Roku, Inc. and VIZIO, Inc.

v.

Ancora Techs. Inc.

IPR2021-01406

U.S. Patent No. 6,411,941

Petitioner's Demonstratives

Instituted Grounds

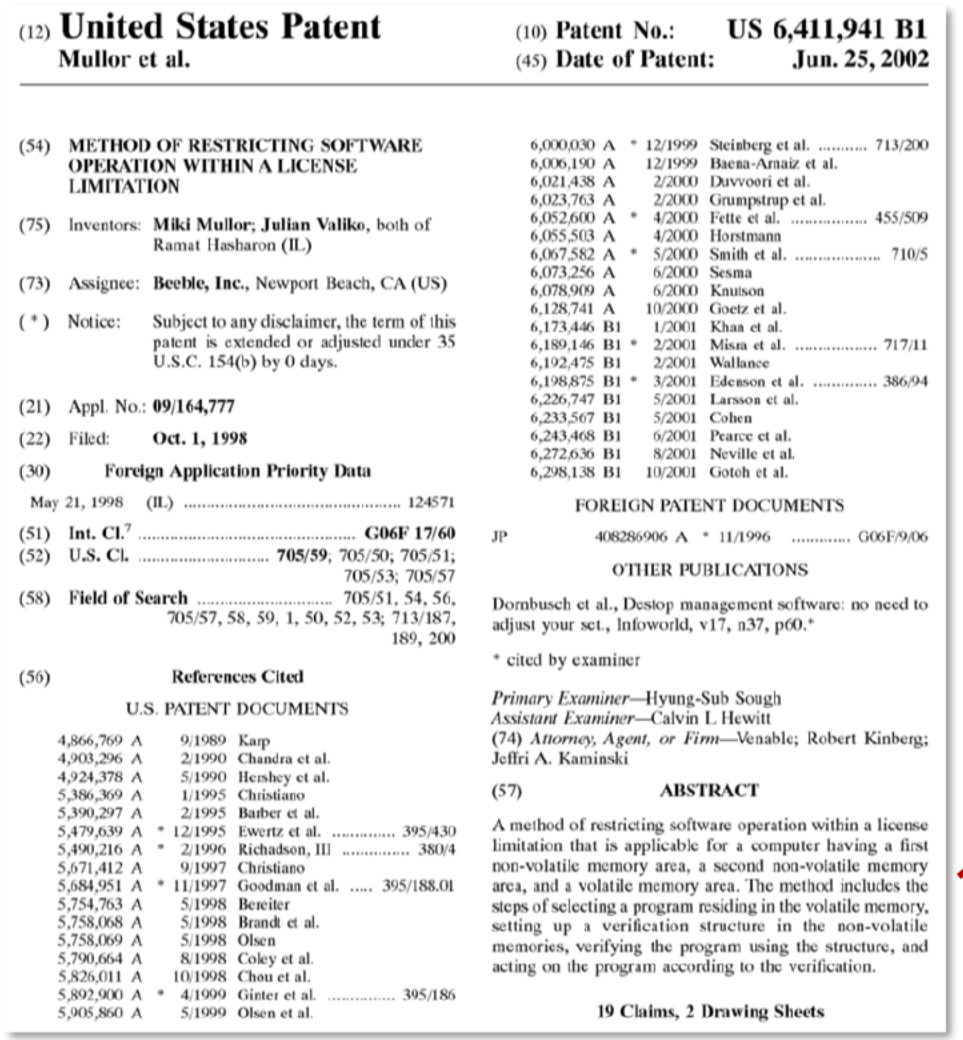
Claims Challenged	35 U.S.C. §	References
1, 2, 11, 13	103(a)	Hellman, Chou
1–3, 6–14, 16	103(a)	Hellman, Chou, Schneck

DI, 6

Disputed Issues

- **Claim Construction**: Whether the term "**agent**" excludes hardware *and* requires an "OS-level software program or routine." (POR, 32-36)
- **Alleged Missing Limitations**: Whether the Hellman-Chou combination renders obvious the step of "using an **agent** to set up a **verification structure** in the erasable, non-volatile memory of the BIOS." (POR, 56-63)
- **Motivation to Combine**: Whether a skilled artisan would have been motivated to combine Hellman and Chou, as proposed in the petition. (POR, 52-56)
- **Dependent Claims**: Whether Petitioner has shown the dependent **encryption-related** claims 3, 8, 9, and 14 to be obvious. (POR, 64-65)
- **Alleged Objective Indicia of Non-obviousness**: Whether Patent Owner's settlement agreements and purported industry praise support its claim of non-obviousness. (POR, 66-70)

The ’941 Patent



(12) **United States Patent**
Mullor et al.

