UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

Samsung Electronics Co., Ltd.,

Petitioner,

v.

Ward Participations B.V.,

Patent Owner.

_____

PGR2022-00007
Patent 10,992,480

_____

PATENT OWNER'S RESPONSE
UNDER 37 C.F.R. § 42.220

Mail Stop PATENT BOARD
Patent Trial and Appeal Board
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, Virginia 22313-1450

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases

Statutes

Exhibit No.	Exhibit Title
2009	Pages from Dictionary of Computer & Internet Terms
2010	Pages from Andrew S. Tanenbaum - Modern Operating Systems
2011	Pages from Microsoft Computer Dictionary
2012	Expert Declaration Prof. Dr.Ing. Bart Preneel

I. INTRODUCTION

The Board instituted post-grant review of the ’480 Patent on five grounds: (i) The ‘480 Patent is eligible for PGR because at least one claim has a priority data after March 16, 2013; (ii) claims 1-19 under 35 U.S.C. § 112(a) for alleged lack of written description support; (iii) claims 1,3,6 11, 14-19 under 35 U.S.C. § 103 based on Ellison in view of Muir; (iv) claims 7,9,13 under 35 U.S.C. § 103 over Ellison in view of Muir and Al-Salqan; (v) claims 8,10 and 12 under 35 U.S.C. § 103 over Ellison in view of Muir and Searle. Pursuant to Rule 42.220, Patent Owner, Ward Participations B.V., submits this Patent Owner Response.

II. THE ‘480 PATENT (EX. 1001)

A. Overview of the ‘480 patent

The application No. 16/597,773 (the “’773 application”) that issued as the ‘480 Patent was filed on October 9, 2019, and is a continuation of U.S. Patent Application No. 10/560,579, filed as PCT Application No.PCT/NL2004/00042(the “’422 application”) filed on June 14, 2004, which further claims priority from Dutch Application PCT/NL03/00436, filed on June 13, 2003. The ’480 patent is a transition application because it was filed after March 16, 2013 and claims priority to application filed before March 16, 2013.

The ‘480 Patent is titled “Method and System for Performing a Transaction and for Performing a Verification of Legitimate Access to, or Use of Digital Data” and discloses an authentication software to generate a digital signature from authentication data. Ex. 1001, Abstract. See Ex. 2012, ¶¶ 35-39, 41-42.

The ‘480 Patent explains three issues on electronic transaction through network as follows:

(1) The data and information exchanged in said transactions may be legally privileged or protected by copyright, for example. However, digital information may be very easily copied and spread without a trace of who illegally copied and spread the data. Ex. 1001, 1:14-18.

(2) Using personal identifiers over public networks like the Internet, there is a possibility that the personal identifier becomes known to another person, enabling this other person to do transactions or gain access to digital data presenting himself as somebody else. If a problem arises after completion of the transaction, it is not possible to track the real transaction partner, as its personal identifier may have been used by a malicious user of the public network. Ex. 1001, 1:27-35.

(3) A digital signature originating from a random token and being different for every subsequent transaction is virtually impossible to forge and therefore uniquely identifies the transaction party. But additional hardware, e.g. a token and

a token reader, should be supplied to every possible transaction party. Ex. 1001, 1:45-52.

To address these issues, the ‘480 Patent describes a method for performing an electronic transaction between a first transaction party and a second transaction party using an electronic device operated by the first transaction party. The method comprises providing authentication data in a memory of the electronic device, the authentication data are inaccessible to a user of the electronic device; providing authentication software in the electronic device, the authentication data being accessible to said authentication software; activating the authentication software to generate a digital signature from the authentication data; providing the digital signature to the second transaction party. Ex. 1001, 1:57-2:3. See Ex. 2012, ¶¶ 46-47.

An illustration of the embodiment of the ’480 patent’s device configuration is depicted in Figure 3, reproduced below:

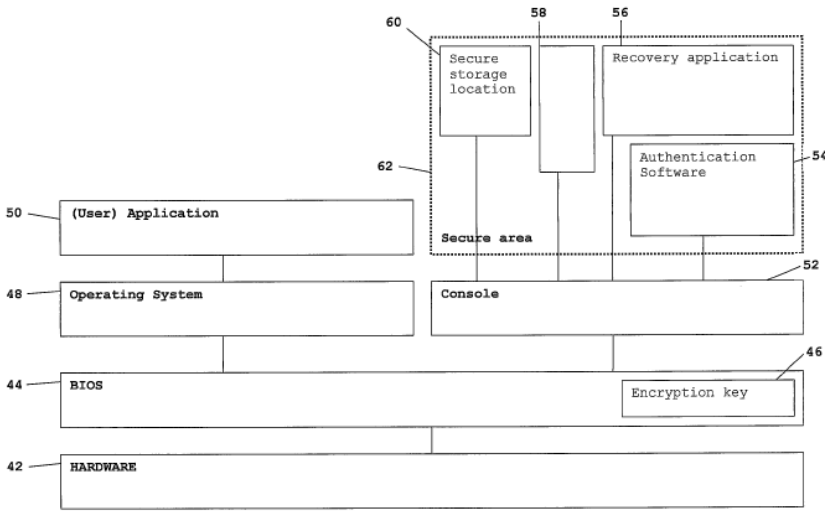


FIG. 3

As shown in Figure 3, the device consists of hardware 42, examples of hardware components being a processor, a hard drive, or other memory and a keyboard Ex. 1001, 8:14-17.

All the hardware devices are controlled by the BIOS 44. BIOS 44 is a software package stored in a read-only-memory (ROM) located on a main board Ex. 1001, 8:15-21. The BIOS 44 controls most of the instructions and data flowing from one component to another. Data or instructions coming from one component and addressed to another need to pass through the BIOS 44. The BIOS 44 may check whether the instructions to the other component are allowed or not. Ex. 1001, 8:22-27. BIOS 44 may comprise encryption key 46 used to encrypt data and only another party who knows encryption key 46 may be able to decrypt the data. Ex. 1001, 8:29–32.

In cryptography private keys are either bit strings, that is, finite sequences consisting of binary zeroes and ones or algebraic objects (such as two prime numbers, finite field elements) that can be represented as a finite number of bit strings. The length in bits of a cryptographic key is an upper bound for the security level of the cryptographic algorithm; for a well-designed symmetric key algorithm, it is also a lower bound for the security level. Private keys are thus not lines of code or programs. If a bit string is encrypted, the result is another bit string of the same or larger length. The same holds for authentication data in the '480 patent: they are plain bit strings and thus not lines of code or programs. Both terms are therefore used interchangeably. All the above elements are familiar to a POSITA. See Ex. 2012, ¶ 46.

Operating system 48 creates a run-time environment for any user application, and is an interface between the user and hardware 42. Ex. 1001, 8:33–36. When a user instructs operating system 48 to run application 50, application 50 requests data from operating system 48, and operating system 48 in turn requests the data through BIOS 44 from hardware 42, i.e., the hard drive, wherein the data coming from hardware 42 pass through BIOS 44 and operating system 48 before they arrive at requesting application 50. Ex. 1001, 8:36–43. The operating system 48 may be prohibited by the BIOS 44 to access certain parts of a hard drive or to start certain applications. Ex. 1001, 8:44–48.

The console 52 is an environment which runs all the processes in the computer without user interruption. Ex. 1001, 8:49–56. Any instructions coming from the operating system 48 intended for the console 52 may be blocked by the BIOS 44. Ex. 1001, 8:59–61.

In or behind console 52, there is secure area 62 only accessible to BIOS 44, secure area 62 comprising applications and storage locations not reported to operating system 48. Ex. 1001, 8:62–65. Secure area 62 comprises secure storage location 60, and authentication software 54, wherein an authentication table may be securely stored in secure storage location 60 in a part of the computer commonly seen as part of BIOS 44 but the authentication table is unreachable for operating system 48, and thus for a user. Ex. 1001, 9:7–11. With the authentication table securely stored in secure storage location 60, authentication software 54 should run in an environment in BIOS 44, thus preventing the authentication table from being accessible to operating system 48 at any time. Ex. 1001, 9:12–16. Therefore, authentication software 54 may be installed in an application environment in secure area 62 such that it may obtain the authentication table without passing through an unsecured part of the device. Ex. 1001, 9:16–19.

As for the "secure storage location", the '480 Patent further describes that the encoded authentication table is stored in a secure part of the BIOS, where it may only be retrieved by the BIOS and not by the operating system. This secure

part of the BIOS may also be any other, secure location in the device. For instance, it may be a separate part of a hard drive of a computer, which part may not be accessible to the operating system, but only to the BIOS and/or authentication software. Ex. 1001, 6:45–55.

Regarding authentication software, the '480 Patent describes "Authentication software 54 preferably has its own unique serial number, software identification (ID) and/or private encryption key. Said unique number may be advantageously employed in the installation method and in a track and trace method". Ex. 1001. 5:39–42. In a further embodiment, authentication software 54 may be provided with a system for signing a digital signature using a part of a software identification number (software ID) or any other application identifying character string, hereinafter referred to as a private key. Ex. 1001. 13:31–36.

B. Claim Construction

We construe each claim "in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent." 37 C.F.R. § 42.100(b) Under this standard, claim terms are generally given their plain and ordinary meaning as would have been understood by a person of ordinary skill in the art (POSITA) at the time of the invention and in the context of the entire patent disclosure. *See Phillips v. AWH Corp*., 415 F.3d 1303, 1313 (Fed. Cir. 2005).

The ordinary and customary meaning of a term may be evidenced by a variety of sources, including the words of the claims themselves, the specification, drawings, and prior art. However, the best source for determining the meaning of a claim term is the specification – the greatest clarity is obtained when the specification serves as a glossary for the claim terms. The words of the claim must be given their plain meaning unless the plain meaning is inconsistent with the specification. *In re Zletz*, 893 F.2d 319, 321, 13 USPQ2d 1320, 1322 (Fed. Cir. 1989); *Chef America, Inc. v. Lamb-Weston, Inc.,*358 F.3d 1371, 1372, 69 USPQ2d 1857 (Fed. Cir.2004).

No formal claim constructions are necessary 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). See Ex. 2012, ¶¶ 50-52.

III. LEVEL OF ORDINARY SKILL IN THE ART

For purposes of this Response, Patent Owner does not dispute Petitioner's POSITA definition. (Petition, 5; Ex. 1003, ¶¶15-16.) See Ex. 2012, ¶¶ 50-52.

IV. THE '480 PATENT IS NOT ELIGIBLE FOR PGR

A patent that issues from a transition application is not available for post-grant review if the claimed subject matter complies with the written description

and enablement requirements of § 112(a) for an ancestor application filed prior to March 16, 2013. See MPEP § 2159.04.

The Board held in the Decision of Instituting Post-Grand Review that the '480 Patent is eligible for PGR because the disclosures of the applications in the '480 patent priority chain do not reasonably convey to an artisan of ordinary skill that the inventor had possession of 1) "providing a private key in a memory of said electronic device, wherein said memory being a secure part of a Basic In Out System or any other secure location in said electronic device operated by the first transaction party," wherein 2) "[the] private key is inaccessible to a user of said electronic device," wherein 3) "the private key is encrypted when the private key is stored in said memory," and wherein 4) "the private key is decrypted in a secure processing environment inaccessible to said user and to any user-operated software," as required by claim 1. (Dec., 10-11)

A patent gains the benefit of an earlier filed application under 35 U.S.C. § 120 when each application in the chain leading back to the earlier application complies with the written description requirement of 35 U.S.C. § 112(a). *Zenon Envtl. Inc., v. U.S. Filter Corp.*, 506 F.3d 1370, 1378 (Fed. Cir. 2007).

