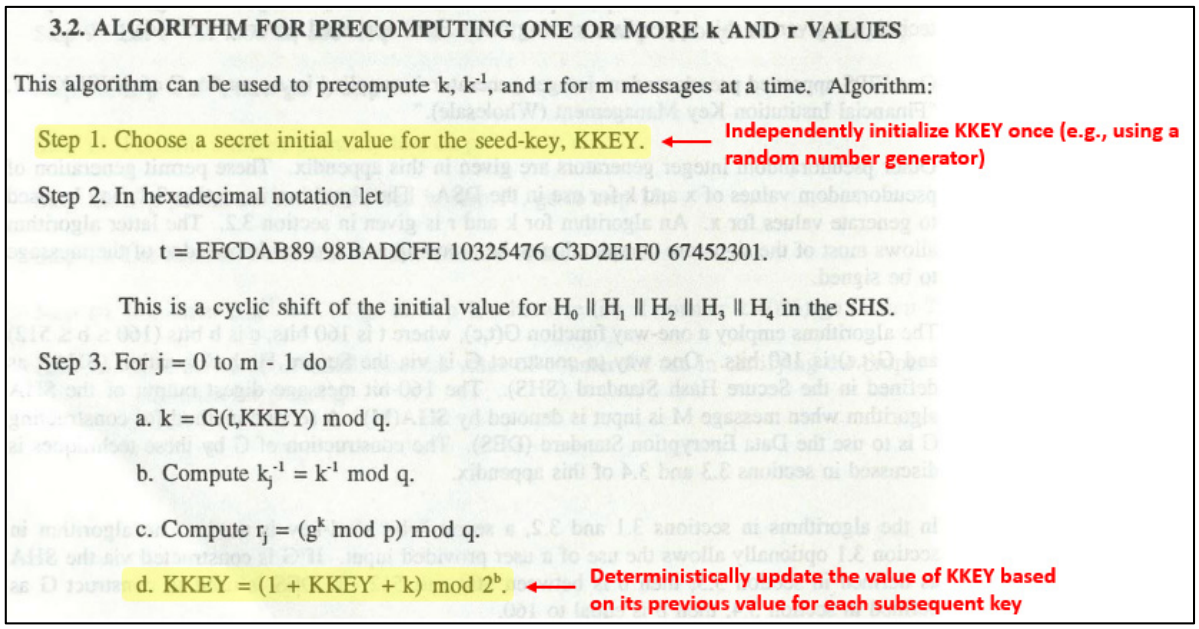
A one-way function f (Definition 1.12) can be utilized to generate pseudorandom bit sequences (Definition 5.3) by first selecting a random seed s, and then applying the function to the sequence of values $s, s+1, s+2, \ldots$; the output sequence is $f(s), f(s+1), f(s+2), \ldots$. Depending on the properties 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)[18]; EX2001, ¶¶63-64. Even if some correlation could have been derived between successive values as alluded to in the above passage, this would have been of no concern for a combination with DSS so long as the initial value of KKEY was independently obtained with a random number generator since a succession of hash values stemming from a same original input would be unavailable to third parties for inspection. Thus, Menezes' approach for

[18] *Supra*, footnote 2.

deterministically incrementing the input to the hash function would have been even more secure in the combination with DSS than in the context described in Menezes. EX2001, ¶64; *Arctic Cat Inc. v. Bombardier Recreational Prods. Inc.*, 876 F.3d 1350, 1363 (Fed. Cir. 2017) ("Evidence suggesting reasons to combine cannot be viewed in a vacuum apart from evidence suggesting reasons not to combine.").

Similarly, in an analogous scenario, DSS's prescribed algorithm for batch key generation independently initializes KKEY *only once* (at the start of the batch key generation) and thereafter applies a deterministic function to update KKEY for each subsequent key generated:



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 = EFCDAB89 98BADCFE 10325476 C3D2E1F0 67452301.

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

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

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

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

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

d. $KKEY = (1 + KKEY + k) \bmod 2^b$. ← Deterministically update the value of KKEY based on its previous value for each subsequent key

EX2001 (annotating DSS's p. 18)[19].

