

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

SEGA OF AMERICA, INC., UBISOFT, INC., KOFAX, INC.,  
CAMBIUM LEARNING GROUP, INC., and  
PERFECT WORLD ENTERTAINMENT, INC.,  
Petitioner,

v.

UNILOC USA, INC. and UNILOC LUXEMBOURG S.A.,  
Patent Owner.

---

Case IPR2014-01453<sup>1</sup>  
Patent 5,490,216 C2

Before WILLIAM V. SAINDON, DONNA M. PRAISS, and  
PATRICK R. SCANLON, *Administrative Patent Judges*.

PRAISS, *Administrative Patent Judge*.

FINAL WRITTEN DECISION  
*35 U.S.C. § 318(a) and 37 C.F.R. § 42.73*

---

<sup>1</sup> Case IPR2015-01026 has been joined with this proceeding.

## I. BACKGROUND

### A. Introduction

SEGA of America, Inc., Ubisoft, Inc., Kofax, Inc., and Cambium Learning Group, Inc. (collectively, “Petitioner”) filed a Petition (Paper 6, “Pet.”) to institute an *inter partes* review of claims 1–20 of U.S. Patent No. 5,490,216 (“the ’216 patent”). Uniloc USA, Inc. and Uniloc Luxembourg S.A. (collectively, “Patent Owner”) timely filed a Preliminary Response (Paper 10, “Prelim. Resp.”). Taking into account the arguments presented in Patent Owner’s Preliminary Response, we determined that the information presented in the Petition establishes a reasonable likelihood that Petitioner would prevail in challenging claims 1–20 of the ’216 patent under 35 U.S.C. §§ 102(b) and 103(a). Pursuant to 35 U.S.C. § 314, we instituted this proceeding on March 10, 2015, as to these claims of the ’216 patent. Paper 11 (“Dec. on Inst.”).

During the course of trial, we joined case no. IPR2015-01206, and Perfect World Entertainment Inc. as a Petitioner, to this proceeding. Paper 16. Patent Owner timely filed a Patent Owner Response (Paper 14, “PO Resp.”), and Petitioner timely filed a Reply to the Patent Owner Response (Paper 19, “Pet. Reply”). An oral hearing was held on December 2, 2015, and a transcript of the hearing is included in the record. Paper 26 (“Tr.”).

We have jurisdiction under 35 U.S.C. § 6(c). This decision is a Final Written Decision under 35 U.S.C. § 318(a) as to the patentability of claims 1–20 of the ’216 patent. For the reasons discussed below, we determine that Petitioner has demonstrated by a preponderance of the evidence that these claims are unpatentable under §§ 102(e) and 103(a).

*B. Related Matters*

The parties indicate that the '216 patent was asserted in complaints filed in the U.S. District Court for the Eastern District of Texas against SEGA of America, Inc. (No. 6:13-cv-627), Ubisoft, Inc. (No. 6:13-cv-628), Cambium Learning, Inc. (No. 6:14-cv-419), Kofax, Inc. (No. 6:14-cv-427), and Perfect World Entertainment, Inc. (No. 6:14-cv-429). Pet. 54; Paper 9. Additional litigations in which the '216 patent has been asserted are listed in Ex. 1031. The '216 patent is also the subject of IPR2015-01207, which was instituted on December 2, 2015, and petitions filed in IPR2016-00414 and IPR2016-00427.

The '216 patent was the subject of *Uniloc USA, Inc. v. Microsoft Corp.*, No. 03-0440 (D.R.I.) in which a decision on claim construction was issued (Ex. 1008) and affirmed by the Federal Circuit (Ex. 1009; Ex. 1010). PO Resp. 4; Pet. 3. The '216 patent was also the subject of two reexamination proceedings (Control Nos. 90/010831 and 90/012179). PO Resp. 4; Pet. 12–15. Additionally, the '216 patent was the subject of petitions for covered business method review (CBM2014-00183) and for *inter partes* review (IPR2015-00178), which were denied. PO Resp. 5 n.2; Pet. 55.

*C. The '216 Patent (Ex. 1001)*

The '216 patent, titled “System for Software Registration,” is directed to a system that allows software to run without restrictions (“use mode”) if a specified licensing procedure has taken place. Ex. 1001, Abstr. A code portion in the digital data to be protected may include an algorithm that generates a registration number unique to a licensee of the digital data. *Id.* The algorithm in the code portion is duplicated at a remote location under

the control of the licensor. *Id.* A mode switching means compares the local and remote registration numbers and, if they match, the program enters into a use mode. *Id.* at 4:49–54, 13:37–40. If they do not match, the program enters into a “demo mode” in which features of the program are disabled. *Id.*

The block diagram of Figure 8 of the '216 patent is reproduced below to illustrate the registration system:

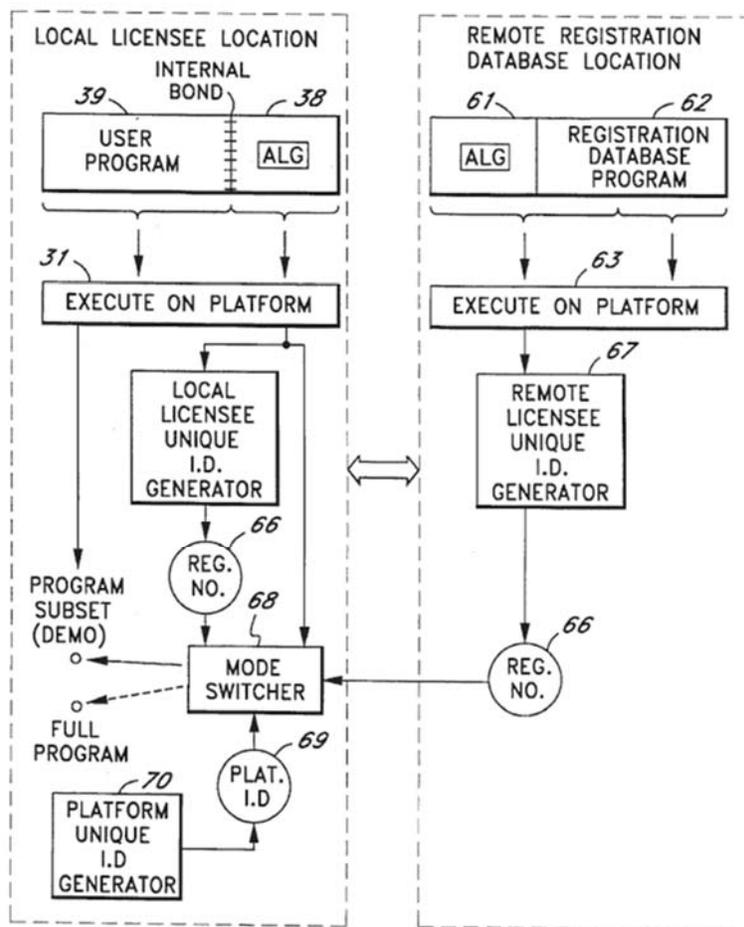


FIG. 8

The registration system depicted in Figure 8 operates in the manner generally described by the embodiments disclosed in the '216 patent. *Id.* at 11:43–45.

*D. Illustrative Claims*

Independent claims 1 and 12 are illustrative of the claims at issue (paraphrasing, indentations, and bracketed matter added):

1. A registration system for licensing execution of digital data in a use mode, said digital data executable on a platform, said system including

[a] local licensee unique ID generating means and remote licensee unique ID generating means,

[b] said system further including mode switching means operable on said platform which permits use of said digital data in said use mode on said platform only if a licensee unique ID first generated by said local licensee unique ID generating means has matched a licensee unique ID subsequently generated by said remote licensee unique ID generating means; and

[c] wherein said remote licensee unique ID generating means comprises software executed on a platform which includes the [sic] algorithm utilized by said local licensee unique ID generating means to produce said licensee unique ID.

12. A registration system attachable to software to be protected,

[a] said registration system generating a security key from information input to said software which uniquely identifies an intended registered user of said software on a computer on which said software is to be installed; and

[b] wherein said registration system is replicated at a registration authority and used for the purposes of checking by the registration authority that the information unique to the user is correctly entered at the time that the security key is generated by the registration system.

*E. Prior Art Relied Upon*

Petitioner relies on the following prior art:

Reference	Publication	Date	Exhibit
Haines	US 5,077,660	Dec. 31, 1991	1005
Logan	US 5,199,066	Mar. 30, 1993	1003
Grundy	US 5,291,598	Mar. 1, 1994	1004
Schull	US 5,509,070	Apr. 16, 1996	1002
Manduley	US 5,956,505	Sept. 21, 1999	1006

Petitioner also relies on the Declaration of Vijay K. Madiseti dated September 5, 2014 (“Madiseti Decl.” Ex. 1007).

*F. Instituted Grounds of Unpatentability*

We instituted this proceeding based on the asserted grounds of unpatentability (“grounds”) set forth in the table below. Dec. on Inst. 23.

Claims Challenged	Basis	Reference(s)
1–11, 17–20	§ 102(e)	Schull
10, 11	§ 103(a)	Schull
12–14	§ 102(e)	Logan
15, 16	§ 103(a)	Logan and Grundy
12–14	§ 103(a)	Haines and Manduley

II. ANALYSIS

*A. Claim Interpretation*

As a first step in our analysis, we determine the meaning of the claims. Because the challenged patent expired on September 21, 2013 and, as such, the claims are not subject to amendment, the rule of “broadest reasonable construction” per 37 C.F.R. § 42.100(b) does not apply. In this circumstance, the Board’s review of the claims is similar to that of a district court. *In re Rambus Inc.*, 694 F.3d 42, 46 (Fed. Cir. 2012). Specifically, the claim terms are given their ordinary and customary meaning, as would be understood by a person of ordinary skill in the art, at the time of the

invention, in light of the language of the claims, the specification, and the prosecution history of record. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313–17 (Fed. Cir. 2005) (en banc).