The written description requirement of § 112 is a "possession" test. It requires a patentee to "describe the invention sufficiently to convey to a person of skill in the art that the patentee had possession of the claimed invention at the time

of the application, *i.e.,* that the patentee invented what is claimed." *LizardTech, Inc. v. Earth Res. Mapping, Inc.*, 424 F.3d 1336, 1345 (Fed. Cir. 2005). The "test requires an objective inquiry into the four corners of the specification from the perspective of a person of ordinary skill in the art." *Ariad*, 598 F.3d at 1351. Here, as will be discussed in further detail, the '480 Patent and the earlier filed applications in the patent chain comply with the written description requirement of § 112(a).

The Board determined in the institution decision that the '422 application, from which the '773 application (issued as the '480 patent) continued, does not provide sufficient written description support for the recited "providing a private key in a memory of said electronic device, wherein said memory being a secure part of a Basic In Out System *or any other secure location in said electronic device*" in the '480 patent (emphasis added).(Dec, at 13)

The Board refers to Figure 1 and detailed description of the '422 application describing the installation method as (Dec, at 12-13)

A new device relates to any electronic device containing a Basic In Out System ("BIOS", "Boot agent" etc.) with any associated secure storage/memory location, e.g. a computer, server, printer, personal digital assistant, mobile telephone, which is being manufactured, or has been manufactured, but has not yet been delivered to an end-ser, whereas an existing device relates to any electronic

device containing a Basic In Out System which has been delivered to an end-user, and may or may not have been used by the end-user. The Basic In Out System is only referred to as a system for accessing a memory location in a memory that is direct or indirectly connected to the electronic device. However, as described hereinafter, the method according to the present invention may employ such a BIOS system to securely store certain digital data and/or such a BIOS system may be provided with an encryption system. A secure storage location and/or an encryption system are not essential to the BIOS system with respect to the present invention. Ex. 1014, at 6-7.

In the passage "a secure storage location (...) is not essential to the BIOS system with respect to the present invention" the average person skilled in the art will read no more than that the secure location in the device in which the authentication table or data is stored can also lie outside the BIOS. The average person skilled in the art will not unambiguously read that **it is not required that the electronic device is equipped with a BIOS, nor that every storage location in the equipment is a secure storage location**. (Emphasis added) Ex. 1015, 5.20.

The Board quotes the Hague court's determination that a person having ordinary skill in the art would have understood that "the invention requires equipment that has a Basic Input/Output System such as a 'BIOS' or "Boot agent'" (Ex. 1015, 5.20), wherein the '422 application "does not describe any alternative

for shutting off the access of the operating system/user of the electronic device to storage locations other than by means of the BIOS." *Id.* at 5.31. That is, "it is clear that the average person skilled in the art will directly and unambiguously deduce from the original application that the security of transactions is achieved by storing the authentication data in a storage location within the BIOS or shield from the operating system by the BIOS," wherein adding the feature that "the authentication data can also be stored in 'any other secure location' in the electronic device" entailed adding "impermissible matter." *Id*. at 5.32. Dec, 13-14.

The Board then contends "[s]ignificant to our determination is the lack of any description of the structure of "any alternative for shutting off the access of the operating system/user of the electronic device to storage locations other than by means of the BIOS," as the Hague court points out. Ex. 1015, 5.31. We note Patent Owner does not provide any evidence to support such alternative means". Dec, at 14. Thus "a person having ordinary skill in the art would not have understood that the patentee was in possession of an invention where a private key provided in a memory of an electronic device being a secure part of any secure location in said electronic device other than a Basic In Out System". Dec, at 14.

The Hague court's determination also concludes that "[a]ll of the foregoing makes it clear that the average person skilled in the art will directly and unambiguously deduce from the original application that the security of

transactions is achieved by storing the authentication data in a storage location within the BIOS or shielded from the operating system by the BIOS. However, this restriction resulting from the original application is not claimed in feature 1.5. **The embodiments with an inextricable link to the BIOS are thus generalised to a more generic embodiment without that link.** Thus, in the independent claim 1 that has been granted (i.e. insofar as it expresses that the authentication data **can also be stored in 'any other secure location' in the electronic device**), impermissible matter has been added. Ex. 1015, 5.32.

The Board's preliminary decision and Hague court's provisional determination are based on an incorrect interpretation of the term "any other secure location" as it is not linked to the BIOS.

The '422 application provides sufficient written description support for the linking between "any other secure location" and the BIOS. For example, the specification of the '422 application includes the following paragraphs:

The encoded authentication table is stored in a secure part of the BIOS, where it may only be retrieved by the BIOS and not by the operating system. This secure part of the BIOS may also be **any other secure location** in the device. For instance, it may be a separate part of a hard drive of a computer, **which part may not be accessible to the operating system, but only to the BIOS** and/or authentication software. (Emphasis added) Ex. 1014, at 10. See Ex. 2012, ¶¶ 53-54.

Figure 3 as shown below also disclose a secure storage location outside the BIOS, which can only exchange data with the BIOS and/or the authentication software stored in the BIOS.

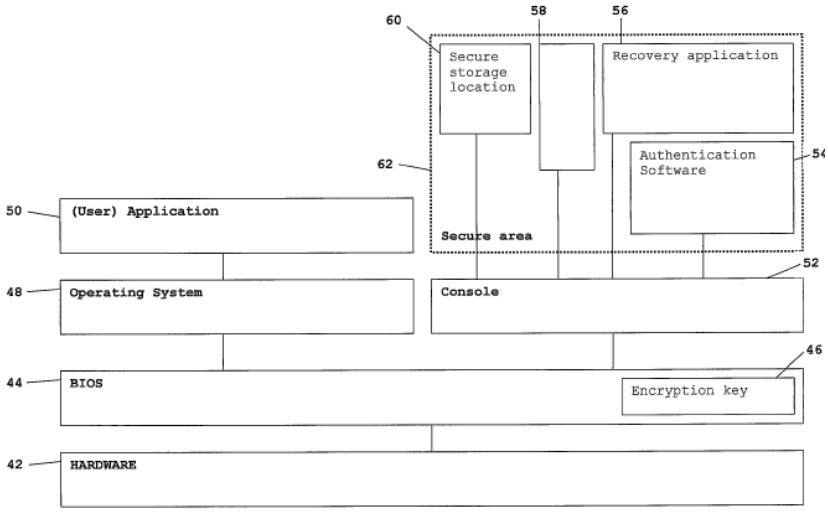


FIG. 3

In or behind the console 52, there is a **secure area 62 only accessible to the BIOS.** The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48. (emphasis added) Ex. 1014, at 14,12-15.

An authentication table may be securely stored in the **secure storage location 60**. In such a part of the computer, commonly seen as a part of the BIOS 44, the authentication table is unreachable for the operating system 48 and thus for the user. Ex. 1014, at 14, 23-26.

Therefore, the authentication software 54 may be installed in an application environment in the secure area 62 such that it may obtain the authentication table without passing through an unsecured part of the device. Ex. 1014, at 14, 31-34.

The average person skilled in the art will directly and unambiguously deduce from the ’422 application that “secure part”, “secure location”, “secure storage location” may only be retrieved by the BIOS and not by the operating system whether or not they are in the BIOS. “any other secure location” is a “secure storage location” not in the BIOS but shielded from the operating system by the BIOS.

The key characteristic of “secure part of a Base In Out System” and “any other secure location in the device” is not “location” but “secure”. Even if the private key is stored in the BIOS, if it is not in the “secure part”, it still can be accessed by the operating system. For example, the ’422 application describes “[t]he authentication software and the bit string are put in **a part of the BIOS which is accessible to the operating system** according to cell 28A. Next, according to cell 28B, the device needs to reboot to store the authentication software and the bit string in **a secure part of the BIOS which is not accessible for the operating**. (emphasis added) Ex. 1014, at 12, 6-12.

Now that the Board’s preliminary decision and Hague court’s provisional determination have no objection that “secure part of a Base In Out System” is supported by the written description, they should also admit that “any other secure location”, since both “secure part of a Base In Out System” and “any other secure location in said electronic device” can shut off the access of the operating

system/user of the electronic device by the BIOS. "any secure location" in the electronic device is not "any other location" in the electronic device because there is a linking between "any other secure location" and the BIOS, but "any other location" cannot shut off the access of the operating system/user of the electronic device, there is no link between "any other location" and the BIOS.

Since the written description of '422 application support that "any other secure location" can shut off the access of the operating system/user of the electronic device by the BIOS, '422 application does not need to describe any alternative for shutting off the access of the operating system/user of the electronic device to storage locations other than by means of the BIOS to support "any other secure location in said electronic device".

Therefore, the '422 application contains an adequate written description of "providing a private key in a memory of said electronic device, wherein said memory being a secure part of a Basic In Out System or any other secure location in said electronic device," as required by claim 1.

The Board hold that the elements "private key in a memory of said electronic device . . . inaccessible to a user of said electronic device," "encrypted when the private key is environment inaccessible to said user and to any user-operated software" and "decrypted in a secure processing environment inaccessible to said user and to any user-operated software" as recited in claim 1 of the '480

patent, lack adequate written description support in the '773 application. Dec. 15-21.

Patent Owner respectfully submits that each limitation is fully supported by the written description of the '773 application and that one of ordinary skill in the art would readily recognize the claim limitations based on the disclosure of the '773 application.

The limitation **"[the] private key is inaccessible to a user of said electronic device".**

Petitioner admits that the specification includes description of a private key, Petition at 14. The '773 application also discloses "[t]he authentication software preferably has its own unique serial number, software ID and/or private encryption key." Ex. 1001 5:39–40. A POSITA would recognize that since the unique serial number, software ID of the authentication software is private key, the private key forms an integral part of the authentication software component and/or is embedded therein. This is especially evident because the '773 application does not provide that the authentication software would have access to its own private key by requesting such access from any other software component. Additionally, the '773 application does not provide description the authentication software's private key would be subsequently retrieved after a request and handed over to the authentication software from any source or memory when need to perform unique

digital signing. Thus, a POSITA would recognize that the private key is with the authentication software. See Ex. 2012, ¶¶ 52-58.

Petitioner contends that, although the '480 patent explains that secure area 62 is "[i]n or behind the console 52" and is "only accessible to the BIOS [44]," "the '480 Patent fails to describe where the private key is stored in the electronic device and certainly does not teach that the secure area 62 stores the private key." Pet. 17. According to Petitioner, although the '480 patent teaches an "authentication software for signing a digital signature [which] uses a private key" that is "preferably stored in a secure part of the BIOS 44" and is protected "in the secure storage location 60," this disclosure "describes the storage of the ***digital signing algorithm and not of the private key***." *Id.* at 18 (citing Ex. 1001, 9:27–34, 13:31–44; Ex. 1003

¶ 173). Petitioner contends that "the '480 Patent only reveals that the private key is '***registered at a third party***, for example the authentication software provider ***which supplied said private key***.'" *Id.* at 18 (citing Ex. 1001, 13:31–44).

Petitioner then contends that, since there is no teaching in the '480 patent of where the private key is stored or maintained in the user's electronic device "or how the private key may be protected from the user when provided by the third party," the '480 patent "fails to provide written description in support of the private

key being inaccessible to a user of the electronic device." *Id.* at 19 (citing 1003 ¶¶ 175–176).

As a result of recognizing that the private key can be an integral part of the authentication software because the '773 application clearly states this, a POSITA would recognize that disclosure relating to the authentication software also includes the private key embedded therein. Ex. 1001 5:39–40; see Ex. 2012, ¶¶ 5, 46, 53, 55-56.

FIG.3 of the '480 Patent clearly discloses authentication software 54 installed and stored in secure area 62. Ex. 1001, FIG. 3. Since the authentication software 54 includes the private key contained or embedded therein, the private key must be provided in the Secure Area 62. See Ex. 2012, ¶¶ 56-57.