The petition's proposal to repeat the independent determination of KKEY using a random number generator thus represents an arbitrary, hindsight-driven design choice not justified by any evidence, or even a purported reason to do so. Indeed, a POSITA would have recognized that, depending on the specific system at issue, random number generators were often more computationally expensive or time-consuming to run than simple arithmetic operations performed on a stored value such as KKEY (e.g., as would be performed in Menezes' suggested incrementing function). EX2001, ¶64. This is especially the case where, as in DSS, the length of KKEY is not less than the output of the function to which it is seeding (i.e., the one-way function G()). EX1004, 17-18; *SuongHyu Hyon*, 679 F.3d at 1366 ("impermissible within the framework of section 103 to pick and choose"); *Plas-Pak*, 600 Fed. Appx. at 759 ("a change in a reference's 'principle of operation' is unlikely to motivate").

Considering these advantages in efficiency of a deterministic approach to updating KKEY, and in view of both Menezes' and DSS's prescription for deterministic approaches to iteratively evaluating hash functions, the evidence here

[19] *Supra*, footnote 2.

shows that a POSITA would not have been led to pair together both the random number generator and the one-way function G(t, KKEY) in the loop that is repeated after each rejected (out-of-range) value. *Polaris*, 882 F.3d at 1069 (a primary reference's "statements regarding preferences are relevant to a finding regarding whether a skilled artisan would be motivated to combine that reference with another reference").

At a minimum, it was Petitioners' burden to explain why a POSITA would have paired the functions together in the manner proposed, but it failed to do so. Here again, the "information presented in the petition" is simply not enough to satisfy Petitioner's burden under §314(a)—especially where Petitioners cannot subsequently add theories/arguments that should have been part of the Petition. *Intelligent Bio-Systems*, 821 F.3d at 1369. This failure, along with the others described above, are fatal to Ground 2, and institution should be denied.

C. The Petition Falsely Contends That A Limited Number Of Solutions To Mitigating Modulo Bias Would Have Rendered Menezes's Approach Obvious To Try, As Evidenced By The Fact That Subsequent Versions Of DSS Addressed The Modulo Bias Using Multiple Approaches Fundamentally Different From Menezes

The petition contends that Menezes' technique would have been "obvious to try" because, as stated by its declarant Dr. Katz, 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 'modulo operation' (*i.e.*, modular reduction) on the number from the underlying generator, or (2) reject the number from the underlying generator if it falls outside the desired range, and try again (as disclosed in Menezes)." Pet., 54-55; EX1002, ¶189. Implicit in Petitioners' statement is that there exists only one viable solution to obtaining a random number in a desired range without introducing bias in the selection of that number. But this statement is not only demonstrably false, it is also misleading for the reasons explained above. *Supra,* Section VI.B.

Namely, even though the invention of the '961 patent is now (today) recognized as being the most elegant solution with benefits of simplified implementation, Petitioners ignore that in the past (and without hindsight) the modulo bias problem could have been addressed in a number of ways never contemplated by the petition's false dichotomy of limited choices, including solutions within either category (e.g., modular reduction, and trial-and-error), both categories, or neither. EX2001, ¶66; *generally* ¶¶66-74. As such, Petitioner's "obvious to try" theory is flawed and fails to actually demonstrate that "there [were] a finite number of identified, predictable solutions" to a known problem. *Sanofi*, 748 F.3d at 1360; *supra*, Section VI.B. Further yet, Petitioners overlooked that the prior art cited in the petition "gave either no indication of which parameters were critical or no direction as to which of many possible choices is

likely to be successful." *Sanofi*, 748 F.3d at 1360.

Indeed, Petitioner's assumption that Menezes' technique was somehow the only alternative available to DSS's original key generation technique is belied by the fact that subsequent versions of the DSS standard were updated to address the modular bias vulnerability in several ways—and *none* of the revised standards adopted Menezes' specific approach. EX2001, ¶67. These approaches are outlined in detail above in Section VI.B, and none of them employed the specific approach described by Menezes or proposed in the Petitioners' DSS-Menezes combination. EX2001, ¶67. For example, in the October 2001 Change Notice to the FIPS 186-2 version of DSS, the key generation algorithm was updated to improve security by doubling the length of the dividend to the modulo operation—but the modular reduction approach was still retained. EX2002, 73; EX2001, ¶¶68-69. Then, in the June 2009 release of the FIPS186-3 version of DSS, two key generation methods were prescribed that addressed the security vulnerability identified by Bleichenbacher. EX2003, 49-50; EX2001, ¶¶70-74. The first method was based on a strategy of increasing the length of the dividend for a modular reduction similar to the October 2001 scheme. EX2003, 49; EX2001, ¶¶70-71. The second method was based on a strategy of repeatedly generating random numbers until one was identified within the desired range—but this strategy recognized that it was unnecessary to hash the random number at each

iteration and thus perform those operations in tandem as proposed in Petitioners' hindsight-driven approach. EX2003, 50; EX2001, ¶¶72-74. More broadly, each of these various approaches stand in contrast to the manner in which Petitioner proposed to combine DSS with Menezes and undermines Petitioner's assertion that a limited number of solutions would have rendered the claimed techniques obvious to try. The Petitioner thus failed to meet its burden of demonstrating that were in fact "a finite number of identified, predictable solutions" to the bias problem. *Sanofi-Aventis*, 748 at 1360. For these additional reasons, Ground 2 is fatally flawed and institution should be denied.