Petitioner proposes that we adopt the claim construction of the District Court for the District of Rhode Island issued in *Uniloc USA, Inc. v. Microsoft Corp.*, No. 03-CV0440 (“Microsoft litigation”) for purposes of the Petition “except where noted”. Pet. 3; *see* Ex. 1008 (claim construction decision and order dated Aug. 22, 2006); Ex. 1011, 27–28 (summary judgment decision clarifying claim construction dated Oct. 19, 2007). Petitioner asserts that the term “checking by the registration authority that the information unique to the user is correctly entered,” as recited in claim 12, lacks written description support. Pet. 6. Petitioner also asserts that there is no structure disclosed in the ’216 patent to support the term “platform unique ID generating means,” as recited in claims 7–9. *Id.* at 9. Nevertheless, Petitioner provides the District Court construction for these terms for the purpose of its unpatentability analysis.

Patent Owner does not propose alternative constructions to the District Court claim constructions, but contends that the Petition “reconstructs” the claim term “security key” and disputes Petitioner’s argument that certain claim terms lack definiteness or support in the specification for the required structure. PO Resp. 9–12. Regarding the term “security key,” Patent Owner acknowledges that the District Court stated “vendor information *may indeed be an input* to creating the licensee unique ID”, but asserts that the District Court “did not hold that the product number *alone* provides ‘a unique identifier associated with a licensee.’” *Id.* at 9–10 (quoting Ex. 1008, 12). Patent Owner further contends that the claim

construction of “security key” was not modified by the Federal Circuit, which stated that “a user intending to use the software in ‘use mode’ enters certain user information when prompted, which *may include* a software serial number and/or name and address information.” *Id.* at 10 (quoting Ex. 1010, 3). Patent Owner notes that the Federal Circuit decision held that the District Court’s construction of security key is correct “as a unique identifier associated with a licensee that can be, but is not limited to, personally identifiable information about the licensee or user” and that “the licensee unique ID *could encompass* vendor-supplied information.” *Id.* at 11 (quoting Ex. 1009, 13).

In the Decision to Institute, we agreed with the analysis by the District Court for the District of Rhode Island and adopted the claim construction issued in the Microsoft litigation. Dec. on Inst. 7; *see* Ex. 1008; Ex. 1011, 27–28. Given the parties’ acceptance of our constructions of each claim phrase in the Decision to Institute, we discern no reason to alter those constructions for the purpose of this Final Written Decision. For convenience, those constructions are reproduced in the table below.

<b>Claim Term</b>	<b>Claim Construction</b>
Licensee Unique ID (claims 1, 19, 20) Security Key (claims 12, 13) Enabling Key (claim 17)	A unique identifier associated with a licensee
Information uniquely descriptive of an intending licensee (claim 2) Information . . . which uniquely identifies an intended registered user (claim 12)	Information that is uniquely associated with a person who intends to become a licensee so as to access full functionality of the digital data
Algorithm (claims 1, 13, 14, 19, 20)	A set of instructions that can be followed to carry out a particular task

Claim Term	Claim Construction
Includes the algorithm utilized by said local licensee unique ID generating means to produce said licensee unique ID (claims 1, 19, 20)	Includes the identical algorithm used by the local licensee unique ID generating means to produce the licensee unique ID
Generated by a third party means of operation of a duplicate copy of said registration key generating means (claim 17)	Generated by a third party's use of a duplicate copy of the registration key generating means
Use mode (claims 1, 7, 19, 20) Fully enabled mode (claim 17) Full version run (claim 15)	A mode/version that allows full use of the digital data or software in accordance with the license
Partly enabled or demonstration mode (claim 17) Demonstration mode (claim 15)	A mode that allows partial use of the digital data or software
Has matched (claims 1, 17, 19, 20)	A comparison between the locally generated licensee unique ID/registration key and the remotely generated licensee unique ID/enabling key shows that the two are the same
Mode switching means will permit said data to run in said use mode in subsequent execution . . . only if said platform unique ID has not changed (claim 7)	The mode switching means will permit the data to run in the use mode only if the platform unique ID is identical to what it was the previous time the digital data were run
Registration system (claims 1, 12, 19, 20)	A system that allows digital data or software to run in a use mode on a platform if and only if an appropriate licensing procedure has been followed
Provided to said mode-switching means by said intending user (claim 17)	Provided to the mode-switching means by the person who intends to become a licensee
Communicated to said intending user (claim 17)	Communicated to the person who intends to become a licensee
Checking by the registration authority that the information	Verification by the registration authority that information unique to the

Claim Term	Claim Construction
unique to the user is correctly entered (claim 12)	user and entered by the user is accurate <sup>2</sup>
Wherein said registration system is replicated at the registration authority (claim 12)	Wherein the portion of the registration system that generates a security key from information input to software to be protected is reproduced exactly at the registration authority. This clarifies that only the portion of the registration system responsible for generating the security key must be replicated exactly at the registration authority, not the entire registration system.
Serial number (claim 14)	A number that is one of a series
Local licensee unique ID generating means (claims 1, 19, 20) Remote licensee unique ID generating means (claims 1, 19, 20) Registration key generating means (claim 17)	<b>Function:</b> to generate a local or remote licensee unique ID  <b>Structure:</b> a summation algorithm or a summer and equivalents thereof
Mode switching means (claims 1, 19, 20) Mode-switching means (claim 17)	<b>Function:</b> to permit the digital data or software to run in a use mode if the locally generated licensee unique ID matches with the remotely generated licensee unique ID  <b>Structure:</b> program code which performs a comparison of two numbers or a comparator and equivalents thereof

---

<sup>2</sup> The District Court used the term “verification” rather than “checking” in its claim construction for this term. Ex. 1008, 53. We note the use of “checking” instead in the Decision on Institution is a typographical error. See Dec. on Inst. 8.

Claim Term	Claim Construction
Platform unique ID generating means (claims 7–9)	<b>Function:</b> to generate a platform unique ID <b>Structure:</b> a summation algorithm or a summer and equivalents thereof

Regarding the Licensee Unique ID (claims 1, 19, 20), Security Key (claims 12, 13), and Enabling Key (claim 17) terms that are each construed to mean “a unique identifier associated with a licensee,” the level of uniqueness need not distinguish an individual licensee from all other licensees or persons. As found by the District Court in construing these claim terms, “one-of-a-kind” uniqueness of the identifier “is inconsistent with the language of the ’216 Patent itself.” Ex. 1008, 11 (citing Ex. 1001, Abstr., 6:23–26 (“[I]n particular preferred forms, a serial number . . . is included in the registration number generation algorithm which introduces an additional level of uniqueness”)). Because “unique” does not mean singularly unique, “the licensee unique ID does not require personal information about the user,’ so long as it is ‘unique,’ and not ‘based solely on platform-related user information.’” *Uniloc USA, Inc. v. Microsoft Corp.*, 632 F.3d 1292, 1300 (Fed. Cir. 2011) (quoting *Uniloc USA, Inc. v. Microsoft Corp.*, 290 Fed. App’x 337, 342–43 (Fed. Cir. 2008)) (Ex. 1010, 11). Because the licensee unique ID need not be personal information about the user, it can be vendor-supplied information; it just cannot be “based solely on platform-related user information” as that disavowal comes from the ’216 patent specification itself. *Id.* (citing *Uniloc USA, Inc.*, 290 Fed. App’x at 344); Ex. 1001, 2:5–7 (distinguishing prior art on the basis that “there is no suggestion or contemplation of linking platform identification with unique user identification”).

Regarding Petitioner’s assertion that certain claim terms lack definiteness or written description support, because those issues are neither appropriate for an *inter partes* review proceeding nor briefed by the parties, we do not address them in this decision. *See* 35 U.S.C. § 311(b). For purposes of this decision and for the reasons expressed by the District Court, we adopt and apply the constructions provided in the table above. *See* Ex. 1008, 53–54, 58–61.

*B. Priority Date for the Challenged Claims of the ’216 Patent*

The ’216 patent issued from U.S. Patent Application No. 08/124,718 (“the ’718 application”), filed September 21, 1993. Ex. 1001, [21]. The ’718 application claims the benefit of the following foreign priority applications: (1) Australian provisional patent application PL4842 (“the ’4842 application”), filed on September 21, 1992 (Ex. 1025); and (2) Australian provisional patent application PL5524 (“the ’5524 application”), filed on October 26, 1992 (Ex. 1026).

In the Decision on Institution, we explained that, based on the record prior to instituting trial, we were persuaded by Petitioner’s argument that the ’4842 and the ’5524 applications (collectively, “the Australian provisional applications”) do not provide sufficient written description support for the “generating means” (“local licensee unique ID generating means,” “remote licensee unique ID generating means,” and “registration key generating means,” collectively) and the “mode switching means” recited in independent claims 1, 17, 19, and 20. Dec. on Inst. 10–12 (citing Pet. 16–18). In particular, Petitioner argued that the structures identified in the District Court claim construction for performing the corresponding functions of the generating means and the mode-switching means are not present in

the Australian provisional applications. Pet. 15–18 (citing Ex. 1007 ¶¶ 23–27). For purposes of the Decision on Institution, we determined that Petitioner presented sufficient evidence indicating that the challenged claims of the '216 patent only are entitled to claim the benefit of the filing date of the '718 application—namely, September 21, 1993. Dec. on Inst. 11–12.