The '773 application depicts that "[i]n or behind the console 52, there is a secure area 62 only accessible to the BIOS. The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48. That means that the operating system 48 does not know that the applications and data in the storage locations are present and maybe even running in a separate environment. Ex., 1001,8:62-9:1. Since the private key is located in the secure area 62, only accessible to the BIOS, the private key is inaccessible to a user of said electronic device.

Although the ’773 application reveals that the private key is ‘registered at a third party, for example the authentication software provider which supplied said private key. Ex. 1001, 13:31–44, Since before an electronic transaction by a user, “the BIOS manufacturer installs authentication software in a BIOS”. Ex., 1001,4:47-48. Which means when the authentication software is provide by the third party during the installation of authentication software in a BIOS, the system of the present invention still does not exit, a user cannot access the nonexistent system, he cannot access the authentication software (with the private key), After installation of the authentication software (with the private key), the private key locates in the BIOS, the user yet cannot access the private key. See e.g., Ex 2012, ¶¶ 57-60.

The board’s preliminary determination held that it is not persuasive of Patent Owner’s contention that a person of ordinary skill in the art would have recognized that authentication software has a “private key” that forms an “integral part of the authentication software component and/or is embedded therein” (Prelim. Resp. 13 (quoting Ex. 1001, 5:39–40)), in view of the cited section in the Detailed Description describing the authentication software as comprising the unique serial number, the unique software ID and/or a private encryption key, but only the unique serial number or the unique software ID is embedded therein. (Dec, 20, (quoting Ex. 1001, 5:39–49)).

Patent Owner respectively submits that the ’773 application discloses in an embodiment that the authentication software may be provided with a system for signing a digital signature according to the present invention using at least a part of a software identification number (software ID) or any other application identifying character string, hereinafter referred to as a private key. The private key is registered at a third party, for example the authentication software provider which supplied said private key. The private key may be used to uniquely sign the digital signature and append the signed digital signature to the digital signature, which is sent to the second transaction party. The second transaction party is not capable of generating the signed digital signature without the private key. The second transaction party only stores the signed digital signature. Ex., 1001, 13:31-44.

Since both the authentication software and the private key are used to sign the digital signature, the ’773 application does not provide description the authentication software’s private key would be subsequently retrieved after a request and handed over to the authentication software from any source or memory when need to perform unique digital signing. Thus, a POSITA would recognize that the private key is with the authentication software. See e.g., Ex. 2012, ¶¶ 55-56.

A POSITA would recognize that the ’773 application contains an adequate written description of “[the] private key is inaccessible to a user of said electronic device” as required by claim 1.

“wherein the private key is encrypted when the private key is stored in said memory”

As discussed above, the authentication software has the private key. Ex. 1001, 5:39–40. Thus, if the authentication software is encrypted when the authentication software is stored in the memory, the private key must also be encrypted when the authentication software (with the private key) is stored in the memory.

The second preferred embodiment of the ’773 application discloses “the private key is encrypted when the private key is stored in said memory”.

In the second preferred embodiment, the installation method is performed on an existing computer device. The existing device does not have authentication software installed in its BIOS, nor does it have an encrypted bit string stored in its BIOS at the time a digital signature. Ex., 1001,6:61-48

The installation of the authentication software includes the steps:

According to cell 24, a device having a BIOS without an authentication table and/or authentication software initiates an on-line transaction with a third party application. The third party application requests a digital signature from the

existing device, which does not have the authentication software and the authentication table. The third party application therefore requests that the device installs the authentication software first before continuing the transaction. Ex., 1001,7:25-32.

The TTP generates and stores a bit string according to cells 26A-26E, …… Further, the bit string is sent to the requesting device together with the authentication software, both encrypted. Ex., 1001,7:33-42. Thus, in this step, the authentication software is encrypted in order to be protected when transmitted from the TTP to the requesting device. Since the authentication software has the private key, the private key is encrypted accordingly. See Ex. 2012 ¶¶ 58-66.

The authentication software and the bit string are put in a part of the BIOS which is accessible to the operating system according to cell 28A. Next, according to cell 28B, the device needs to reboot to store the authentication software and the bit string in a secure part of the BIOS which is not accessible for the operating system. Ex., 1001,7:43-48. In this step the encrypted authentication software with its private key is stored in a secure part of the BIOS which is not accessible for the operating system. See Ex. 2012, ¶¶ 46-47.

A POSITA would recognize that the second preferred embodiment in the ’773 application contains an adequate written description of “wherein the private key is encrypted when the private key is stored in said memory”.

"wherein the private key is decrypted in a secure processing environment inaccessible to said user and to any user-operated software"

As discussed above, the second preferred embodiment of the '773 application discloses that the authentication software with its private key is stored in a secure part of the BIOS which is not accessible for the operating system.

The second preferred embodiment of the '773 application also discloses "the private key is decrypted in a secure processing environment inaccessible to said user and to any user-operated software".

After the encrypted authentication software is stored in a secure part of the BIOS, the installation of the authentication software further includes the steps:

At this point, the method continues as the method illustrated in FIG. 1 from cell 10 and further. According to cell 30 the device having newly stored authentication software and a bit string in its BIOS contacts the TTP again and requests a decrypt key, for example the data string, according to cell 32. Ex., 1001,7:49-54.

According to cell 34, the decrypt key is used to decrypt the authentication software and to decrypt the bit string and generate the corresponding authentication table. Next, according to cell 36, the authentication software verifies the authentication table and requests BIOS-specific data from the BIOS to encrypt the authentication table again. Ex., 1001,7:55-60. In this step, the authentication

software is decrypted in order to generate the corresponding authentication table. Since the authentication software has the private key, the private key is decrypted accordingly.

The ’773 application further discloses:

To prevent that the authentication data becomes accessible to a user, the authentication data should only be decrypted in a secure processing environment such as a 'Ring Zero' processing environment, which is embedded in a processing unit of commonly used personal computers. Such a secure processing environment may only be accessible to selected software applications, preferably not to user-operated software applications. Ex. 1001, 9:42–49.

A POSITA would readily acknowledge that Ring 0 processing, also known as kernel mode processing, is a particular execution mode of the processor that is inaccessible by the user and the user-operated software. Thus, the ’773 application discloses decryption in a secure environment inaccessible to user-operated software applications.

A POSITA would recognize that the second preferred embodiment in the ’773 application contains an adequate written description of “wherein the private key is decrypted in a secure processing environment inaccessible to said user and to any user-operated software”.

Therefore, the ’422 application contains an adequate written description of “providing a private key in a memory of said electronic device, wherein said memory being a secure part of a Basic In Out System or any other secure location in said electronic device,” as required by claim 1. The ’773 application contains an adequate written description of “private key in a memory of said electronic device . . . inaccessible to a user of said electronic device,” “encrypted when the private key is environment inaccessible to said user and to any user-operated software” and “decrypted in a secure processing environment inaccessible to said user and to any user-operated software” as recited in claim 1 of the ’480 patent. The earliest filing date for this subject matter is the filing date of the ’422 application when the subject matter was first introduced, i.e., June 14, 2004. As such, Patent Owner submits that the ’480 patent is not eligible for post-grant review. (See Ex. 2012, ¶¶ 46-51.)

V. THE ’480 PATENT SATISFIES THE WRITTEN DESCRIPTION REQUIREMENT UNDER 35 U.S.C. §112

Petitioner contends that the limitations “providing a private key in a memory of said electronic device, wherein said memory being a secure part of a Basic In Out System or any other secure location in said electronic device,” “encrypted when the private key is stored in said memory,” and “decrypted in a secure

processing environment inaccessible to said user and to any user-operated software" lack written description support. Pet. 16–24.

As discussed above, in connection with the analysis of whether the '480 patent is eligible for post-grant review, the '422 application (from which the '773 application was filed as a continuation) and the '773 application contain adequate written description of the limitations, the '480 patent satisfy the written description requirement under 35 U.S.C. §112. See Ex. 2012, ¶¶ 52-71.

Out of an abundance of caution, Patent Owner notes the following with regard to claim 1 of the '480 patent:

Support for the claim language, "1. A method for performing an electronic transaction between a first transaction party and a second transaction party using an electronic device operated by the first transaction party," can be found in the '480 patent at Col. 1, lines 57-60: "At least this object is achieved in the present invention by a method for performing an electronic transaction between a first transaction party and a second transaction party using an electronic device operated by the first transaction party."

Support for the claim language, "the electronic device having an operating system creating a run-time environment for user applications, the method

comprising:" can be found in the '480 patent at Col. 8, lines 33-34: "The operating system 48 creates a run-time environment for any user applications."

Support for the claim language "providing a private key in a memory of said electronic device, said memory being a secure part of a Basic In Out System or any other secure location in said electronic device," can be found at Col. 5, lines 39-40: "The authentication software preferably has its own unique serial number, software ID **and/or private encryption key**. The authentication software is installed, thus stored in the electronic device. Cf. FIG 3. with provided authentication software 54. *Therefore, the character string (=private key) is provided in a memory of secure area 62.*

Additionally, support for the claim language "providing a private key in a memory of said electronic device, said memory being a secure part of a Basic In Out System or any other secure location in said electronic device," can be found at Col. 13, lines 31-36: "In a further embodiment, the authentication software may be provided with a system for signing a digital signature according to the present invention using at least a part of a software identification number (software ID) or any other application identifying character string, **hereinafter referred to as a private key**." (Emphasis added).

Further support for the claim language "providing a private key in a memory of said electronic device, said memory being a secure part of a Basic In Out System or any other secure location in said electronic device," can be found at Col. 3, line 64 – Col.4, line 3: "However, as is described hereinafter, the method according to the present invention **may employ** such a BIOS system to securely store certain digital data and/or such a BIOS system may be provided with an encryption system. **A secure storage location and/or an encryption system are not essential to the BIOS system with respect to the present invention**." (Emphasis added).

Support is also found Col. 6, lines 48-53, which recites, "This secure part of the BIOS may also be **any other, secure location in the device.** For instance, it may be a separate part of a hard drive of a computer, which part may not be accessible to the operating system, but only to the BIOS **and/or authentication software**." (Emphasis added).

Even more support is located at Col. 7, lines 2-8, which explain "Thus, compared to a new computer device, further steps are necessary in the above-described installation method… in a secure memory location of the device. Such **a secure memory location** may be **a part of a secure memory** or it may be **an**

encrypted part of a non-secure memory, such as a user-accessible memory." (Emphasis added).

Significantly, support for the claim language "which private key is inaccessible to a user of said electronic device," can be found at Col. 6, lines 48-53: "This secure part of the BIOS may also be any other, secure location in the device. For instance, it may be a separate part of a hard drive of a computer, **which part may not be accessible to the operating system, but only** to the BIOS **and/or authentication software"** and at Col. 9, lines 9-11: "…the authentication table is **unreachable for the operating system 48 and thus for a user**." (Emphasis added.)

Support for the claim language, "wherein the private key is encrypted when the private key is stored in said memory, a decryption key for decrypting the private key being incorporated in the electronic device, said decryption key being inaccessible to said user, to any user-operated software and to said operating system," can be found at Col. 10, lines 23-33: "After installation of the authentication software, the authentication software obtains a master bit string (MBS) according to cell 202. The master bit string (MBS) may be a large array of characters, for example 2048 numbers. **The MBS may be embedded in the authentication software package, and may in that case not need to be obtained**

separately as indicated in FIG. 4. From said master bit string (MBS), according to cell 204, the authentication software selects a bit string. Said bit string thus comprises a number of said characters, for example 128 characters randomly selected from the MBS." (Emphasis added).

Support for the claim language, "wherein the private key is encrypted when the private key is stored in said memory, a decryption key for decrypting the private key being incorporated in the electronic device, said decryption key being inaccessible to said user, to any user-operated software and to said operating system," can also be found at Col. 11, lines 42-45: "The authentication software is also installed and stored in a user accessible memory location as indicated. The authentication software may be encrypted, or not."