VIII. GROUNDS 3 AND 4 BASED ON COMBINATIONS CITING RAO AND FLOYD ARE DEFICIENT

The petition independently challenges independent claim 23 and dependent claims 24 and 27 based on combinations of DSS, Schneier, Rose, Rao, and Floyd (Ground 3) and DSS, Schneier, Menezes, Rao, and Floyd (Ground 4). Pet. 57-62. Claim 23 recites seven elements that identically correspond to limitations that are also recited in independent claim 1, and the petition relies on the teachings of DSS-Schneier-Rose or DSS-Schneier-Menezes for these elements in the same manner as Grounds 1 and 2. *Id.*

Rao and Floyd are purportedly added to the combinations in Grounds 3 and 4 to address the additional requirement for the cryptographic unit to include "an

arithmetic processor." *Id.* Critically, neither Rao nor Floyd are relied upon to remedy the deficiencies described in the preceding sections of this Preliminary Response for the limitations commonly recited in claims 1 and 23. Therefore, Patent Owner respectfully submits that Grounds 3 and 4 are fatally flawed for at least the same reasons as those identified in Grounds 1 and 2. For this additional reason, institution should be denied.

IX. THE BOARD SHOULD EXERCISE ITS DISCRETION UNDER 35 U.S.C. §314(a) TO DENY INSTITUTION DUE TO THE MATURE POSTURE OF THE LITIGATION, IN COMBINATION WITH THE PETITION'S KNOWN DEFECTS

Beyond the Petition's many flaws detailed above, multiple factors indicate that the Board should exercise its broad discretion under 35 U.S.C. §314(a) to deny institution, especially where Petitioners delayed in pursing IPR while the concurrent litigation matured towards a posture where the jury trial will resolve Petitioners' prior art invalidity allegations long before any final decision in this forum. *See E-One, Inc. v. Oshkosh Corp.*, IPR2019-00162, Paper 16, 7-20 (PTAB June 5, 2019) (denying institution because "the district court is set to complete trial before any final decision from the Board would be due" and "instituting a trial would be an inefficient use of Board resources."); *ZTE (USA), Inc. v. Fractus, S.A.*, IPR2018-01457, Paper 10, 15-16 (PTAB Feb. 28, 2019); *see also NHK Spring Co. Ltd. v. Intri-Plex Techs. Inc.,* IPR2018-00752, Paper 8, 19-20 (PTAB Sept. 12,

2018) (precedential) (denying institution, in part, because "the district court proceeding will analyze the same issues and will be resolved before any trial on the Petition concludes" and institution would not be consistent with the objective of the AIA "to provide an effective and efficient alternative to district court litigation").

Under the law, the Board is "never compelled" to institute IPR. *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1367 (Fed. Cir. 2016). Indeed, the Office has expressly authorized denial where (1) less than all grounds meet the "reasonable likelihood" standard; and (2) doing so would further the Board's statutory, 35 U.S.C. § 316(b), and regulatory, 37 C.F.R. § 42.1(b), efficiency goals. *See* SAS Q&A's, Part D, Effect of SAS on Future Challenges that Could Be Denied for Statutory Reasons (June 5, 2018); *Deeper, UAB v. Vexilar, Inc.*, IPR2018-01310, Paper 7 (PTAB Jan. 24, 2019) (Informative); *Chevron Oronite Company LLC v. Infineum USA L.P.*, IPR2018-00923, Paper 9 (PTAB Nov. 7, 2018) (informative).