Although the burden of persuasion with respect to the unpatentability of the challenged claims remains with Petitioner, the burden of production of demonstrating that the challenged claims for the '216 patent are entitled to the earlier priority dates of the Australian provisional applications lies with Patent Owner. *See Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1379–80 (Fed. Cir. 2015). Therefore, we turn to Patent Owner's showing whether the Australian Provisionals “necessarily disclose” or “reasonably convey” the structure for (1) a summation algorithm or a summer and equivalents thereof; and (2) program code, which performs a comparison of two numbers or a comparator and equivalents thereof. *See Lockwood v. Am. Airlines, Inc.*, 107 F.3d 1565, 1572 (Fed. Cir. 1997) (a patentee demonstrates possession of the invention by describing it “in sufficient detail that one skilled in the art can clearly conclude that the inventor invented the claimed invention”); *Waldemar Link v. Osteonics Corp.*, 32 F.3d 556, 558 (Fed. Cir. 1994) (quoting *In re Rasmussen*, 650 F.2d 1212, 1215 (CCPA 1981)) (“The fact finder must determine if one skilled in the art, reading the original specification, would immediately discern the limitation at issue in the parent.”).

#### 1. *Generating Means*

The structural disclosure in the '216 patent for the term “licensee unique ID generating means” and other “generating means” listed in the

above table is “a summation algorithm or a summer and equivalents thereof”. *See* Ex. 1008, 25–27; Ex. 1010, 10, 20; Ex. 1001, 11:54–57, 12:62–65. The specific disclosure in the ’216 patent specification was described by the District Court as follows:

the only algorithm specified in the ’216 Patent for generating a licensee unique ID is found in the sixth embodiment, which states:

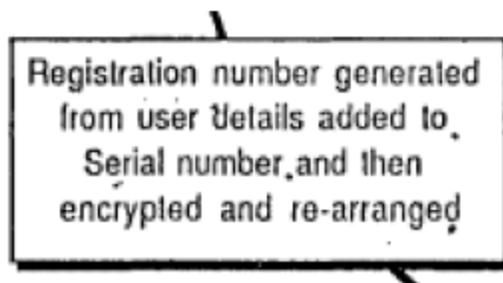
The algorithm, in this embodiment, combines by addition the serial number 50 with the software product name 64 and customer information 65 and previous user identification 22 to provide registration number 66.

’216 Patent, col. 11, ll. 53–56. Similarly, the only hardware component disclosed for performing the stated function is a ‘summer.’ *Id.* at col. 12, ll. 62–65.

Ex. 1008, 27.

Patent Owner contends that “[s]imilar structural language exists in the Australian Provisionals.” PO Resp. 16. Specifically, Patent Owner relies on the following disclosures in the ’4842 application text and Figure 2B as disclosing the claimed algorithm (*id.* at 16–17 (citing Ex. 1025, 4, Fig. 2B)):

Preferably said registration number algorithm combines information entered by a prospective registered user unique to that user with a serial number generated from information provided by the environment in which the software is to run ([e.g.,] system clock, last modify date, user name).



Registration number generated from user details added to Serial number, and then encrypted and re-arranged

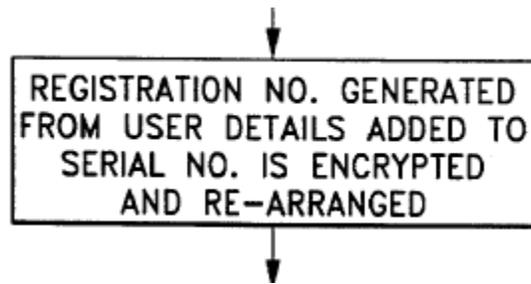
Fig. 2B, above, is a single box excerpted from a portion of a flowchart. *See* Ex. 1025, 5. Patent Owner asserts that the '5524 application contains similar disclosures and that "as a whole, each teaches a security key generated by a registration number algorithm that combines, by addition, information unique to an intended registered user, with a serial number." PO Resp. 17.

Patent Owner relies on the Declaration of Dr. Val DiEuliis (Ex. 2008) for the proposition that these disclosures of the Australian provisional applications convey to a person of ordinary skill in the art that "both the registration number and the serial number are numerical data (that is, numbers) and, as such, that FIG. 2B's disclosure of 'adding' means the addition of two numbers." PO Resp. 19 (citing Ex. 2008 ¶ 51). According to Dr. DiEuliis, Figure 2B "teaches that the algorithm that generates the registration number also adds the registration number (which was generated from user unique information) to the serial number . . . ." Ex. 2008 ¶ 53; PO Resp. 20. Dr. DiEuliis concludes that "the written description and figures taught an algorithm that uses addition to combine two numbers to arrive at a Licensee Unique ID, and that this algorithm would be 'fairly capable of categorization as a "summation algorithm,"' as explained by the Federal Circuit." Ex. 2008 ¶ 54; PO Resp. 20.

We are not persuaded by Patent Owner's evidence that the Australian provisional applications conveyed to a person of ordinary skill in the art a summation algorithm or a summer and equivalents thereof. The preponderance of the evidence is that the cited disclosures from the Australian provisional applications are insufficient for this purpose.

The two disclosures in the Australian provisional applications on which Patent Owner relies also appear in the '216 patent and are reproduced below from column 4 and Figure 2b:

10 Preferably, the registration number algorithm combines information entered by a prospective registered user unique to that user with a serial number generated from information provided by the environment in which the software to be protected is to run (e.g., system clock, last modify date, user name).



The excerpt from column 4 states that “the registration number algorithm combines information entered by a prospective registered user unique to that user with a serial number” and the excerpted box from Figure 2b states that the “registration no. generated from user details added to serial no. is encrypted and re-arranged”. Ex. 1001, 4:6–11, Fig. 2b. Despite these same two disclosures from the Australian provisional applications being present in the '216 patent, only the sixth embodiment in the '216 patent was found to provide structure for the generating means. Ex. 1008, 27 (“[H]aving scrutinized the '216 Patent in detail, the Court concludes that the only algorithm specified in the '216 Patent for generating a licensee unique ID is found in the sixth embodiment . . . .”); Ex. 1038, 24 (“There is no dispute that the generating means structure is fleshed out only in the sixth embodiment . . . .”). There is no dispute that the sixth embodiment is only present in the '216 patent; it is not disclosed in the Australian provisional applications. Therefore, these additional disclosures from the specification are insufficient for one skilled in the art to “immediately discern the

limitation at issue,” namely, a summation algorithm. *See Waldemar Link*, 32 F.3d at 558.

In addition, we do not find credible Dr. DiEuliis’s testimony that the text in Figure 2B, which states “registration no. generated from user details added to serial no. is encrypted and re-arranged,” describes the addition of two numbers (registration no. and serial no.). There is no previous step in the flowchart depicted in Figure 2B that describes the step of generating the registration number to indicate that the registration number is an input, rather than an output. Ex. 1025, Fig. 2B; *see* Ex. 1026, Fig. 2. Further, as explained next, the “user details” encompass text, which further explains why the flowchart is not discussing the addition of numbers.

If the text box in Figure 2B is properly understood to describe the generation of the registration number from the inputs of user details and serial number, the Australian provisional applications lack a disclosure of how to combine user data and information that is in different formats. Ex. 2009, 127:2–4, 62:19–23, 79:19–80:2, 125:7–14, 82:2–7, 87:13–18, 88:7–15, 132:24–133:14. “[I]nformation entered by a prospective registered user” and “user details” are not necessarily numbers because user details includes such information as name and address according to the ’216 patent and the Australian provisional applications. *See* Ex. 1001, 3:50–53 (“Preferably, the information utilized by the local licensee unique ID generating means to produce the licensee unique ID comprises prospective licensee credit card number, date of birth and full name and address.”); Ex. 1025, 7 (“The registration dialogue box C prompts the user for details unique to that user (including, for example, name, company, address, state, contact number) . . . .”); Ex. 1026, 8 (“The registration dialogue box C prompts the

user for details unique to that user (including, for example, name, company, address, state, contact number) . . .”).

Even if the flowchart box in the Australian provisional applications were to reasonably convey to one skilled in the art a simple addition operation of adding two numbers together, as Dr. DiEuliis contends by the addition of a registration number to a serial number (Ex. 2008 ¶ 53), that disclosure also would be insufficient to reasonably convey a summation algorithm. The summation algorithm structure is not simple addition. Ex. 1010, 20 (the Federal Circuit held that “[t]he structural disclosure in the ’216 patent is not limited to simple addition in the colloquial sense of adding numbers together and nothing more”). As the Federal Circuit explained, the “combination by addition” taught by the sixth embodiment of the ’216 patent “necessarily incorporates an initial step of converting the information into a common format to be added, which requires more than simple addition.” *Id.*

Regarding the textual disclosure in the Australian provisional applications that the “registration number algorithm combines information” (Ex. 1025, 3), Patent Owner contends that the multitude of ways one could have combined information in 1992 would have narrowed to “summation” upon reading the totality of the Australian provisionals, particularly the “adding” disclosed in Figure 2B. PO Resp. 20–21 (citing Ex. 2008 ¶¶ 58, 61). A disclosure of a genus is not necessarily a disclosure of all of its species, however. *In re Baird*, 16 F.3d 380, 382 (Fed. Cir. 1994). We are not persuaded by Patent Owner’s evidence that the disclosure teaches combination by addition based on the trial record. Dr. Madisetti testified that there are a number of different ways to combine letters and numbers

without mathematical addition. Ex. 1007 ¶ 24; Ex. 2009, 121:13–124:1, 128:5–16, 128:25–129:12, 132:11–20, 154:22–155:5. As examples of how information can be combined in a non-mathematical manner, Dr. Madisetti testified that “[y]ou could put a code for the different digits and scramble them up. You could take portions of each and try to create another registration number. You could use different operations in different ways.” *Id.* at 121:20–24. Dr. Madisetti also testified that the word “add” does not necessarily mean “sum” because it can also describe “adding a redundancy” and “add[ing] a header”. Thus data can be amalgamated into an alphanumeric number to form a registration number. *See* Tr. 83. We credit Dr. Madisetti’s testimony on the meaning of “add” and “combine” in the context of the Australian provisional applications. In addition, Patent Owner does not dispute that there is more than one way to combine information. *Id.* at 44 (“[T]here are only two forms of combining information in the totality of the evidence in this case: Summation, the mathematical operation, and concatenation”).