Additionally, support can be found at Col. 5, lines 39-40: "The authentication software preferably has its own unique serial number, software ID **and/or private encryption key**." (Emphasis added).

Support can also be found at Col. 6., lines 39-42: "The BIOS-specific number may be a unique serial number or any ordinary encryption key incorporated in or generated by the BIOS, **for example**. The authentication software is inaccessible by the operating system/user while stored in the secure

area 62 **of the console** or while stored (encrypted) in the secure memory location. (Emphasis added).

Moreover, the ’480 description does **NOT** disclose that this selected bit string is transferred by the authentication software to any other memory. It remains **IN** the authentication software all times. In the ‘’480 patent, the authentication software is encrypted in user accessible memory, as with the third embodiment. For example, it is disclosed in Col. 7, lines 2-8 that an encrypted part of non-secure memory is **a secure memory location, as per claim 1.**

Support for the claim language, “providing authentication software in said electronic device, the private key being accessible to said authentication software, wherein authentication software is stored in said secure memory inaccessible to said operating system;” can be found at Col. 1, lines 63-66: “providing authentication software in said electronic device, the authentication data being accessible to said authentication software.” See also FIG 3, which shows Authentication software 54 stored in secure area 62 of the console.

Support for the claim language, “activating the authentication software to generate a digital signature from the private key, wherein the authentication software is run in a secure processing environment inaccessible to said operating system;” can be found in the ’480 patent at Col 1, lines 66-67: “activating the

authentication software to generate a digital signature from the authentication data." Additional support is provided at Col. 13, lines 38-41: "The private key may be used to uniquely sign the digital signature and append the signed digital signature to the digital signature, which is sent to the second transaction party."

Lastly, support for the claim language, "providing the digital signature to the second transaction party" can be found in the '480 patent at Col 1, line 67 – Col. 2, line 1 which recites, "providing the digital signature to the second transaction party" and at Col. 13, lines 38-41 which recites, "The private key may be used to uniquely sign the digital signature and append the signed digital signature to the digital signature, which is sent to the second transaction party."

Each element of claim 1 of the '480 patent thus finds adequate support in the written description of the '480 patent. The arguments supplied above apply mutatis mutandis to the other independent claims of the '480 patent.

As the authentication software is encrypted, the selected character string (=private key) contained therein is also encrypted whilst stored in the secure memory location.

Again, the description does not disclose that the authentication software transfers its private encryption key to any other memory. It thus forms an integral part of the authentication software. Nor does the '480 Patent provide that the

authentication software would have (undefined) access to its own private key by requesting such access from any other software component. Ex. 2012, ¶ 54.

VI. CLAIMS 1, 3, 6, 11, 14-19 OF THE '480 PATENT ARE PATENTABLE OVER ELLISON AND MUIR

A. Ellison

Ellison is entitled "Protecting Software Environment In Isolated Execution" and describes a method and apparatus for protecting a subset of a software environment. Ex. 1006, code (54); Abstract.

Ellison provides an isolated execution architecture as shown in Fig. 1A. (reproduced below).

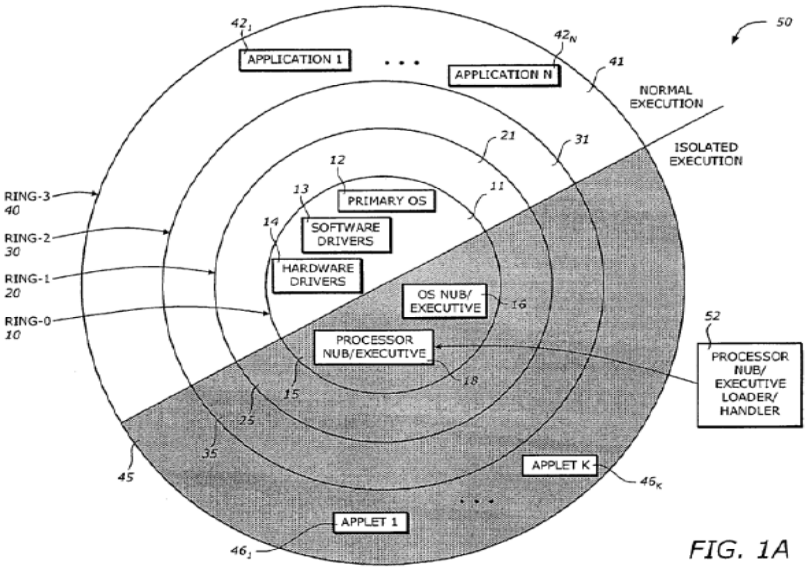


FIG. 1A

The logical operating architecture 50 shown in FIG. 1A "includes ring-0 10, ring-1 20, ring-2 30, ring-3 40, and a processor nub loader 52." Ex.1006, 2:62-3:1.

A ring is a logical division of hardware and software components that are designed to **perform dedicated tasks within the operating system**. The different ring levels are associated with different privilege levels. Ex. 1006, 2:41-61.

The logical operating architecture 50 has two modes of operation: normal execution mode and isolated execution mode. Each ring in the logical operating architecture 50 can operate in both modes. The processor nub loader 52 operates only in the isolated execution mode." Ex. 1006, 3:4-8. "Ring-0 10 includes two portions: a normal execution Ring-0 11 and an isolated execution Ring-0 15." Ex.1006, 3:8-10. "The isolated execution Ring-0 15 includes an operating system (OS) nub 16 and a processor nub 18." Ex.1006, 3:15-16.

Ellison's isolated execution architecture includes "an isolated region in the system memory" and "is protected by both the processor and chipset in the computer system." Ex.1006, 3:32-36, FIGS. 1A, 1B. **"The isolated area 70 is accessible only to elements of the operating system and processor operating in isolated execution mode."** Ex.1006, 4:8-22.

The processor nub 18 provides hardware-related services for the isolated execution. Ex.1006, 3:47-49.

The operating system nub 16 provides links to services in the primary OS 12 (e.g., the unprotected segments of the operating system), **provides page management within the isolated area**, and has the responsibility for loading ring-

3 application modules 45, including applets 46_1 to 46_N, into protected pages allocated in the isolated area. Ex.1006, 3:32-36, 4:30-37. The operating system nub 16 is also responsible for encrypting and hashing the isolated area pages before evicting the page to the ordinary memory, and for checking the page contents upon restoration of the page. Ex.1006, 3:64-4:7.

The isolated mode applets 46_1 to 46_N and their data are tamper-resistant and monitor-resistant from all software attacks from other applets, as well as from non-isolated space applications (e.g., 42_1 to 42_N), dynamic link libraries (DLLs), drivers and even the primary operating system 12. Only the processor nub 18 or the operating system nub 16 can interfere with or monitor the applet's execution. Ex.1006, 4:1-7. The isolated area 70 is accessible only to elements of the operating system and processor operating in isolated execution mode. Ex.1006, 4:19-21.

The isolated execution ring-015, including the OS nub 16 and the processor nub 18, can access to both of the isolated area 70, including the applet pages 72 and the nub pages 74. and the non-isolated area 80, including the application pages 82 and the OS pages 84. The isolated execution ring-3 45, including applets 46 to 46, can access only to the application pages 82 and the applet pages 72. The applets 46_1 to 46_K reside in the isolated area 70. Ex.1006, 4:30-37.

A computer system 100 to implement the isolated execution architecture includes a processor 110, a host bus 120, a memory controller hub (MCH) 130, a

system memory 140, an input/output controller hub (ICH) 150, a non-volatile memory, or system flash, 160, a mass storage device 170, input/output devices 175, a token bus 180, a motherboard (MB) token 182, a reader 184, and a token 186. Ex.1006, 4:40-45.

The MCH 130 provides **control and configuration of memory and input/output devices such as the system memory 140 and the ICH 150**. The MCH 130 provides interface circuits to recognize and service isolated access assertions on memory reference bus cycles, including isolated memory read and write cycles. In addition, the MCH 130 has memory range registers (e.g., base and length registers) to represent the isolated area in the system memory 140. Once configured, the MCH 130 aborts any access to the isolated area that does not have the isolated access bus mode asserted. Ex.1006, 5:57-67.

The system memory 140 stores system code and data. The system memory 140 is typically implemented with dynamic random access memory (DRAM) or static random access memory (SRAM). The system memory 140 includes the accessible physical memory 60. The accessible physical memory includes a loaded operating system 142, the isolated area 70, and an isolated control and status space 148. The loaded operating system 142 is the portion of the operating system that is loaded into the system memory 140. The isolated area 70, is the memory area that is defined by the processor 110 when operating in the isolated execution mode.

Access to the isolated area 70 is restricted and is enforced by the processor 110 and/or the MCH 130 or other chipset that integrates the isolated area functionalities. The isolated control and status space 148 is an input/output (I/O)-like, independent address space defined by the processor 110 and/or the MCH 130. The isolated control and status space 148 contains mainly the isolated execution control and status registers. The isolated control and status space 148 does not overlap any existing address space and is accessed using the isolated bus cycles. Ex.1006, 6:1-24. See Ex. 2012, ¶¶ 72-73.

B. Muir

Muir is entitled "Apparatus and Methods for Using a Virtual Smart Card," and describes an apparatus that includes a smart card interface and a virtual smart card configured to emulate a physical smart card as if a smart card is connected with the smart card interface. Ex. 1008, code (54); Abstract.

Muir's virtual smart cards 151 include a restricted memory space 170 to store a private key 167 and for performing operations such as encryption of data or generation of a digital signature using the private key. Ex. 1008, at 3, 6.

The virtual smart card 151 in the computing devices "provides software code, which upon execution, emulates a physical smart card for device operating system 150." Ex. 1008, at 7. The virtual smart card 151 includes a "restricted memory space 170 used for storing private keys 167. Restricted memory space 170

and private key 167 are normally inaccessible to operating system 125." Ex. 1008, at 7, FIG. 3. The private key 167 is stored in a restricted memory space 170 to avoid the private key 167 from being exposed to the operating system 125 or an unauthorized user, which was a problem in earlier systems. Ex. 1008, at 3, 7. The private key 167 may be used to generate a digital signature and sent to another device. Ex. 1008, at. 9. See Ex. 2012, ¶¶ 74-75.

C. Claim 1 is not obvious over Ellison and Muir

Petitioner's reliance on a proposed combination of Ellison and Muir is supported by ignoring features of claim 1 and improperly interchanging multiple words contrary to both their meaning and the respective disclosures in an attempt to mask a failure of both Ellison and Muir to disclose elements of claim 1.

The combination of Ellison and Muir cannot form a proper ground for claim 1 of the '480 Patent being unpatentable, because: (i) neither Ellison nor Muir teach or suggest said decryption key being inaccessible to said user, to any user-operated software and to said operating system, wherein said memory is inaccessible to said operating system of said electronic device; (ii) neither Ellison nor Muir teach or suggest authentication software stored in a secure area inaccessible to the operating system; and (iii) a POSITA would not have looked to Muir to cure the deficiencies of Ellison Each of these reasons alone is sufficient grounds for the failure of Ellison and Muir to teach the obviousness of Patent Owner's features.

Patent Owner notes further that the OSNK 203 of Ellison is used to decrypt the register values or to sign the software to be run in the secure processing environment—not to sign a transaction between a first and second transaction party. See Ex. 2012, ¶ 77. It is unclear, however, where the OSNK key 203 is stored or whether is it generated every time it is needed. See Ex. 2012, ¶ 78.

Ellison's OSNK 203 is generated by the key generator 240 after the device has been put in circulation. See Ex. 2012, ¶ 79.

Neither Ellison nor Muir teach or suggest authentication software is stored in said secure memory inaccessible to said operating system. See Ex. 2012, ¶ 80.

The goal of usage protector 250 is to protect the software and registry states in the secure environment, that is to protect the secure environment itself and not to protect electronic transactions by a user (first transaction party). See Ex. 2012, ¶ 81.