Denial of institution in view of the petition's many defects will not prejudice Petitioners or prevent Petitioners from raising invalidity grounds in the upcoming

jury trial. Moreover, because Petitioners waited nearly a year[20] to initiate this IPR, the litigation has advanced to a mature posture where invalidity allegations will be resolved earlier in the district court proceeding. For example:

- the Honorable Judge Chu has already issued a Final Ruling on Claim Construction under the same *Phillips* standard the Board now employs;
- discovery in the concurrent litigation will close on August 30, 2019—certainly before any first discovery phase in this IPR proceeding; and
- the jury trial is scheduled for April 13, 2020, which will resolve Petitioners' invalidity allegations against the '961 patent long before any Final Written Decision in this proceeding.

Based on these facts, the Board's institution of IPR here would result in an unnecessary duplication of efforts for both parties and the Board.

The fact that Petitioners are not prohibited from seeking IPR within the one-year time bar of 35 U.S.C. 315(b) does not mean Petitioners are entitled to institution of IPR, nor does it justify the inefficiencies and duplication of resources introduced by its unexplained delay. The *NHK* case recently designated

---

[20] As noted in the petition, the initial complaint was filed on March 6, 2018, Petitioners were served on April 6, 2018, and the petition was not filed until April 3, 2019. *See* Pet., 1, 65.

"precedential" by the Office stands for this very proposition—i.e., where a petition is otherwise defective, inefficiencies introduced through overlapping district court proceedings may warrant denial. *See NHK*, Paper 8, 19-20. The Board in *E-One* also denied institution on a fact pattern similar to the present case. *See E-One*, Paper 16, 8-9. Much like the present case, the petition in *E-One* suffered from known flaws on the merits, and the mature posture of the concurrent litigation left no room for the IPR to add efficiency for the parties or the district court. *Id.*, 8-9 and 13-20.

For the many reasons discussed above, this petition is deeply flawed. Petitioners' analysis of the prior art overlooks multiple requirements of the claim elements of the '961 patent and erroneously relies upon hindsight-based reasoning to fill gaps in the cited references. Petitioners cannot explain away or repair these deficiencies without impermissibly shifting theories and positions. *Intelligent Bio-Systems*, 821 F.3d at 1369; *Magnum Oil*, 829 F.3d at 1380. These readily apparent flaws in the petition, in combination with the mature posture of the concurrent litigation, provide an overwhelming basis for the Board to exercise its discretion to deny IPR here for purposes of fairness and efficiency.

Simply put, institution in this case will not serve the AIA's objective of providing an effective and efficient alternative to district court litigation. In this case, institution will unnecessarily multiply the proceedings, wasting the resources

of both the parties and the Board. For this reason alone, the Board should exercise its discretion under 35 U.S.C. § 314(a) and deny institution.

X. CONCLUSION

For the foregoing reasons, Patent Owner respectfully requests that the Board deny institution of Grounds 1-4 of the Petition, and thus decline to institute *inter partes* review of any claim of the '961 patent.

Respectfully submitted,

Date: August 9, 2019

/Michael T. Hawkins/
Michael T. Hawkins, Reg. No. 57,867
Attorney for Patent Owner,
Blackberry Limited

Customer Number 26191
Fish & Richardson P.C.
Telephone: (612) 337-2508
Facsimile: (612) 288-9696

CERTIFICATION UNDER 37 C.F.R. §42.24(d)

Under the provisions of 37 C.F.R. §42.24(d), the undersigned hereby certifies that the word count for the foregoing Patent Owner's Preliminary Response to Petition totals 13,068, which is less than the 14,000 allowed under C.F.R. §42.24(b)(1).

Respectfully submitted,

Date: August 9, 2019

/Michael T. Hawkins/
Michael T. Hawkins, Reg. No. 57,867
Attorney for Patent Owner,
Blackberry Limited

Customer Number 26191
Fish & Richardson P.C.
Telephone: (612) 337-2508
Facsimile: (612) 288-9696

CERTIFICATE OF SERVICE

Pursuant to 37 C.F.R. §§42.6(e), the undersigned certifies that on August 9, 2019, a complete and entire copy of this Patent Owner's Preliminary Response was provided via email to Petitioners by serving the correspondence email address of record as follows:

Heidi L. Keefe
Andrew C. Mace
Mark R. Weinstein
Yuan Liang
Cooley LLP
Attn: Patent Group
1299 Pennsylvania Ave, NW, Suite 700
Washington, DC 20004
Email: hkeefe@cooley.com
Email: amace@cooley.com
Email: mweinstein@cooley.com
Email: yliang@cooley.com

/Jessica K. Detko /
Jessica K. Detko
Fish & Richardson P.C.
60 South Sixth Street, Suite 3200
Minneapolis, MN 55402
(612) 337-2516