At oral hearing, Patent Owner argued that the sixth embodiment in the ’216 patent that describes the summation algorithm structure for the claimed generating means incorporates by reference prior embodiments including “everything that is disclosed in figure 2.” *Id.* at 41. It is Patent Owner’s position that for the summation algorithm structure disclosed in the sixth embodiment in the ’216 patent, “there is [] traceability back to the Australian provisionals.” *Id.* at 42. As Petitioner pointed out in rebuttal, however, the ’216 patent characterizes the disclosures that also appear in the Australian provisionals as a “generalized description”. Ex. 1001, 11:40–43; Tr. 84. Thus, in the context of the ’216 patent itself, the disclosures that appear in

the Australian provisional applications do not convey the specific algorithm disclosed in the sixth embodiment, namely, combination by addition, nor an equivalent, but, rather, a generalized description. *See* Ex. 1001, 11:53–56.

In sum, we credit Dr. Madisetti’s opinion on whether one of ordinary skill in the art would have immediately recognized from the Australian provisional applications the structure of a summation algorithm or summer and its equivalents. Accordingly, we find that the preponderance of the evidence supports the earliest priority date of September 21, 1993 for the ’216 patent based on the earliest disclosure of structure and hardware for the claim term “licensee unique ID generating means” and similar terms “remote licensee unique ID generating means,” and “registration key generating means”.

## 2. *Mode Switching Means*

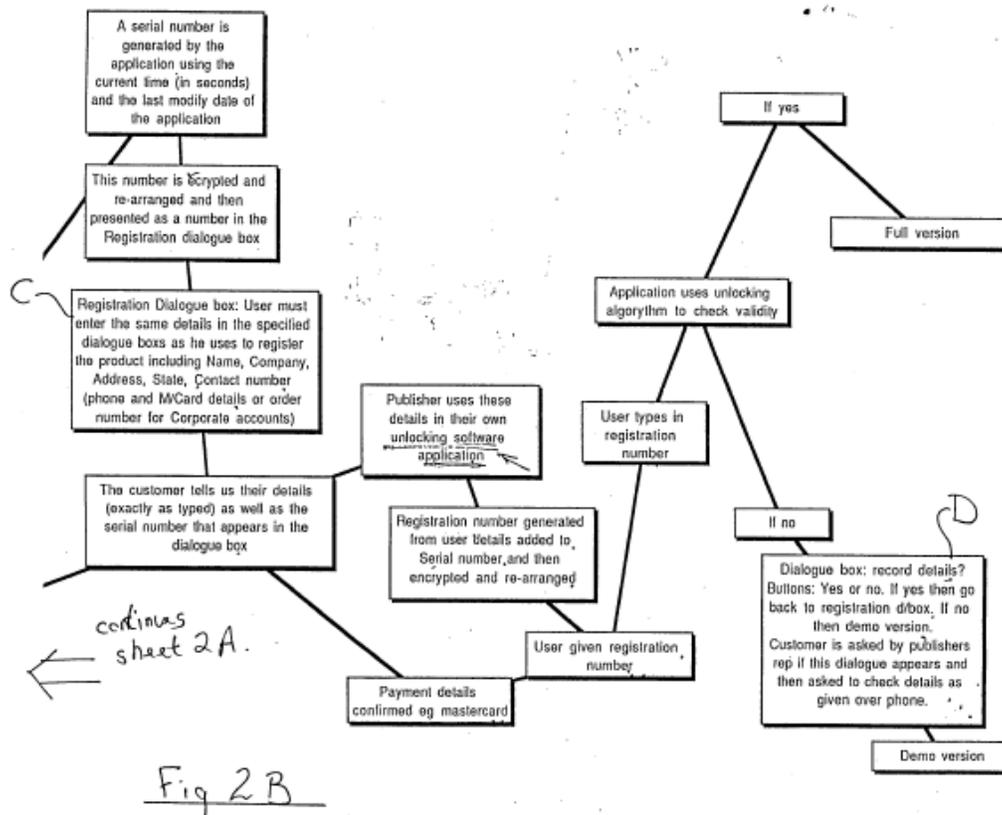
The structural disclosure in the ’216 patent for the term “mode switching means” is “program code which performs a comparison of two numbers or a comparator and equivalents thereof”. *See* Ex. 1008, 41–44; Ex. 1001, 13:37–40 (“Comparator 90 together with gates 91, 92, and relay 93 comprise one particular form of mode switcher or switching platform 83 of various kinds of code such as the code of types D and U”), 6:12–14 (“[m]ode switching means can comprise execution of the code portion which additionally performs a comparison of the locally and remotely generated registration numbers”).

Patent Owner asserts that the function of the mode switching means is described in the ’4842 application as follows:

As the final stage in registration the registration authority 16 provides the registration number generated by the registration authority PC 15 to the user 11. The user 11 enters the registration

number into the user PC 12 where the security routine checks to see whether the entered registration number matches the calculated registration number. If the two match then a valid registration has taken place and access is provided by the security routine to a full operating version of the software protected by the security routine.

Ex. 1025, 6–7; PO Resp. 26 (citing Ex. 2008 ¶ 63). Patent Owner further asserts one skilled in the art would have understood from the above disclosure in the Australian provisional applications that (1) they “teach a software invention”, (2) “a ‘routine attachable to software’ is software,” (3) “‘software’ is implemented with ‘program code’”, and (4) “the terms ‘program code’ and ‘software’ are often used interchangeably.” PO Resp. 27 (citing Ex. 2008 ¶¶ 65, 66). Patent Owner also relies on Figure 2B of the Australian provisional applications as supporting disclosure of a “mode switching means”. *Id.* at 28 (citing Ex. 2008 ¶ 68). At the oral hearing, Patent Owner argued that the third block from the top on the right side of Figure 2B is a decision block that is a software equivalent to a comparator and a model for software, which is program code. Tr. 47–48; Ex. 1025, 13. Figure 2B is shown below:



According to Patent Owner, Figure 2B shows a decision block in the third block from the top (“Application uses unlocking algorithm to check validity”) because there is an input and two outputs that are a “yes” and a “no”. PO Resp. 28 (citing Ex. 2008 ¶ 68); Ex. 1025, 13; Tr. 47.

Patent Owner provides evidence that the function of the mode switching means was disclosed in the Australian provisional applications, but not the structure, namely, program code or its equivalent, or the hardware, a comparator. The fact that the security routine, which provides the function, is “attachable to software” and that software is “often used interchangeably” with program code, is not evidence that one of ordinary skill in the art would have clearly concluded from the Australian provisionals that program code, or an equivalent, is the structure that performs the mode switching means function. *See Lockwood*, 107 F.3d at

1572. Moreover, Patent Owner does not provide evidence of a comparator or an equivalent being disclosed in the Australian provisionals to demonstrate that the inventor was in possession of the invention at the time of the filing of the Australian provisional applications. *See id.* Figure 2B does not reasonably convey a comparator, nor does the record reflect evidence that the block in Figure 2B to which Patent Owner directs us is a comparator or an equivalent.

Accordingly, we find that the preponderance of the evidence supports the earliest priority date of September 21, 1993 for the '216 patent based on the earliest disclosure of structure and hardware for the claim term “mode switching means”.

### 3. *Conclusion*

The preponderance of the evidence on this record shows that the earliest priority date to which the '216 patent claims are entitled is September 21, 1993. We, therefore, discern no reason to alter our determination in this regard for the purposes of this Final Written Decision.

### C. *Patentability*

To prevail on its challenges to the patentability of claims, Petitioner must prove unpatentability by a preponderance of the evidence. 35 U.S.C. § 316(c); 37 C.F.R. § 42.1(d).

#### 1. *Principles of Law*

##### a. *Anticipation*

In order for a prior art reference to serve as an anticipatory reference, it must disclose every limitation of the claimed invention, either explicitly or inherently. *In re Schreiber*, 128 F.3d 1473, 1477 (Fed. Cir. 1997). We must

analyze prior art references as a skilled artisan would. *See Scripps Clinic & Research Found. v. Genentech, Inc.*, 927 F.2d 1565, 1576 (Fed. Cir. 1991), overruled on other grounds by *Abbott Labs. v. Sandoz, Inc.*, 566 F.3d 1282 (Fed. Cir. 2009) (to anticipate, “[t]here must be no difference between the claimed invention and the reference disclosure, as viewed by a person of ordinary skill in the field of the invention”).

*b. Obviousness*

A claim is unpatentable under 35 U.S.C. § 103(a) if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of underlying factual determinations including: (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of ordinary skill in the art; and (4) objective evidence of nonobviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). An invention “composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the prior art.” *KSR*, 550 U.S. at 418. Moreover, a determination of unpatentability on the ground of obviousness must include “articulated reasoning with some rational underpinning to support the legal conclusion of obviousness.” *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006). The obviousness analysis “should be made explicit” and it “can be important to identify a reason that would have prompted a person of ordinary skill in the

relevant field to combine the elements in the way the claimed new invention does.” *KSR*, 550 U.S. at 418.

## 2. *Schull (Ex. 1002)*

Petitioner asserts that claims 1–11 and 17–20 of the ’216 patent are anticipated by Schull under 35 U.S.C. § 102(e). Pet. 19–37. Petitioner also asserts that claims 10 and 11 would have been obvious over Schull under 35 U.S.C. § 103(a). *Id.* at 53. Having determined that the earliest priority date of the ’216 patent is September 21, 1993, we confirm that Schull is prior art to the ’216 patent under 35 U.S.C. § 102(e) because it issued from an application filed in the United States on December 15, 1992. *See* Ex. 1002, [22].