The execution environment of Ellison's usage protector 250 is not the same as where usage protector 250 is stored. See Ex. 2012, ¶ 82.

There is not a single instance in Ellison disclosing where the usage protector 250 is stored. See Ex. 2012, ¶ 83.

Muir's Virtual Smart Card Driver 155 is fully accessible because it is stored in one of fixed disk 225, RAM 205 or ROM 204, or other memory devices within computer system 110. See Ex. 2012, ¶ 84.

None of Muir's fixed disk 225, RAM 205 or ROM 204, or other memory devices within computer system 110 are secure memory. See Ex. 2012, ¶ 85.

Neither Ellison nor Muir teach or suggest authentication software is run in a secure processing environment inaccessible to said operating system. See Ex. 2012, ¶ 86.

Ellison's failure to disclose the storage location of usage protector 250 implies the primary OS 12 would have access to usage protector 250 for running in the normal execution environment. See Ex. 2012, ¶ 87.

Ellison's drivers 13, 14 all run in normal execution mode and any combination of Ellison and Muir would require Muir's Virtual Smart Card Driver 155, which also runs openly accessible to the operating system. See Ex. 2012, ¶ 88. Ellison fails to disclose encryption of usage protector 250. See Ex. 2012, ¶ 89.

Ellison's primary OS 12 would always have access to the usage protector 250 and could subsequently run the usage protector 250 in the normal execution environment. See Ex. 2012, ¶ 90.

Ellison cannot be combined with Muir's pointer 171 to restricted memory space 170 without also incorporating Muir's Virtual Smart Card Driver 155. 91. Indeed, combining Muir with Ellison would nullify the purpose of Ellison because any request to create a digital signature would come through the primary OS 12

operating with the driver 155 without even touching the isolated execution environment of Ellison. See Ex. 2012, ¶ 92

There is a fundamental difference between Ellison and ’480 in terms of goals of the system, which also leads to substantial differences in properties and implementation. The goal of Ellison is to create an isolated execution environment for software. Ellison excludes transactions by applications to be executed in the isolated execution environment. Ellison does not disclose any `Authentication Software' or for that matter any Cryptographic Key Storage (secure or not) in the Normal Execution Environment. Nor does Ellison disclose any secure processing in the Normal Execution Environment. Samsung acknowledges Ellison does not disclose a system or transaction between multiple parties (insert references here). See Ex. 2012, ¶¶ 93-94.

In order to protect this isolated execution environment, Ellison discloses a combination of cryptographic keys and cryptographic software including software for digital signature schemes. However, the goal of these cryptographic keys and cryptographic software is to protect the isolated environment from tampering by the operating system and the applications, while in ’480 the goal is to protect user transactions. Any system for secure transactions would also use a combination of cryptographic keys, cryptographic software, secure memory and a secure execution environment. To a POSITA, it would have not at all been obvious how to create a

system or method on how to combine all these elements properly neither in a legacy hardware system nor in a system with additional hardware to enforce isolation. See Ex. 2012, ¶ 95.

In more detail, Ellison discloses a master binding key BK0 202 that is included in processor nub 18. BK0 202 is generated a random when the processor nub 18 is first invoked, i.e., when it is first executed on the secure platform (Ellison 8:66-9:2). Processor nub 18 is stored in Non-Volatile Memory 160 (Ellison Fig. 1C). From the isolated area 80, the processor nub loader 52 copies the processor nub 18 from the system flash memory (e.g., the processor nub code 18 in non-volatile memory 160) into the isolated area 70, verifies and logs its integrity, and manages asymmetric key used to protect the nub's secrets. First, from Ellison Fig. 1B it is clear that area 80 is a non-isolated area rather than isolated area[1]; probably isolated area 70 is intended, but then the question arises how the processor nub loader 52 is loaded securely in this secure area; if a non-isolated area is intended, the question arises how this processor nub loader is protected. Second, Ellison does not at all disclose where these asymmetric keys to protect the nub's keys would be stored. Third, it is manifestly unclear how master binding key BK0 202 is protected, as the Non-Volatile Memory 160 is not a secure memory. From

[1] This error is also copied on page 16 of the PTAB decision Granting Institution of *PGR*.

the master binding key BK0 202 and some data derived from the OS nub 6, the key OSNK 203 is computed by the key generator 240 after the device has been put in circulation. See Ex. 2012, ¶ 96.

Ellison discloses that OSNK 203 is input to the usage protector 250. OSNK 203 can be used to decrypt the register values or to decrypt a private key 204 that is used to digitally sign[2] the software to be run in the secure processing environment. However, the usage protector is not used to sign a transaction between a first and second transaction party. See Ex. 2012, ¶ 97.

There is also confusion where Ellison stores the (protected) private key 204. Ellison discloses that in one embodiment the protected private key 204 can be stored in the multiple key storage 164 of the non-volatile flash memory 160 (Ellison 8:55-57). While Figure 1C shows many details, it does not show such a key storage in non-volatile flash memory 160. It would seem appropriate to store said key in cryptographic key storage 155, but Ellison does not teach or suggest this, nor indeed does it specify how the access to this private key is restricted when it is stored in non-volatile memory. See Ex. 2012, ¶ 98.

The execution environment of Ellison's usage protector 250 is not the same as where usage protector 250 is stored. There is not a single instance in Ellison disclosing where the usage protector 250 is stored; the usage protector is not shown

[2] Ellison uses the term encrypt for digital signature computation.

in Figure 1C, which would be the natural place to illustrate this. Ellison's failure to disclose the storage location of usage protector 250 implies the primary OS 12 would have access to usage protector 250 and would be able to run it in the normal execution environment. See Ex. 2012, ¶ 99.

Muir discloses only one CPU mode. The calculation of the digital signature ("cryptographic calculation") and the sending of the result of the calculation (the actual digital signature) would be performed by Virtual Smart Card Driver 155 which is stored in main memory and executed in regular CPU mode (which is the only mode available, as there is no secure processing environment). See Ex. 2012, ¶ 100.

Muir discloses only secure memory ("restricted memory space") for the authentication key (for a more detailed discussion cf. infra). It does not disclose a secure memory for authentication software, nor does it disclose a secure processing environment in which to run the authentication software. All cryptographic computations are performed in the Virtual Smart Card Driver 155. See Ex. 2012, ¶ 101.

Muir's Virtual Smart Card Driver 155 is a software component (code) fully accessible by the OS 125 because it is stored in one of fixed disk 225, RAM 205 or ROM 204, or other memory devices within computer system 110 (thus in unprotected memory). The Virtual Smart Card Driver 155 performs cryptographic

calculations and sends the calculated outcome to the device Operating System 125. This implies that malware in the operating system or the operating system itself would be able to send requests to generate digital signatures from the virtual smart card without the user noticing that said signatures have been requested or generated. This is a fundamental difference with '480. See Ex. 2012, ¶ 102.

None of Muir's fixed disk 225, RAM 205 or ROM 204, or other memory devices within computer system 110 are secure memory. Two methods are disclosed to secure this memory to store a key: mark the entries as bad in the FAT (File Allocation Table) or use memory pointers. Both methods are questionable and unlike '480 they do not protect against powerful attacks at the level of the operating system or attacks by malware. See Ex. 2012, ¶ 103.

Marking entries in the FAT as bad implies that the OS will not read these entries. However, it would be trivial to make a modification to the OS or to insert malware that would read the values. In terms of protection, it can be compared to writing on a locker a note stating, "do not open this locker", while a secure storage would correspond to using a padlock to protect the locker. To the contrary, by marking specific sectors as bad, it would be easier for attackers to find the memory locations where the secret keys are stored (similar to how lockers marked with a sign reading, "do not open this locker" would draw attention). See Ex. 2012, ¶ 104.

Using specific pointer ranges to limit attacks is not an effective way to create secure memory. Muir discloses that the address pointer 171 is a regular part of the Driver code 155. As there is no additional hardware protection of these pointer mechanism, it would be very easy to modify the operating system or to write malware that bypasses this protection –it is comparable to declaring a series of lockers out of reach rather than locking them with padlocks. See Ex. 2012, ¶ 105.

The combination of Ellison and Muir (EMC) will not teach or suggest "providing a private key in a memory of said electronic device, said memory being a secure part of a Basic In Out System (BIOS) or any other secure location in said electronic device, which private key is inaccessible to a user of said electronic device, where the private key is encrypted when the private key is stored in said memory." Ellison discloses a private key 204 for the protection of the isolated environment itself, but not for the protection of digital transactions involving two parties. Petitioner then contends that Ellison's OS nub 16 "retrieves the encrypted private key 204 and decrypts it for use in the isolated environment." Thus, Petitioner contends that "Ellison and Muir render obvious the private key 204 being encrypted when stored in a secure memory (e.g., memory 160 or Muir's restricted memory space 170) and obtained therefrom by OS nub 16." There are several problems with this statement: first, Ellison stores the private key 204 (long term) in encrypted form in insecure memory (non-volatile memory 160) and

decrypts it for use inside the secure environment. Second, it is fully unclear how OS nub 16 would be able to access Muir's restricted memory space 170, as this space is only accessible to the Virtual Smart Card Driver 155 that runs in normal mode and not to the regular OS 12 in Ellison nor to the OS nub 6 in Ellison. See Ex. 2012, ¶ 106.

The combination of Ellison and Muir (EMC) will not teach or suggest "authentication software [..] stored in said secure memory inaccessible to said operating system" for the following reason: Ellison does not teach or suggest authentication software for securing user transactions, but only authentication software to authenticate the code of the secure processing environment (processor nub 18). This is an essential difference. Even if abstraction is made of this, the software in Ellison performs is the usage protector 250, but there is not a single instance in Ellison disclosing where the usage protector 250 is stored. Drivers 13, 14 in Ellison are not stored in secure mode. Muir does not teach or suggest that the Virtual Smart Card Driver 155 is stored in secure memory. As a consequence, the authentication software in any combination will be accessible to the operating system. For a POSITA, it would be not at all obvious how to achieve from EMC "authentication software [..] stored in said secure memory *inaccessible to said operating system.*" See Ex. 2012, ¶ 107.

Neither Ellison nor Muir teach or suggest authentication software is run in a secure processing environment inaccessible to said operating system. In Ellison drivers 13, 14 all run in normal execution mode and not in isolated execution mode (Ellison Fig. 1A). Ellison's primary OS 12 would always have access to the usage protector 250 and could subsequently run the usage protector 250 in the normal execution environment. Ellison cannot be combined with Muir's pointer 171 to restricted memory space 170 without also incorporating Muir's Virtual Smart Card Driver 155. Any combination of Ellison and Muir would require Muir's Virtual Smart Card Driver 155, which also runs openly accessible to the operating system. As a consequence, the authentication software in any combination will be accessible to the operating system and any request to create a digital signature would come through the primary OS 12 operating with the driver 155 without even touching the isolated execution environment of Ellison. For a POSITA, it would definitely not be clear how to achieve from EMC "authentication software […] run in a secure processing environment inaccessible to said operating system". See Ex. 2012, ¶ 108.

Muir discloses applications running in the normal CPU mode; when these applications need to authenticate transaction data, they call a smart card API and this call is interrupted by the Virtual Smart Card driver 155 that runs in the main CPU mode. If the same application is run in normal mode in Ellison, there would

be no benefit from Ellison's secure execution environment and no access to the cryptographic keys. If on the other hand, if (part of) the application would run in the secure execution environment, it is not clear at all how Muir's Virtual Smart Card driver 155 would be addressed (that needs to run in normal mode as explained above) and how it would be able to address secure locations in memory as disclosed in Muir. Because this incompatibility, the only option would be to redesign a secure transaction system from scratch which is well beyond the skills of a POSITA. See Ex. 2012, ¶ 109.

About EMC, Dr. Black states that "The combination would have been obvious, predictable, and a POSITA would have had a reasonable expectation of success in implementing the combination because both Ellison and Muir are related to providing improved security for transactions involving keys using cryptographic techniques." However, Ellison does not teach how to secure transactions but only how to create a secure execution environment. While it is correct that both Ellison and Muir use cryptographic algorithms and keys in a complex processing environment, every expert in the field will agree that developing a system for secure transactions based on general purpose computers with hardware extensions is very delicate and subtle and extremely prone to errors. Patent Owner does not agree that a POSITA would have been capable of combining these two systems in an effective way, the more so because Ellison

discloses a very complex design with many options that are very hard to understand (which by itself would make it extremely challenging for a POSITA to produce a secure solution) and because Muir and Ellison are inherently incompatible as explained above. See Ex. 2012, ¶ 110.

Al-Salqan does not overcome the deficiencies described in the Patent Owner's Response relating to the combination of Ellison and Muir. See Ex. 2012, ¶ 111.

Al-Salqan teaches to twice encrypt a key with a very specific goal, namely, to create a key recovery file. The creation of such a file increases convenience at the cost of a reduction in security: now a second party is also capable of recovering a key. This belies the statement in Black 79 that "by using two encryption layers, […] the private key generated based on Ellison and Al-Salqan's teachings would have additional layers of protection." In '480 multiple encryption (at least twice) is implemented to ensure that transactions are more secure, which is a completely opposite goal. See Ex. 2012, ¶ 112.

In order to achieve the key recovery goal, Al-Aqsan requires that "the twice encrypted first key can only be decrypted by a second party to yield the once encrypted first key; and wherein the party and the second party are different parties" (Al-Aqsan, claim 1, but this condition applies to all claims). The requirement of Al-Aqsan requires that the first key is encrypted by the first party

but can only be decrypted by a second party necessarily imposes that public key encryption is used for the second encryption. This condition is added in claims 9, 19 and 28. As an expert in the field, I do not see how it would be possible to achieve the goal in claims 1, 12 and 21 of Al-Aqsan without public-key encryption for the second encryption. SAMSUNG 1007 and Black both add the same constraint when discussing Al-Aqsan. See Ex. 2012, ¶ 113.

With respect to Claim 9 in '480, it is important to note that there is a substantial difference in nature and entropy of data when comparing private data disclosed in Al-Saqan (mother's maiden name and social security number) with personal identifying numbers, PIN, and a template associated with a fingerprint of the user). In the latter case some of the information that is chosen by the user, information that is not easy to infer and information with higher entropy; all of these and any combination of these offer a substantial improvement of security over Al-Aqsan and are thus not the same. See Ex. 2012, ¶ 114-117.

i. Neither Ellison nor Muir teach or suggest said decryption key being inaccessible to said user, to any user-operated software and to said operating system, wherein said memory is inaccessible to said operating system of said electronic device.

Regarding the limitation "the electronic device having an operating system creating a run-time environment for user applications" of claim 1, Petitioner provides annotated Figures 1A–1B of Ellison, Pet., 53,54. reproduced below:

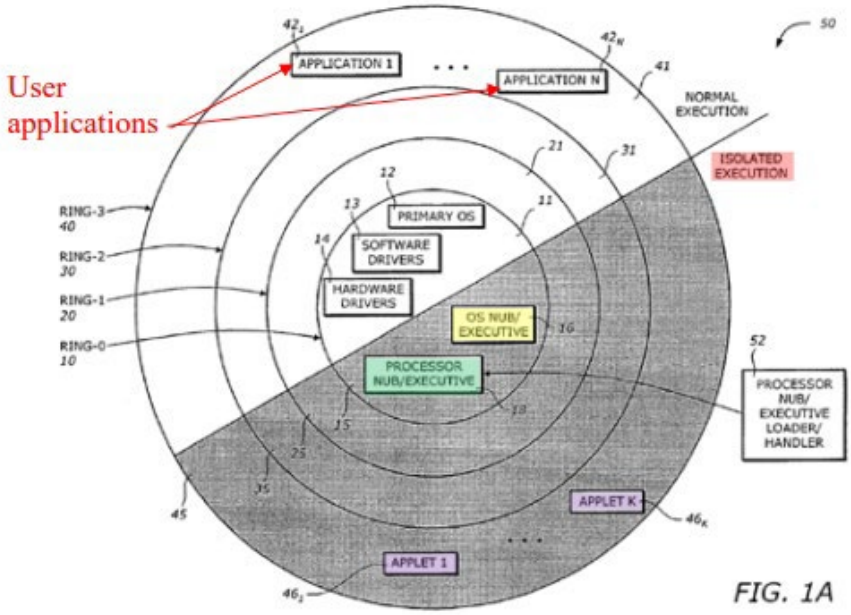


FIG. 1A

SAMSUNG-1006, FIG. 1A

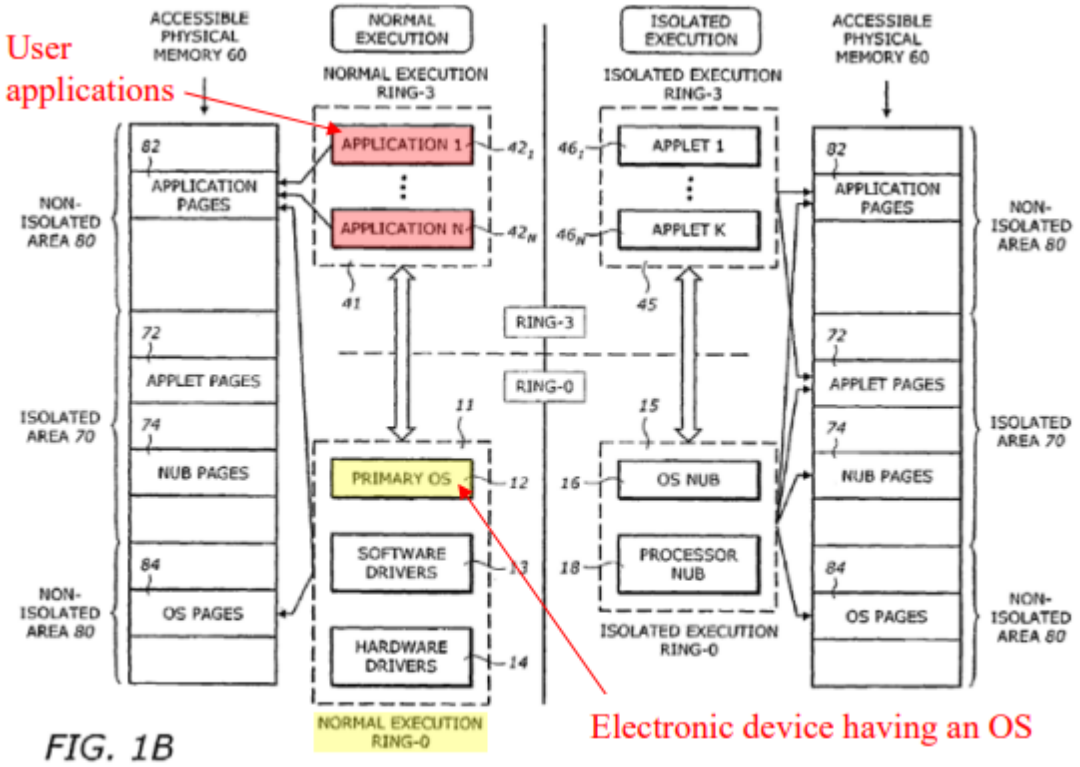


FIG. 1B

SAMSUNG-1006, FIG. 1B

As shown in annotated Figure 1A, logical operating architecture 50 has two modes of operation: normal execution mode (white) and isolated execution mode (gray). According to Petitioner, Ellison depicts **"a primary operating system (OS) 12 that creates a run-time environment for user applications, such as applications 42_1-42_N which can be executed in the normal execution mode and are isolated from components of the isolated execution rings."** Pet. 55 (citing Ex. 1006, 3:9–61, 4:1–29, 2:57–61).

As for the limitation, "said decryption key being inaccessible to said user, to any user-operated software and to said operating system", Petitioner contends that

"OSNK 203 ('decryption key') is generated and used within the **isolated environment region, which is not accessible to a user, any user-operated software, and the operating system (OS) 12** of the electronic device in the normal execution mode." Pet. at 64 (citing Ex. 1006, 8:25–65, 11:1–12:26, 9:1–14, Figs. 2, 3C, 3D).

It is clear that Petitioner wrongly equates the "primary operating system 12" of Ellison with "operating system" in the '480 Patent. Correspondingly, Petitioner misapprehends that the "isolated environment region" is not accessible to the "operating system".

An ordinary meaning of the term "operating system", as known in the art, is defined in Dictionary of Computer and Internet Terms as "a program that controls a computer and makes it possible for users to enter and run their own programs". Ex. 2009, at 343. Andrew S. Tanenbaum describes the term "operating system" in the book "Modern Operating Systems" as

A modern computer consists of one or more processors, some main memory, disks, printers, a keyboard, a mouse, a display, network interfaces, and various other input/output devices. All in all, a complex system. If every application programmer had to understand how all these things work in detail, no code would ever get written. Furthermore, managing all these components and using them optimally is an exceedingly challenging job. For this reason, computers are

equipped with a layer of software called the **operating system**, whose job is to provide **user programs** with a better, simpler, cleaner, model of the computer and to handle managing all the resources just mentioned. Ex. 2010, at 1.

According to the ordinary meaning of the "operating system", in a computer system, the "operating system" help user programs to manage resource.

The specification of the '480 Patent describe "operating system" as:

As shown in Fig. 3 of the '480 Patent (reproduced below), [t]he operating system 48 creates a run-time environment for any user applications. Therefore, it may be stated that the operating system 48 is an interface between a user and the hardware 42. A user may instruct the operating system 48 to run an application 50. If the application 50 needs data that are stored on the hard drive, the application 50 requests the data from the operating system 48. The operating system 48 in turn requests the data through the BIOS 44 from the hardware 42, i.e. the hard drive. The data coming from the hardware 42 pass through the BIOS 44 and the operating system 48 before they arrive at the requesting application 50. Ex. 1001,8:33-43.

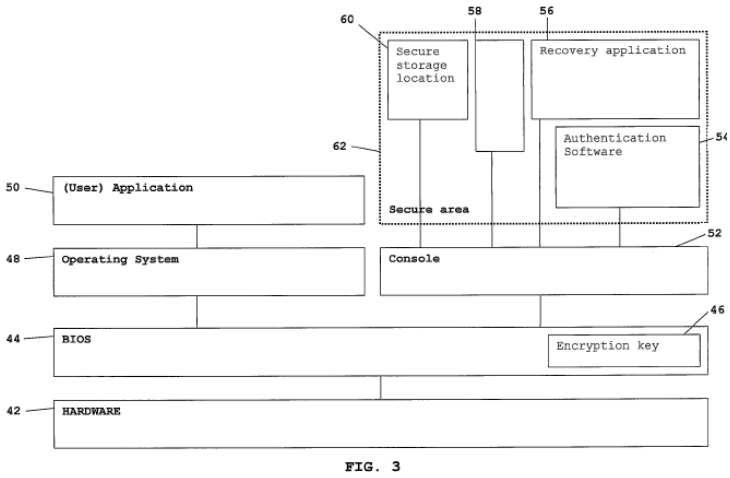


FIG. 3

According to the specification of the ’480 Patent and the ordinary meaning of the “operating system”, in a computer system, the “operating system” help user programs to manage resource. Any program supported by the “operating system” can be referred to as a user program **(**user application).