### *a. Overview of Schull*

Schull teaches a method of distributing, registering, and purchasing digital information whereby access to advanced features of the digital information is given in the presence of a valid password that is generated on the user’s system. Ex. 1002, Abstr. The password is generated using ID target information that can be unique to the user, such as the user’s voice or telephone number, or specific to the user’s processor. *Id.* at 5:20–47; 6:65–7:27; 8:26–30; 17:13–20. An algorithm is used to transform the information into a unique ID. *Id.* at 7:16–27. The same password-generating algorithm is used on a licensing processor that is remote from the user’s computer to transmit the password back to the user’s processor where it is installed and found for subsequent executions or boots. *Id.* at 6:1–11, 8:55–9:4, 11:8–13, 35–40, 51–54. A check is conducted between the user’s processor and the licensing processor to determine whether the installed password correctly matches the password generated in the user’s processor. *Id.* The protected

software may then be run on a user's processor, which "is typically a traditional computer". *Id.* at 6:46–53.

*b. Anticipation of Claims 1–11 and 17–20*

The key issue disputed by the parties is whether Schull teaches the "generating means" required by the claims. Petitioner contends that the preferred algorithm taught by Schull is a summation algorithm or an equivalent and provides the Madisetti Declaration in support of its position. Pet. 21 (citing Ex. 1007 ¶¶ 40–47). According to Dr. Madisetti, the preferred algorithm disclosed in Schull is a summation algorithm because it prominently uses addition to perform the function of generating an ID. Ex. 1007 ¶¶ 40–47.

Petitioner also argues that the presence of an operating system, as required by dependent claims 10 and 11, is implicitly if not expressly present in Schull's disclosure because traditional personal computers were almost universally being used by 1993, thus any program running on a computer would have been adapted to run under that operating system or in an operating system environment. Pet. 22–23. Petitioner alternatively argues that claims 10 and 11 would have been obvious in view of Schull. *Id.* at 53. For both positions, Petitioner provides the Madisetti Declaration as evidence. Ex. 1007 ¶ 38.

Patent Owner contends that Petitioner's challenge based on Schull is defective "because Schull fails to teach a summation algorithm, summer, or equivalent structure for anticipating the licensee unique ID generating means." PO Resp. 33 (citing Ex. 2008 ¶¶ 75–105). According to Patent Owner, Schull uses "concatenation" to generate its "Passwordable ID" and "concatenation" is not a summation algorithm. *Id.* at 33–34 (citing Ex. 2008

¶ 77). Dr. DiEuliis describes concatenation as

the linking together of entities (e.g., characters or numbers), not a mathematical computation, summation or otherwise. In a computer system, data is stored in memory as a linear array of bytes, and concatenation is normally accomplished by copying the data to a contiguous section of memory so that the result is stored as a continuous array.

Ex. 2008 ¶ 78. Dr. DiEuliis provides as an example of concatenation “the concatenation of three numbers—X=1234; Y=56; Z=789—to arrive at the number 123456789” provided in Dr. Madisetti’s declaration. *Id.* ¶ 82 (citing Ex. 1007 ¶ 44). According to Dr. DiEuliis, to concatenate these three numbers stored in memory requires rearranging the numbers by moving or copying into a contiguous section of memory. “To accomplish this rearrangement, at most, the program need only to move each byte to its new location in memory.” *Id.* ¶ 83. Dr. DiEuliis states “[m]oving data from one memory location to another . . . is a basic processor operation and is fast and efficient. Any type of arithmetic operation such as addition by byte-wide or multi-byte numbers is much more complicated and significantly slower.” *Id.* (footnote omitted).

Because Petitioner relies upon the Declaration of Dr. Madisetti to explain the method of concatenation in Schull, Patent Owner argues that Petitioner supplements the teachings of Schull with other teachings of Dr. Madisetti. PO Resp. 40; Tr. 52.

Patent Owner does not present separate argument as to anticipation of claims 1–11 and 17–20 by Schull and obviousness of 10 and 11 over Schull. PO Resp. 32–40, 60.

The passages at issue in Schull describing the algorithms by which the Passwordable ID is generated read:

The ID must be generated in such a way that two ID-Targets will generate different IDs. Also, in order that a plurality of Licensed-features in a plurality of software programs be independently licensable on the same ID-Target, any two Licensed features must be able to generate different IDs even in conjunction with a single ID-Target. Those familiar with the art will recognize that this can be achieved a variety of ways, in one preferred embodiment, each item of protected software is assigned an adequately unique P-digit Program ID, and each licensed Feature is assigned an F-digit Feature-ID, and each ID-Target can be associated with a T-digit Target-ID such as a serial number. Once assigned (using methods described below) these ID numbers are combined in a fashion which preserves their uniqueness (e.g., by concatenating them to produce a number with N+M+T digits capable encoding  $10^{(N+M+T)}$  values) and then using this combination, an encryption of it, or some other adequately-unique transform of it, as the ID.

Ex. 1002, 7:10–27.<sup>3</sup> Schull goes on to describe “error-checking” by having the Passwordable ID “satisfy some kind of coherence constraint” that “would be to append two more digits to the ID which would constitute a checksum for the preceding digits.” *Id.* at 7:28–36.

We are persuaded by Petitioner’s argument that Schull discloses a summation algorithm for generating its Passwordable ID. As explained by Petitioner, Schull’s algorithm combines three ID numbers: Program ID, Feature ID, and Target ID, which Petitioner refers to as X, Y, and Z in its explanation. Pet. 21. The X number has “N” digits, the Y number has “M” digits, and the Z number has “T” digits. *Id.* The algorithm combines the numbers X, Y, and Z “(e.g., by concatenating them to produce a number

---

<sup>3</sup> Patent Owner does not dispute Dr. Madisetti’s observation that Schull contains a typographical error (referring first to P, F, and T-digits and then to N, M, and T-digits) and his determination that N and M are equivalent to P and F. Ex. 1007 ¶ 41.

with N+M+T digits)”. Ex. 1002, 7:24–25; Pet. 21; Ex. 1007 ¶ 41. In 1992, concatenating three integers to produce the desired result “boils down to two basic approaches: (1) multiplying the first integer and second integer by a power of ten (dependent on the number of digits of the subsequent numbers) and adding the three integers together, or (2) converting the integers to ‘strings,’ concatenating the strings, and converting the result back to an integer.” Ex. 1007 ¶ 42. Using the former approach to produce a number with the combined number of digits as the numbers being combined, which Petitioner argues is the one disclosed in Schull, Dr. Madisetti describes the mathematical operations that can be performed programmatically. The mathematical operation provided by Dr. Madisetti is  $X*10^{(M+T)} + Y*10^T + Z$ , which uses multiplication and addition. *Id.* ¶¶ 44; Pet. 21. According to Dr. Madisetti, this approach is a summation algorithm and is “computationally quicker” for the reason that “it is a matter of performing basic calculations and processing smaller numbers, as opposed to converting, combining and reconverting large strings . . . .” Ex. 1007 ¶ 43.

To explain why Dr. DiEuliis is incorrect about Schull using the other method of concatenating three integers to generate the PasswordableID, Petitioner submits with its Reply a Supplemental Declaration of Dr. Vijay K. Madisetti (Ex. 1039). Pet. Reply 9–10. In his Supplemental Declaration, Dr. Madisetti provides a detailed explanation based on Schull’s teachings about the programming used to generate a PasswordableID. Ex. 1039 ¶¶ 7–14. Regarding how the invention of Schull is implemented, Schull states that “[o]ne object of [its] invention is to allow programmers to conveniently invoke the first-described methods by adding a relatively small number of lines of code to their own programs.” Ex. 1002, 12:46–50. Schull further

discloses that this object can be achieved by implementing “Pascal language” and discloses programming that describes the Passwordable ID as a “longint.” *Id.* at 12:53–14:13; Ex. 1039 ¶¶ 7–14. According to Dr. Madisetti, “longint” means the Passwordable ID is “a single integer, single number, or a whole number.” Ex. 1039 ¶ 11. Concatenating a PasswordableID in the manner described by Dr. DiEuliis would not produce a single integer, single number, or a whole number because the components of the number would maintain their separate identities stored in separate places in memory. *Id.* ¶¶ 15–16. In view of the evidence before us, we are persuaded that Schull’s longint concatenating implementation describes a summation algorithm.

Even if the concatenating procedure disclosed by Schull for generating a Passwordable ID does not necessarily utilize a “summation algorithm” or equivalent, we find that Schull’s disclosure of appending two additional digits to the concatenated number using a checksum does. Schull describes this further step in creating the Passwordable ID as being for the purpose of error checking. Ex. 1002, 7:28–36. The preponderance of evidence on this record is that a checksum is a summation algorithm. Dr. Madisetti testifies that “[a]s of 1992, using a checksum to detect an error in a number, as described by Schull, was done using what is known as a ‘check digit[,]’ [and all] of the methods for calculating check digits utilize some form of addition.” Ex. 1007 ¶ 48. Dr. DiEuliis confirmed that most checksums use addition and that he had never created a checksum that did not use summation. Ex. 1041, 63:10–17, 88:8–10. In addition, the publications relied upon by Dr. DiEuliis for disclosing checksum methods also confirms that summation is used in those methods. Ex. 1039, 24–36;

2009, 144:16–20. The issue of whether a checksum is a summation algorithm was also answered in the affirmative in prior litigation involving the '216 patent. Ex. 1032, 177:7–21; Ex. 1042, 20–21; Ex. 1010, 17, 24; Ex. 1037, 30, 41–42; Ex. 1033, 52, 53–55; Ex. 1016, 27; Ex. 1041, 61:21–62:20, 70:8–72:22, 80:10–81:21. Therefore, Schull discloses a licensee unique ID generating means because Schull discloses the step of appending digits for error-checking purposes to the Passwordable ID using a checksum.

At oral hearing, Patent Owner asserted that (1) Petitioner is relying on an inherency argument to establish that Schull used a summation algorithm, (2) that inherency argument must fail because the experts agree that there are two methods in which concatenation may be performed, and (3) [t]here is no express disclosure in Schull for using summation, only concatenation.” Tr. 52–55. When asked how the other form of concatenation would produce a long integer, Patent Owner conceded that there is no evidence in the record to that level of detail. *Id.* at 54:8–16 (“We don’t have expert testimony on that level of detail”).

After considering Petitioner’s and Patent Owner’s positions, as well as their supporting evidence, we credit Dr. Madisetti’s testimony regarding the disclosures in Schull describing the use of a summation algorithm to generate the PasswordableID. For the reasons provided by Petitioner, we determine that the preponderance of the evidence demonstrates that claims 1–11 and 17–20 would have been anticipated by Schull. *See* Pet. 19–37; Pet. Reply 9–14.

*c. Obviousness of Claims 10 and 11*

Claim 10 depends from claim 1 and recites “said platform comprises a computer operating system environment.” Ex. 1001, 14:35–36. Claim 11

depends from claim 10 and further requires “said digital data comprises a software program adapted to run under said operating system environment.” *Id.* at 14:37–39. Petitioner argues that if Schull’s disclosure of a traditional computer on which protected software may be run is not sufficient to anticipate dependent claims 10 and 11, then “modifying the ‘traditional computer’ of Schull to include an operating system under which protected software can be run would have been obvious to one of skill in the art.” Pet. 53 (citing Ex. 1007 ¶ 38). Patent Owner contends that claims 10 and 11 are not unpatentable because Schull is not prior art. PO Resp. 60.

A disclosure that anticipates under 35 U.S.C. § 102 generally renders the claim unpatentable under 35 U.S.C. § 103, because anticipation is the “epitome of obviousness.” *See In re Fracalossi*, 681 F.2d 792, 794 (CCPA 1982); *In re Pearson*, 494 F.2d 1399, 1402 (CCPA 1974). For the reasons we find that claims 10 and 11 have been shown by a preponderance of the evidence to be anticipated by Schull, we also determine that Petitioner has shown by a preponderance of the evidence that claims 10 and 11 are unpatentable under 35 U.S.C. § 103(a) as obvious over Schull.

We further find that if Schull’s disclosure of a “traditional computer” is not a disclosure of a personal computer with an operating system, the preponderance of the evidence on this record shows that it would have been obvious to modify Schull’s traditional computer to include an operating system under which protected software can be run as required by claims 10 and 11. According to Dr. Madisetti, “operating systems for personal computers had become ubiquitous and necessary to the operation of software on the computer” prior to 1993. Ex. 1007 ¶ 38. Dr. Madisetti testifies to the introduction of Windows 3.1 in April 1992, the release of Apple’s

System Software in 1984, System 7 in 1991, and LINUX in 1991. *Id.* The '216 patent also identifies existing operating system environments. Ex. 1001, 2:32–36. These facts are not disputed by Patent Owner. Therefore, if Schull's traditional computer did not include an operating system environment on which software can be run, it would have been obvious to modify Schull with a computer that did include an operating system environment. Accordingly, the preponderance of the evidence supports our finding that claims 10 and 11 would have been obvious over Schull.

3. *Logan (Ex. 1003) and Grundy (Ex. 1004)*

Petitioner asserts that claims 12–14 are anticipated under 35 U.S.C. § 102(e) by Logan (Pet. 37–41) and claims 15 and 16 would have been obvious under 35 U.S.C. § 103(a) over the combination of Logan and Grundy (*id.* at 41–46). Claim 12 requires that the registration system “is replicated at a registration authority and used for the purposes of checking by the registration authority that the information unique to the user is correctly entered at the time that the security key is generated by the registration system.” Ex. 1001, 14:45–49. Claim 15 depends from claim 12 and further requires that “said registration system checks at the time of boot of said software as to whether it is a first boot of the software to be protected or a subsequent boot”. *Id.* at 14:57–60.

*a. Overview of Logan*

Logan discloses a method and system for protecting a software program. Ex. 1003, Abstr. A first software code or serial number is provided by the vendor that is unique to each original copy of the software. *Id.* at 4:19–31. A second software code is stored within the software that the software supplier is able to identify by reference to the first software code.

*Id.* at 4:32–43. The user must provide the software supplier with the hardware and software serial numbers. *Id.* at 6:33–39. An activation code is generated by the software supplier by adding together the serial numbers and in the same manner accomplished by the software locally to generate a first intermediate code. *Id.* at 6:51–67. A mathematical operation is performed on the first intermediate code and the activation code to produce a second intermediate code. *Id.* at 5:53–65. The program compares the second intermediate code with the second software code and, if they are identical, then the user is permitted to operate the software uninhibited. *Id.* at 5:67–6:7.

*b. Anticipation of Claims 12–14*

Petitioner asserts that the disclosures in Logan anticipate claims 12–14. Pet. 37–41. Petitioner identifies Logan’s first intermediate code as the security key required by independent claim 12 and Logan’s first software code or serial number as the information uniquely associated with a person who intends to become a licensee that the user inputs. *Id.* at 38–39. Logan’s software supplier is identified as the registration authority that checks whether the software serial number entered by the user is accurate because the second intermediate code will not match the stored hidden number if the serial number is not correctly input by the user. *Id.* at 40.

Patent Owner contends that the first intermediate code derived in Logan “is not generated from information unique to the user because different users who install copies of the software will have the same software serial number.” PO Resp. at 41–42 (citing Ex. 2008 ¶ 121; Ex. 1003, 6:7–29). According to Patent Owner, vendor-supplied information may be an input to generate a “security key,” but it is not necessarily

“information that is uniquely associated with a person” as recited in claim 12. *Id.* at 43. Patent Owner also contends that Logan’s system does not anticipate the final element required by claim 12, “checking by the registration authority that the information unique to the user is correctly entered at the time the security key is generated by the registration system” because “the temporal aspect of the checking (i.e. ‘at the time the security key is generated’) is not disclosed anywhere in Logan.” *Id.* at 45 (citing Ex. 2008 ¶ 131). Patent Owner further contends that Petitioner’s citation to Logan for this element is misleading because Logan “describes actions that take place on the user’s computer, not actions taken by the software supplier.” *Id.*

Regarding whether Logan’s first intermediate code is an identifier associated with the licensee, the uniqueness of the identifier will vary depending on the inputs by the user, which may include vendor supplied information and does not require personal information of the user according to the claim construction analysis of record. *See* Ex. 1008, 16–21; Ex. 1009, 11. Logan teaches that the software serial number is unique to each original copy of the software and that it is combined with the hardware serial number input by the user to generate the first intermediate code. Ex. 1003, 4:19–31 (“Each original copy or embodiment of the computer software has a first software code which is uniquely associated with that one particular embodiment”), 4:65–5:30; *see* Pet. 38–39. The preponderance of evidence in this record also supports a serial number supplied by a vendor being unique to the user. Ex. 1046, 27: 1–3 (the inventor of the ’216 patent testifying that “the serial numbers of each piece of software . . . identifies the owner of the software . . .”), 29:1–3 (the inventor of the ’216 patent

testifying that his idea was “linking a serial number to a specific machine”); Ex. 1044, 10 (Patent Owner arguing in district court that it “expressly contemplates information that is not one-of-a-kind”); Ex. 1016, 39 (Patent Owner arguing during reexamination that nexus between the claims and some commercial embodiments was satisfied because they used “a unique serial number . . . that is assigned to each copy of the software”); Ex. 1008, 12 (District Court finding in its claim construction of “unique” that “[t]o construe the word unique to mean no possibility of duplication would simply be inconsistent with the specification.”). In addition, the claim construction analysis explicitly rejected “one-of-a-kind information that describes/identifies a person” when construing “information . . . which uniquely identifies an intended registered user” recited in claim 12. Ex. 1008, 22.

At the oral hearing, Patent Owner conceded that the Federal Circuit held that vendor-provided information “could be” the basis for a licensee unique ID, but argued that whether a particular vendor supplied information is uniquely associated with a licensee is a fact question. Tr. 61–62. In Logan, the software serial number is input by the user together with the hardware serial number to generate the first intermediate code. Therefore, the information inputted by the user is not solely platform related and together with the hardware serial number “uniquely identifies an intended registered user” as recited in claim 12 compared to the software serial number alone. Accordingly, we are persuaded that the first intermediate code meets the recited “security key” requirement of claims 12–14, as that term is construed.

Logan also teaches that the accuracy with which the software serial number is input by the user would be checked by the registration authority because “the software supplier is able to identify the second software code for each particular embodiment of the software by reference to the first software code or serial number.” Ex. 1003, 4:37–40. It is also checked by the registration authority in Logan by preventing full access to the digital data if the second intermediate code does not match the stored hidden number. *Id.* at 5:65–6:7; *see* Pet. 39–40. Regarding whether algorithms are replicated at a registration authority in Logan, Patent Owner does not dispute that the same algorithm is replicated at the registration authority to produce the first intermediate code. We do not read into the claims a temporal limitation requiring that the security key is generated at the same time at the local and remote locations, as argued by Patent Owner. Claim 12 requires that the registration system is replicated at the registration authority for the purpose of check on the information unique to the user. Therefore, we are persuaded that the registration authority disclosed by Logan meets the requirements of claims 12–14.

After considering Petitioner’s and Patent Owner’s positions, as well as their supporting evidence, we determine that Petitioner has shown by a preponderance of the evidence that claims 12–14 would have been anticipated by Logan.