Ellison does not define “applet” in the specification, an ordinary meaning of the term “applet”, as known in the art, is defined in “Dictionary of Computer and Internet Terms” as “(1) a small application program that is inexpensive and designed to do a small, specific job. Most operating systems come with several applets, such as a calculator, a calendar, and a note editor. (2). an application program that is downloaded automatically through a World Wide Web browser and executed on the recipient’s machine. Applets are normally written in Java.” Ex. 2009, at 25.

Petitioner admits that Applications 42_1-42_N are configured to operate in Ring-3, which "typically encompasses users or applications level and has the least privilege," and would therefore be understood by a POSITA as encompassing user applications. Pet., 55 (citing Ex.,1006, 2:57-61; Ex.,1003, ¶ [90]).

Both meanings of the term "applet" explain that "applet" is an application program. Since applets 46_1-46_K are configured to operate in isolated execution Ring-3, Ex.,1006, 3:30-31, which "typically encompasses users or applications level and has the least privilege," Ex.,1006, 2:57-61, applets 46_1-46_K also would therefore be understood by a POSITA as encompassing user applications.

Ellison describes the applets 46_1-46_K are configured to operate on isolated execution mode as:

The isolated mode applets 46_1 to 46_K and their data are tamper-resistant and monitor-resistant from all software attacks from other applets, as well as from non-isolated-space applications (e.g., 42_1 to 42_N), dynamic link libraries (DLLs), drivers and even the primary operating system 12. Only the processor nub 18 or the operating system nub 16 can interfere with or monitor the applet's execution. Ex.,1006,4:1-7.

The operating system of Ellison includes two execution mode: normal execution mode and isolated execution mode. Applications 42_1-42_N and applets 46_1 to 46_K are all user applications, Applications 42_1-42_N are configured to

operate in normal execution mode while applets 46_1 to 46_K are configured to operate in isolated execution mode. Thus, a POSITA would understand that the counterpart of "user applications" of the '480 Patent includes applications 42_1-42_N and applets 46_1 to 46_K, correspondingly, the counterpart of "operating system" of the '480 Patent in Ellison is the operating system including normal execution mode and isolated execution mode, not only the primary operating system (OS) 12 operating in the normal execution mode, thus, the operating system should include the processor nub 18 and the operating system nub 16 which operate in the isolated execution mode.

An ordinary meaning of the term "access", as known in the art, is defined in "The Microsoft Computer Dictionary" as "(1) The act of reading data from or writing data to memory. (2) To gain entry to memory in order to read or write data. Ex. 2011, at 13. Thus, "inaccessible" would be understood by a POSITA as "having no ability to reading data from or writing data to memory".

Regarding the limitation "said decryption key being inaccessible to said user, to any user-operated software and to said operating system, wherein said memory is inaccessible to said operating system of said electronic device" of the '480 Patent, Ellison disclose that the decryption key and the memory are inaccessible to said user, to any user-operated software and even inaccessible to

normal execution mode of the operating system, but are accessible to isolated execution mode of the operating system.

Petitioner contends that Ellison's "OSNK 203 is used as a decryption key for decrypting the private key 204," wherein it is "generated by the key generator 240, which is part of the isolated execution environment in the electronic device of the first transaction party." Pet. at 63 (citing Ex. 1006, 8:25–65, 9:1–14, 11:11–30, Figs. 2, 3C). According to Petitioner, "OSNK 203 ('decryption key') is generated and used within the isolated environment region, which is not accessible to a user, any user-operated software, and the operating system (OS) 12 of the electronic device in the normal execution mode." *Id.* at 64 (citing Ex. 1006, 8:25–65, 11:1–12:26, 9:1–14, Figs. 2, 3C, 3D).

As for "the isolated environment region", Petitioner then contends that Ellison discloses "a secure platform 200 that 'includes the OS nub 16, the processor nub 18, a key generator 240, . . . , all operating within the isolated execution environment.'" Pet. 58 (citing Ex. 1006, 8:25–32; Ex. 1003 ¶ 95). Petitioner also contends that "the private key 204 can be stored in the cryptographic key storage 155 and in a multiple key storage of the non-volatile flash memory 160," and "only the OS nub 16 can retrieve the private key." Pet. at 59–60. According to Petitioner, because "the OS nub 16 is within the secure platform and isolated environment," it would have been obvious to a person of

ordinary skill in the art (POSITA) that "the cryptographic key storage 155 and the non-volatile flash memory 160 are secure storage locations so that the private key 204 can be accessed securely by OS nub 16." Pet. at 59–60, (citing Ex. 1003 ¶ 96).

It is apparent that Petitioner admits that Ellison disclose the decryption key and the memory are inaccessible to said user, to any user-operated software and even inaccessible to normal execution mode of the operating system, but are accessible to isolated execution mode of the operating system, such as can be "accessed securely by OS nub 16".

As set forth above, the counterpart of "operating system" of the '480 Patent in Ellison is the operating system including normal execution mode and isolated execution mode, not only the primary operating system (OS) 12 operating in the normal execution mode, thus, the operating system should include the processor nub 18 and the operating system nub 16 which operate in the isolated execution mode. Since the decryption key and the memory are accessible to isolated execution mode of the operating system, the decryption key and the memory are accessible to the operating system even though they are inaccessible to normal execution mode of the operating system.

Fig. 3 of the '480 Patent (reproduced below) illustrate a computer configuration which comprises a secure storage location inaccessible the operating system.

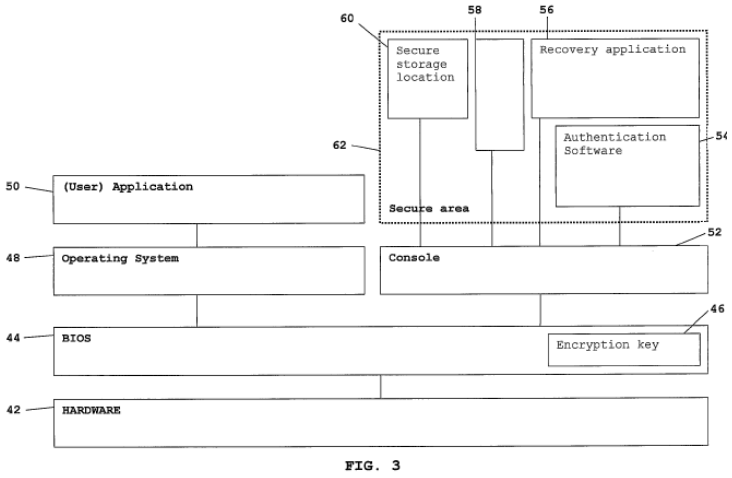


FIG. 3

Fig. 3 shows The BIOS 44 is a software located between the operating system 48 and the hardware 42, The BIOS 44 is located at a lower level than the operating system in the computer configuration.

The ’480 Patent describes that the BIOS 44 can prohibit the operating system 48 to access some storage location. “A user may instruct the operating system 48 to run an application 50. If the application 50 needs data that are stored on the hard drive, the application 50 requests the data from the operating system 48. The operating system 48 in turn requests the data through the BIOS 44 from the hardware 42, i.e. the hard drive. The data coming from the hardware 42 pass through the BIOS 44 and the operating system 48 before they arrive at the requesting application 50. Via this protocol the BIOS 44 may control the accessibility of certain data or hardware devices 42 and thus the use of particular

software. **It may prohibit, for example, the operating system 48 to access certain parts of a hard drive or start certain applications".** Ex.1001, 8:37-48.

A console 52 is an environment, which runs all the processes in the computer without user interruption.…….Any instructions coming from the operating system 48 intended for the console 52 may therefore be blocked by the BIOS 44. Ex.1001, 8:49-61.

In or behind the console 52, there is a **secure area 62 only accessible to the BIOS**. **The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48**. That means that the operating system 48 does not know that the applications and data in the storage locations are present and maybe even running in a separate environment. Ex.1001, 8:62-9:1.

With the authentication table securely stored in the secure storage location 60, the authentication software 54 should run in an environment in the BIOS 44, thus preventing that the authentication table is accessible to the operating system 48 at any time. Therefore, the authentication software 54 may be installed in an application environment in the secure area 62 such that it may obtain the authentication table without passing through an unsecured part of the device. Ex.1001, 9:11-19.

To guarantee the security of transactions, the '480 Patent storing the authentication data and authentication software in a secure storage location within

the BIOS or shield from the operating system by the BIOS, such that the decryption key and the memory are inaccessible to the operating system.

Ellison does not teach the secure platform 200 (on which the "OSNK 203 ('decryption key') is stored) is within the BIOS or shield from the operating system by the BIOS, On the contrary, Ellison disclose that the secure platform 200 operates by the operating system in isolated execution mode. Ellison disclose "a secure platform 200 that 'includes the OS nub 16, the processor nub 18, a key generator 240, . . . , all operating within the isolated execution environment.'" Ex. 1006, 8:25–32.

In isolated execution mode, the OS nub 16 can access application program (such as applets 46_1 to 46_K) located in the isolated area 70 of the secure environment. For example, [t]he operating system nub 16 provides links to services in the primary OS 12 (e.g., the unprotected segments of the operating system), provides page management within the isolated area, and has the responsibility for loading ring-3 application modules 45, including applets 46_1 to 46_K, into protected pages allocated in the isolated area. Ex. 1006, 3:54–59. The operating system nub 16 may choose to Support paging of data between the isolated area and ordinary (e.g., non-isolated) memory. If so, then the operating system nub 16 is also responsible for encrypting and hashing the isolated area pages before evicting the

page to the ordinary memory, and for checking the page contents upon restoration of the page. Ex. 1006, 3:62–4:1.

The OS nub 16 has the ability of loading including applets 46_1 to 46_K into the isolated area, of providing page management within the isolated area, of evicting the page (of the isolated area) to the ordinary memory, which means that the OS nub 16 can access the isolated area 70 (Which is in the system memory 140, Ex. 1006, 7:23-24). The secure platform 200 is operated in the isolated area 70, thus the secure platform 200 and the decryption key and the memory are accessible by the OS nub 16.

When a transaction is performed on Ellison's operating system, an attack can invade into the operating system to alter the execution mode of applets 46_1 to 46_K from isolated execution mode to normal execution mode, or invade into the isolated execution mode of operating system and control the operating system nub 16 which operate in the isolated execution mode to access the secure platform 200 to steal the decryption key and information in the memory of the isolated area 70. But when an attack invades into the operating system of the '480 Patent, since the secure area 62 only accessible to the BIOS, the attack can not steal any information from the secure area 62, the secure area 62 is transparent to the operating system of the '480 Patent. Therefore, the '480 Patent provide better protection from attack than Ellison's system.

Muir disclose an apparatus for using a virtual smart card, Muir's virtual smart cards 151 include a restricted memory space 170 to store a private key 167 and for performing operations such as encryption of data or generation of a digital signature using the private key. Ex. 1008, at 3, 6. The private key 167 is stored in a restricted memory space 170 to avoid the private key 167 from being exposed to the operating system 125 or an unauthorized user. Space is designated as restricted memory space 170, making it generally inaccessible to the operating system and normal applications running on top of the operating system. For one embodiment, this may be done by altering pointers or marking the sector bad, to indicate to the operating system that the data in that portion of memory is inaccessible. Ex. 1008, at. 9.

Muir only teaches that the private key 167 is inaccessible to the operating system, not disclose that the decryption key and the memory are inaccessible to the operating system.

Therefore, Ellison in view of Muir does not teach the limitation "said decryption key being inaccessible to said user, to any user-operated software and to said operating system, wherein said memory is inaccessible to said operating system of said electronic device"of the '480 Patent, particularly, Ellison in view of Muir does not teach the decryption key and the memory are inaccessible to the operating system.

ii Neither Ellison nor Muir teach or suggest the authentication software stored in a secure area inaccessible to the operating system

Petitioner relies on Ellison's usage protector 250 to allegedly disclose authentication software. Petition at 71. Usage protector 250 are loaded into the execution space by processor nub loader 52. Ex. 1006, FIG. 1A and 6:27–67. These loaded modules are stored in mass storage device 170 which can be CD ROM 172, floppy diskettes 174, and hard drive 176. Ex. 1006, 7:33–39. Petitioner also contends "that the usage protector 250 is run in an isolated environment ("***secure processing environment***") that is not accessible to the OS 12 ("***said operating system***")".Pet. at 73.(citing Ex.1006,3:9-61, 4:1-29, 4:57-5:27, FIGS. 1A, 1B; Ex.1003, ¶[111]).