*c. Overview of Grundy*

Grundy is directed to “a computer-based method and apparatus to control the distribution of information . . . whereby a user of computer software becomes the primary agent of manufacture and distribution of the software under the direct monitoring and control of a centralized control

point.” Ex. 1004, 1:7–13. The software is “capable of being operated in two modes.” *Id.* at 4:28–29. “The first mode is a full-function mode, where all the functions and features of the software product are available to the user” and “[t]he second mode is an evaluation mode, where only certain functions, decided by the software developer, can be accessed by the user.” *Id.* at 4:29–34. The software product in evaluation mode is distributed to the user community. *Id.* at 9:3–6. The user “is in fact supplied with a complete copy of the software, but can not operate the software in full-function mode until after the registration process . . . is completed.” *Id.* at 9:14–17. An ownership check is performed “[e]ach time a user starts the software product” in order to determine whether the software should be executed in an “evaluation mode” or “full-function mode”. *Id.* at 5:37–48.

*d. Obviousness of Claims 15 and 16*

Petitioner asserts that, like Logan, Grundy also teaches a system for ensuring that copied software is properly registered (Pet. 41), but differs from Logan in that the software runs in both a demonstration or evaluation mode and a full-function mode depending on an ownership check each time the software is executed (*id.* at 41–42). Petitioner argues that it would have been obvious to one of ordinary skill in the art to combine the option of running software in a demonstration mode, as taught by Grundy, with the software protection system of Logan, because Grundy teaches “the benefit of providing an evaluation mode to a software consumer is to allow the consumer to try and evaluate features of the product prior to making a decision to purchase.” *Id.* at 42 (citing Ex. 1004, 4:27–36, 9:6–11; Ex. 1007 ¶¶ 49–57). Petitioner contends that modifying the software registration system of Logan with the ownership check process of Grundy would have

been within the skill and common sense for a skilled artisan; it would have provided the predictable result of executing the software in two different modes as described by Grundy; and claims 15–16 would have been obvious over the combination. *Id.* at 42–46. Petitioner provides the Madisetti Decl. as evidence to support the obviousness of claims 15 and 16 over the combination of Logan and Grundy. *Id.* (citing Ex. 1007 ¶¶ 52–57).

Patent Owner contends that “[t]here is no disclosure in Grundy’s system to check at the time of boot whether it is a first or subsequent boot.” PO Resp. 47. According to Patent Owner, Grundy teaches away from claim 15 because it “emphasi[zes] an entirely different check—the mode check—at start up.” *Id.* (citing Ex. 2008 ¶ 139). Patent Owner further contends that the requirements of claim 15 are not met because “neither *Logan* nor *Grundy* disclose detecting first or subsequent boots.” *Id.* at 48 (citing Ex. 2008 ¶ 142). Regarding the reason to combine Logan and Grundy, Patent Owner argues that one skilled in the art would have not modified Logan with the demo mode feature of Grundy because “[t]he common sense approach to adding a demonstration mode to a software product in view of Logan and Grundy would have been to abandon Logan and incorporate Grundy alone.” *Id.* at 49 (citing Ex. 2008 ¶ 146).

After considering Petitioner’s and Patent Owner’s positions as to claims 15 and 16, as well as their supporting evidence, we determine that Petitioner has shown by a preponderance of the evidence that those claims are unpatentable under 35 U.S.C. § 103(a) as obvious over Logan and Grundy. Regarding detecting first or subsequent boots as required by claim 15, Grundy discloses the step of determining “if this is the first use,” which indicates a first or subsequent boot, by checking whether an ownership

details record exists every time the software is executed. *See* Ex. 1004, 16:41–49; Ex. 1007 ¶ 52. If no details are present, an initialized ownership record will be created. Ex. 1004, 16:41–49. On a subsequent boot, an ownership record will exist and be checked to determine whether the software should run in evaluation or full-function mode. *Id.* at 5:37–48, 16:18–17:39. Therefore, the preponderance of the evidence supports Petitioner’s argument that claims 15 and 16 would have been obvious over the combined teachings of Logan and Grundy. Petitioner also provides a reason for combining this particular feature of Grundy with the system of Logan, namely, “allow[ing] potential users to try and evaluate features of the software product” as stated by Grundy. Ex. 1004, 4:34–36; Pet. 42. Therefore, Petitioner’s reason to modify the teachings of Logan with the teachings of Grundy is apparent from Grundy and has a rational underpinning. *See KSR*, 550 U.S. at 418.

4. *Haines (Ex. 1005) and Manduley (Ex. 1006)*

Petitioner contends that claims 12–14 would have been obvious under 35 U.S.C. § 103(a) over Haines and Manduley. Pet. 46–53.

*a. Overview of Haines*

Haines discloses a system for reconfiguring postage meters that selectively enable and disable features. Ex. 1005, Abstr. The user’s meter generates a request code number from user-entered information in both systems. *Id.* at 5:1–6:48. Haines discloses that the meter and the data center each use the same encryption routine and input numbers so that the data center can control the feature set of the meter. *Id.* at 4:17–26. The data center computer in Haines checks that its configuration request code matches the configuration request code generated by the user’s meter.

Haines states that if “the agent has improperly entered numbers,” the codes will not match. *Id.* at 7:15–26.

*b. Overview of Manduley*

Manduley also discloses a system that selectively activates and unactivates features in a data processing device, such as a parcel manifest system. Ex. 1006, Abstr., 1:35–39. The user’s meter generates a request code number from user-entered information as in Haines’ system. *Id.* at 5:63–6:50. Manduley expressly discloses location data is entered by the user, such as “zip code or other data identifying the location” of the device. *Id.* at 5:53–6:50. “From that information, the data center determines the identity of the customer holding that device 20 and checks the customer’s file to determine whether the request is appropriate (step 210).” *Id.* at 7:63–8:2.

*c. Obviousness of Claim 12*

Petitioner argues that it would have been obvious to one of ordinary skill in the art to modify the system of Haines with the use of location data as the information that the user inputs because Manduley teaches the benefit of using location data is to allow the data center to determine the customer identity. Pet. 47 (citing Ex. 1006, 7:63–8:2). Petitioner provides as evidence in support of its position the Madisetti Declaration, which characterizes the modification of Haines as a substitution that is contemplated by Haines’s suggestion that “other meter specific identifying information” may be used instead of an ascending register value in generating a request code. Ex. 1007 ¶ 61 (citing Ex. 1005, 5:28–30). Petitioner also argues that the use of a zip code, as in Manduley, is

exemplified as a user input to generate the security key in the '216 patent. Pet. 47 (citing Ex. 1001, Fig. 4).

Patent Owner contends that Haines and Manduley are not analogous art to the '216 patent because “neither reference is directed towards a user who desires to obtain a license, install, and use software on their computer” and, as such, one skilled in the art would not look to these references “to solve the problems that the inventor of the '216 Patent faced.” PO Resp. 52 (citing Ex. 2008 ¶ 152); *id.* at 55. Patent Owner asserts that neither discloses “licensing procedures or a registration system” or “installation of software into a user’s computer” because both teach “select[ing] operating features already installed in a device.” *Id.* at 52 (citing Ex. 2008 ¶ 154); *id.* at 54.

Patent Owner further contends that Haines has not been shown to disclose the “registration system” preamble of claim 12 because “none of the cited passages disclose any licensing procedure”. *Id.* at 56. In addition, Patent Owner asserts that neither Haines nor Manduley discloses the use of information that is uniquely associated with a person who intends to become a licensee as required by independent claim 12. *Id.* at 57. Patent Owner also argues that Petitioner’s reason for combining Haines’s system with the zip code input taught by Manduley is not supported by the record because Manduley identifies the location of the device, not the agent, and “the data center computer in Haines already knows the identity of the agent.” *Id.* In addition, Patent Owner asserts that the “agent” disclosed by Haines and Manduley cannot be the “intended registered user of said software on a computer on which said software is to be installed” as recited in claim 12 because “the owner of the business or home (or a post office) is the user of the meter.” *Id.* at 58 (citing Ex. 2008 ¶ 180). According to Patent Owner,

Petitioner has not identified “an intended registered user of said software on a computer on which said software is to be installed”, “any user’s computer on which software is to be installed”, or “‘licensee,’ ‘licenses,’ or ‘licensing procedures’” as recited in claim 12. *Id.* (citing Ex. 2008 ¶¶ 181, 182).

Based on the trial record, we find the preponderance of the evidence supports Petitioner’s assertion that claim 12 would have been obvious over the combination of Haines and Manduley. As an initial matter, we find Haines and Manduley to be analogous art. The PTO and its reviewing courts have developed and applied a two-step “test” to determine whether a prior art reference is “analogous” art and therefore may be used as evidence with respect to a question of obviousness under § 103. *See In re Oetiker*, 977 F.2d 1443, 1447 (Fed. Cir. 1992); *In re Clay*, 966 F.2d 656, 659 (Fed. Cir. 1992); *In re Wood*, 599 F.2d 1032 (CCPA 1979). Step 1 requires an answer to the following question: “Is the reference within the field of the inventor’s endeavor?” If the answer is “yes,” then the reference is “analogous” and therefore may be used as evidence. If the answer is “no,” then Step 2 requires an answer to the following question: “Is the reference reasonably pertinent to the particular problem the inventor was trying to solve?” If the answer is “yes,” then the reference is analogous and therefore may be used as evidence.

Haines and Manduley are analogous art because, like the ’216 patent, they relate to systems for remotely permitted use of software on a device. Ex. 1007 ¶ 59. The scope of the ’216 patent is not limited to personal computers, but, rather, any “platform” on which software runs. Ex. 1001, 2:52–55 (“In broad terms, the system according to the invention is designed and adapted to allow digital data or software to run in a use mode on a

platform . . .”), 2:24–30 (“[T]he term ‘platform’ denotes an environment to be associated with a computer device such as a microprocessor or other data processing device which permits execution of the digital data . . .”). Even if the Haines and Manduley references could be considered in a different field of endeavor than the ’216 patent because their intended use is postage and parcel systems rather than a personal computer, they are analogous art for the additional reason that they relate to the problem of restricting access to software to those who have a right to use it. Ex. 1001, 2:40–44 (“In this specification, ‘use mode’ refers to use of the digital data or software by its execution on a platform so as to fulfill the seller’s/licensor’s obligations in relation to the sale or license of the right to execute the digital data or software in the use mode.”); Ex. 1005, Abstr., 1:41–49 (“only authorized meter reconfigurations can occur”), 11:13–17; Ex. 1006, Abstr, 1:29–32 (“the marketer may wish to charge the customer a separate amount for each software function or each variety of data and the customer may wish to pick and choose among the functions and/or varieties”), 2:31–63.

Petitioner demonstrates by a preponderance of the evidence that Haines discloses a “registration system” as recited in the preamble of claim 12 and construed in the table above. Pet. 48–49. The reconfiguration process of Haines is a licensing procedure because it is a system that provides security for the authorized use of software features. Ex. 1005, 1:45–47, 4:17–26; *see* Pet. 48–49. Patent Owner argues that Haines does not disclose a registration system “because it does not disclose any licensing procedure that must be followed by a prospective user,” citing paragraph 172 of the DiEuliis Declaration. PO Resp. 56. The argument appears to be that there is no *ipsis verbis* disclosure of the term “license” in Haines

because Haines does disclose a system to protect authorized use. Dr. DiEuliis acknowledged that licensing means you have obtained permission to use something. Ex. 1048, 106:6–8 (“Licensing is, if you buy a — a product, a software product, and you have the license to use it, which means you’re allowed to use it.”). Accordingly, the preponderance of the evidence is that Haines discloses a registration system as recited in claim 12.

Petitioner has shown by a preponderance of the evidence that Haines’s registration system, as modified by Manduley, generates “a security key from information input to said software”, as required by claim 12. Pet. 46–52. Haines’s registration system generates a “request code” from the meter serial number, other meter-specific information, and user-entered information. Ex. 1005, 5:1–6:48. The request code generated in Manduley’s registration system reflects user-entered location data. Ex. 1006, 5:63–6:50 (“zip code or other data identifying the location”); *see* Ex. 1001, Fig. 4 (“ADDRESS,” “CITY,” and “ZIP/POST CODE” listed as examples of personal information inputs), 5:10–12. The preponderance of the evidence shows that one of ordinary skill in the art would have considered it obvious to have modified Haines by substituting information inputs disclosed in Haines for generating the “request code” with Manduley’s input of a zip code “which uniquely identifies an intended registered user of said software on a computer on which said software is to be installed”, as further recited by claim 12, for the reason disclosed in Manduley. Pet. 46–48; Ex. 1007 58–63. Manduley itself discloses the benefit of user-entered location data as allowing the data center to determine the customer’s identity. Ex. 1006, 7:63–8:2 (the location data entered by the user is used to “determine[] the identity of the customer holding that

device”). Dr. Madisetti states that substitution of the inputs used in Haines is contemplated by Haines, particularly with respect to “other meter specific identifying information”. Ex. 1007 ¶ 61 (quoting Ex. 1005, 5:28–30). According to Dr. Madisetti’s testimony, using Manduley’s device zip code, or other location data, instead of Haines’s ascending register value, is nothing more than simple substitution of one known element for another and that the modification would achieve the result described by Manduley without undue experimentation. *Id.* ¶¶ 61–63.

The preponderance of the evidence also shows that Haines’s registration system, as modified by Manduley, is “replicated at a registration authority and used for the purposes of checking by the registration authority that the information unique to the user is correctly entered at the time that the security key is generated by the registration system” as further required by claim 12. Haines explicitly states that the request code “is checked by the data center computer which generates the configuration request code using the same algorithm.” Ex. 1005, Abstr., 1:59–64, 4:17–26.

Patent Owner’s argument regarding the “owner of the business or home” rather than the “agent” in Haines being the “intended registered user” as recited in claim 12 is not persuasive. Manduley explicitly teaches “the user enters data”. Ex. 1006, 6:39–41. Who can be properly characterized as the “intended registered user” of the registration system in Haines and Manduley is not relevant. It is the teachings of the references regarding a registration system that are pertinent to the obviousness analysis of claim 12. Dr. DiEuliis conceded that the person entering information in the registration system of claim 12 is irrelevant. Ex. 1041, 111:17–22 (“There’s no explicit

reference to [ ] any one person entering this [user] information [of claim 12]”).

We also are not persuaded by Patent Owner’s argument regarding the degree of uniqueness that should be accorded a zip code in identifying the agent in Manduley (PO Resp. 57) because the claim construction analysis explicitly rejected “one-of-a-kind information that describes/identifies a person” when construing “information . . . which uniquely identifies an intended registered user” as recited in claim 12. Ex. 1008, 22.

*d. Obviousness of Claim 13*

Claim 13 depends from claim 12 and further requires that “said security key is generated by a registration number algorithm.” Ex. 1001, 50–51. Petitioner has shown that Haines’s registration system explicitly states that a non-linear encryption algorithm is used to generate its request code. Pet. 52; Ex. 1005, 9:66–10:19 (“[T]he configuration request code and the configuration enable code are generated by an encryption routine, stored both in the meter ROM and in the data center computer. The encryption routine is a nonlinear algorithm that generates a number that is apparently random to an outside person.”), 6:43–50.

Patent Owner contends that obviousness of claim 13 has not been shown because the Petition contradicts itself as to whether the “security key” recited in claim 13 is disclosed in Haines. PO Resp. 59 (citing Pet. 50, 52; Ex. 2008 ¶¶ 186–87). We have considered Patent Owner’s argument and are not persuaded. Petitioner argues that one of ordinary skill in the art would have modified the registration system of Haines, particularly the request code generated in Haines, with the location input taught by Manduley. Pet.

46–53. This is clear from the complete statements on pages 50 and 52 of the Petition that Patent Owner cites:

Haines does not expressly disclose that the configuration request code (i.e., security key) is generated from inputted information that is uniquely associated with a person who intends to become a licensee. However, Manduley discloses generation of a request code that reflects user entered location data, such as the “zip code or other data identifying the location” of the device, which is used by the data center to “determine[] the identity of the customer holding that device.”

*Id.* at 50, 52–53. Therefore, the preponderance of the evidence demonstrates that claim 13 would have been obvious in view of Haines and Manduley.

*e. Obviousness of Claim 14*

Claim 14 depends from claim 13 and further requires that the “registration number algorithm combines information entered by a prospective registered user unique to that user with a serial number generated from information provided by the environment in which the software to be protected is to run.” Ex. 1001, 53–56. Petitioner has shown that Haines, as modified by Manduley’s location data, generates a request code, which includes information entered by a prospective register user unique to that user and a meter serial number, using a non-linear encryption algorithm. Pet. 52–53; Ex. 1005, 5:1–6:50, 9:66–10:19; Ex. 1006, 5:39–6:50, 7:63–8:2.

Patent Owner asserts that Petitioner’s argument is flawed because claim 14 depends from claim 13 and the analysis of whether Haines discloses the required “security key” required by claim 13 is contradicted by the analysis of claim 14. PO Resp. 59 (citing Pet. 52–53). Patent Owner

further contends that “Petitioners fail to point to any evidence that either *Haines* or *Manduley* discloses combining information ‘provided by the environment in which the software to be protected is to run,’ as recited in claim 14.” *Id.* at 59–60 (citing Ex. 2008 ¶ 190).

For the reason explained above in connection with claim 13, we are not persuaded that Petitioner failed to show that the combination of *Haines* and *Manduley* discloses the “security key” recited in claim 13. Regarding the “information provided by the environment in which the software to be protected is to be run” recited in claim 14, Petitioner has shown that *Haines* discloses generating a request code from information that includes a meter serial number. Pet. 52, 47; Ex. 1005, 5:1–6:48; Ex. 1007 ¶ 60. Therefore, the preponderance of the evidence demonstrates that claim 14 would have been obvious in view of *Haines* and *Manduley*.

*f. Conclusion*

In sum, the preponderance of the evidence shows that claims 12–14 would have been obvious over *Haines* and *Manduley*.

*D. Conclusion*

Petitioner has demonstrated by a preponderance of the evidence that (1) claims 1–11 and 17–20 of the ’216 patent are anticipated under § 102(e) by Schull; (2) claims 10 and 11 of the ’216 patent are unpatentable under § 103(a) over Schull; (3) claims 12–14 of the ’216 patent are anticipated under § 102(e) by Logan; (4) claims 15 and 16 of the ’216 patent are unpatentable under § 103(a) over the combination of Logan and Grundy; and (5) claims 12–14 of the ’216 patent are unpatentable under § 103(a) over the combination of *Haines* and *Manduley*.

III. ORDER

In consideration of the foregoing, it is  
ORDERED that claims 1–20 of the '216 patent are held to be  
unpatentable; and

FURTHER ORDERED that, because this is a Final Written Decision,  
parties to this proceeding seeking judicial review of our decision must  
comply with the notice and service requirements of 37 C.F.R. § 90.2.

IPR2014-01453  
Patent 5,490,216 C2

PETITIONER:

Eric A. Buresh  
Mark C. Lang  
ERISE IP, P.A.  
eric.buresh@eriseip.com  
mark.lang@eriseip.com

Don Daybell  
James Maune  
ORRICK, HERRINGTON & SUTCLIFFE LLP  
d2dptabdocket@orrick.com  
jmaune@orrick.com

PATENT OWNER:

Sean D. Burdick  
Brett Mangrum  
UNILOC USA, INC.  
sean.burdick@unilocusa.com  
brett@etheridgelaw.com