As set forth above, the operating system of Ellison includes two execution mode: normal execution mode and isolated execution mode. the counterpart of "operating system" of the '480 Patent in Ellison is the operating system including normal execution mode and isolated execution mode, not only the primary operating system (OS) 12 operating in the normal execution mode, thus, the operating system should include the processor nub 18 and the operating system nub 16 which operate in the isolated execution mode.

Usage protector 250 operates in secure processing environment that is not accessible to the OS 12, but can be accessed by the processor nub 18 and the operating system nub 16 which operate in the isolated execution mode, that is,

usage protector 250 stored in a secure area accessible to the operating system in the isolated execution mode, thus, usage protector 250 stored in a secure area accessible to the operating system.

Muir only teaches that the private key 167 is inaccessible to the operating system, not disclose that the authentication software is inaccessible to the operating system.

Therefore, Ellison in view of Muir does not teach the limitation "the authentication software stored in a secure area inaccessible to the operating system" of the '480 Patent.

iii. Ellison in view of Muir does not render claim 1 obvious

Petitioner's combinations lack evidence of crucial claim limitations, such as the "decryption key and the memory are inaccessible to the operating system" and "the authentication software stored in a secure area inaccessible to the operating system" of claim 1 of the '480 Patent. Petitioner fails to show with particularity the *why* and *how* required for its obviousness combinations. Therefore, combination of Ellison and Muir does not render claim 1 obvious because it has not established its *prima facie* case, since "obviousness requires a suggestion of all limitations in a claim." *CFMT, Inc. v. Yieldup Intern. Corp.,* 349 F.3d 1333, 1342 (Fed. Cir. 2003).

D. Claims 16 -18

Claims 16 - 18 are similar to claim 1 but claims 16 and 18 are device claim. Claims 16-18 contain similar elements to Claim 1. Obviousness requires that all elements of the claim are disclosed by the cited references. Similar to argued above, Ellison in view of Muir does not teach "decryption key and the memory are inaccessible to the operating system" and "the authentication software stored in a secure area inaccessible to the operating system". Therefore, the references fail to teach or suggest all elements in Claims 16 -18. Therefore, Ellison in view of Muir does not render obvious claims 16 - 18.

E. Claims 3, 6, 11, 14-15

As discussed above, Ellison in view of Muir does not render Claim 1 obvious. Claims 3, 6, 11, and 14-15 depend from claim 1 and are not obvious over Ellison in view of Muir for at least the same reasons.

F. Claim 19

As discussed above, Ellison in view of Muir does not render Claim 18 obvious. Claim 19 depends from claim 18 and is not obvious over Ellison in view of Muir for at least the same reasons.

VII. CLAIMS 7,9,13 ARE NOT OBVIOUS OVER ELLISON, MUIR AND AL-SALQAN, CLAIMS 8,10 AND 12 ARE NOT OBVIOUS OVER ELLISON, MUIR AND SEARLE

In Grounds B2 – B3, Petitioner challenges claims 7, 8, 9, 10, 12, and 13 using the combination of Ellison and Muir with Al-Salqan (Ground B2) and Searle (Ground B3). Petitioner's position in Grounds B2 – B3 turns on the fundamentally false assertion that the proposed combination of Ellison and Muir can teach or suggest all of Patent Owner's independent claim features.

However, Petitioner does not argue that Al-Salqan or Searle cure the deficiencies noted above with respect to Ellison and Muir regarding claims 1, 3, 6, 11, and 14-19. Instead, Al-Salqan is cited for teaching multiple layers of encryption using private information (see, e.g., Petition, 93-95) and Searle is cited for teaching encryption based on hardware device serial number (see, e.g., Petition, 100-101).

Thus, Grounds B2 – B3 must fail for the reasons noted above with regard to the combination of Ellison and Muir. Therefore, Grounds B2 – B3 are fatally deficient and claims 7,9,13 are not obvious over Ellison, Muir And Al-Salqan, claims 8,10,12 are not obvious over Ellison, Muir And Searle.

VIII. COMMENTS ON DECISION Granting Post Grant Review

The PTAB states in the Decision granting institution of post grant review in Section II.B.1 on page 13 that:

"Nothing in the Specification, or the general knowledge in the art, limits "secure memory inaccessible to said operation system" to Ellison's secure "isolated storage area 70," or precludes Ellisons's "secure platform 200," as Patent Owner proposes. Prelim. Resp. 8-9. In fact, nothing in the Specification or the general knowledge in the art limits the "storage" to any amount of time in the secure memory.

We are unpersuaded by Patent' Owner's apparent position that software that is loaded into a secure location of execution is not software that is "stored" in a "secure memory" location. Prelim. Resp. 8-9.

In view of the Specification and the general knowledge in the art, we determine that the term "stored in the secure memory inaccessible to the operating system" encompasses being held in any secure location that is inaccessible to the operating system for an undefined amount of time. Our determination here is preliminary, and the parties are invited to address this claim construction at trial, if so desired." See Ex. 2012, ¶ 118.

A similar statement is made by the PTAB in the Decision granting institution of post grant review in Section II.B.5.iv on page 26 on the difference between "storing in secure memory" (permanently) and "loading into secure memory" (temporarily). See Ex. 2012, ¶ 119.

Patent Owner respectfully disagrees with the analysis of the PTA board based on how computer technology in 2000-2003 (the relevant period between the filing dates of Ellison, Muir and the application of '480) worked when retrieving data from permanent memory (hard disk, floppy disk, non-volatile memory also known as flash memory) to main memory (RAM memory) and when storing data from main memory to permanent memory. See Ex. 2012, ¶ 120.

When data/a data file (e.g., a program or a private key) is stored in permanent memory (hard disk, floppy disk, non-volatile memory also known as flash memory), it remains there until it is deleted. When instructed by the OS, the CPU reads/retrieves a particular data set through the intermediation of a data controller. The way this is accomplished differs depending on the hardware architecture, OS and the actual task at hand. However, every such read/data retrieval involves some sort of buffering: most programming languages and operating systems buffer at least part of I/O operations such as 'read' and 'write' to memory. This is typically done asynchronously: a buffer is created, filled, and then processed. For a read, the CPU would (working with the disk controller) create IO instructions to fetch data and a place to put it in RAM memory, fill that space, and then present its contents to the program making the request. For a write request, this would include queuing write operations and their associated data and then

sending them off to the IO controller and eventually the permanent memory to be executed. See Ex. 2012, ¶ 121.

The main principles that are adhered to in processors are the following:

a) Regardless of how the CPU reads/fetches/buffers/presents the contents of the data, the actual data remain **stored** in their original location in the permanent memory.

b) The data contents the CPU acquires are merely the result of an instance of a read/data retrieval. The number of CPU reads of one and the same data set in the specified location in permanent memory are without limit (e.g. reading the private key each time for successive electronic transactions)

c) The accessibility of the actual data in RAM memory is a completely different issue. If there are no architectural security restrictions in place, the CPU will execute the request to read the data and present the data to the program making the request. See Ex. 2012, ¶ 122.

The main conclusion is that data (or software) remains stored in the original location (permanent memory); this is independent of where read/retrieved data or software is subsequently 'held', 'located', or 'executed' or whether the retrieved data or software are stored/buffered in another memory as well. See Ex. 2012, ¶ 123.

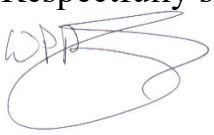
The read instruction has as consequence that data or software are loaded into a secure location of execution. The data or software itself remains stored in permanent memory and if the OS has unrestricted access to this permanent memory it continues to have access to the data of software. As a first example, consider Muir's Driver 155, containing address pointer 71 and the decryption key. As a second example, consider information in Ellison's such as code (e.g., processor nub 18), programs, files, data, applications (e.g., applications 421 to 42N), applets (e.g., 35 applets 461 to 46K) and operating systems stored in permanent memory (e.g., mass storage device 170 or non-volatile memory). In conclusion: while it helps to protect data or software during execution from the operation system by loading it temporarily into a secure memory, this is by itself not sufficient: it is also essential to store the said data or software securely in permanent memory in such a way that it cannot be read, accessed, or modified by the operating system. A temporary protection only is not sufficient as it will not allow to achieve the intended goals of the system (a secure processor or a secure digital transaction). See Ex. 2012, ¶ 124.

IX. CONCLUSION

For the reasons provided, Patent Owner respectfully requests that the Board confirm that the Challenged Claims of the '480 patent are patentable. See e.g., Ex. 2012, ¶¶ 4-11.

Dated: July 19, 2022

Respectfully submitted,



William P. Ramey, III
RAMEY LLP
Registration No. 44,295
Lead Counsel for Patent Owner
5020 Montrose Blvd., Suite 800
Houston, Texas 77006
(713) 426-3923 (telephone)
(832) 900-4941 (fax)
wramey@rameyfirm.com
uspto@rameyfirm.com
Counsel for Patent Owner
Ward Participations B.V.

CERTIFICATE OF SERVICE

I the undersigned, counsel with the law firm of Ramey & Schwaller, LLP, hereby certify that the following statements are true and correct under penalty of perjury, pursuant to 28 U.S.C. § 1746:

On this PATENT OWNER WARD PARTICIPATIONS B.V. RESPONSE TO PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 10,992,480 and EXHIBITS were served via email to the addresses below:

W. Karl Renner (Lead Counsel)
Registration No. 41,265
FISH & RICHARDSON P.C.
3200 RBC Plaza
60 South Sixth Street
Minneapolis, MN 55402
Tel: (202) 783-5070
Fax: (877) 769-7945
Service E-mail: IPR39843-0109IP1@fr.com

Jeremey J. Monaldo (Back-up Counsel)
Registration No. 58,680
Usman A. Khan (Back-up Counsel)
Registration No. 70,439
FISH & RICHARDSON P.C.
3200 RBC Plaza
60 South Sixth Street
Minneapolis, MN 55402
Tel: (202) 783-5070
Fax: (877) 769-7945 Service email:
PTABInbound@fr.com
Axf-ptab@fr.com
monaldo@fr.com
khan@fr.com

Dated: July 19, 2022

William P. Ramey, III
Ramey LLP
Registration No. 44,295
Lead Counsel for Patent Owner
5020 Montrose Blvd., Suite 800
Houston, Texas 77006
(713) 426-3923 (telephone)
(832) 900-4941 (fax)
wramey@rameyfirm.com
uspto@rameyfirm.com
Counsel for Patent Owner
WARD PARTICIPATIONS B.V.

CERTIFICATION OF WORD NUMBER

Counsel for Patent Owner hereby certifies that this PATENT OWNER AUTHWALLET, LLC'S PRELIMINARY RESPONSE TO PETITION FOR INTER PARTES REVIEW OF U.S. PATENT NO. 10,992,480 and EXHIBITS contains 16,295 words as calculated using the word count function of Microsoft Word, not including material excluded under 37 C.F.R. 42.24(a)(1).

Dated: July 19, 2022

William P. Ramey, III
Ramey LLP
Registration No. 44,295
Lead Counsel for Patent Owner
5020 Montrose Blvd., Suite 800
Houston, Texas 77006
(713) 426-3923 (telephone)
(832) 900-4941 (fax)
wramey@rameyfirm.com
uspto@rameyfirm.com
Counsel for Patent Owner
WARD PARTICIPATIONS B.V