(10) **Patent No.: US 6,411,941 B1**
(45) **Date of Patent: Jun. 25, 2002**

(54) **METHOD OF RESTRICTING SOFTWARE OPERATION WITHIN A LICENSE LIMITATION**

(75) Inventors: **Miki Mullor**; **Julian Valiko**, both of Ramat Hasharon (IL)

(73) Assignee: **Beeble, Inc.**, Newport Beach, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/164,777**

(22) Filed: **Oct. 1, 1998**

(30) **Foreign Application Priority Data**

May 21, 1998 (IL) 124571

(51) **Int. Cl.**[7] **G06F 17/60**

(52) **U.S. Cl.** **705/59**; 705/50; 705/51; 705/53; 705/57

(58) **Field of Search** 705/51, 54, 56, 705/57, 58, 59, 1, 50, 52, 53; 713/187, 189, 200

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,866,769	A		9/1989	Karp
4,903,296	A		2/1990	Chandra et al.
4,924,378	A		5/1990	Hershey et al.
5,386,369	A		1/1995	Christiano
5,390,297	A		2/1995	Barber et al.
5,479,639	A	*	12/1995	Ewertz et al. 395/430
5,490,216	A	*	2/1996	Richadson, III 380/4
5,671,412	A		9/1997	Christiano
5,684,951	A	*	11/1997	Goodman et al. 395/188.01
5,754,763	A		5/1998	Bereiter
5,758,068	A		5/1998	Brandt et al.
5,758,069	A		5/1998	Olsen
5,790,664	A		8/1998	Coley et al.
5,826,011	A		10/1998	Chou et al.
5,892,900	A	*	4/1999	Ginter et al. 395/186
5,905,860	A		5/1999	Olsen et al.
6,000,030	A	*	12/1999	Steinberg et al. 713/200
6,006,190	A		12/1999	Baena-Arnaiz et al.
6,021,438	A		2/2000	Duvvoori et al.
6,023,763	A		2/2000	Grumpstrup et al.
6,052,600	A	*	4/2000	Fette et al. 455/509
6,055,503	A		4/2000	Horstmann
6,067,582	A	*	5/2000	Smith et al. 710/5
6,073,256	A		6/2000	Sesma
6,078,909	A		6/2000	Knutson
6,128,741	A		10/2000	Goetz et al.
6,173,446	B1		1/2001	Khan et al.
6,189,146	B1	*	2/2001	Misra et al. 717/11
6,192,475	B1		2/2001	Wallance
6,198,875	B1	*	3/2001	Edenson et al. 386/94
6,226,747	B1		5/2001	Larsson et al.
6,233,567	B1		5/2001	Cohen
6,243,468	B1		6/2001	Pearce et al.
6,272,636	B1		8/2001	Neville et al.
6,298,138	B1		10/2001	Gotoh et al.

FOREIGN PATENT DOCUMENTS

JP	408286906	A	* 11/1996	..... G06F/9/06

OTHER PUBLICATIONS

Dornbusch et al., Destop management software: no need to adjust your set., Infoworld, v17, n37, p60.*

* cited by examiner

Primary Examiner—Hyung-Sub Sough
Assistant Examiner—Calvin L Hewitt
(74) *Attorney, Agent, or Firm*—Venable; Robert Kinberg; Jeffri A. Kaminski

(57) **ABSTRACT**

A method of restricting software operation within a license limitation that is applicable for a computer having a first non-volatile memory area, a second non-volatile memory area, and a volatile memory area. The method includes the steps of selecting a program residing in the volatile memory, setting up a verification structure in the non-volatile memories, verifying the program using the structure, and acting on the program according to the verification.

19 Claims, 2 Drawing Sheets

(57) **ABSTRACT**

A method of restricting software operation within a license limitation that is applicable for a computer having a first non-volatile memory area, a second non-volatile memory area, and a volatile memory area. The method includes the steps of selecting a program residing in the volatile memory, setting up a verification structure in the non-volatile memories, verifying the program using the structure, and acting on the program according to the verification.

EX1001, Abstract

’941 Patent – Independent Claim 1

1. A method of restricting software operation within a license for use with a computer including an erasable, non-volatile memory area of a BIOS of the computer, and a volatile memory area; the method comprising the steps of:

selecting a program residing in the volatile memory,

using an agent to set up a verification structure in the erasable, non-volatile memory of the BIOS, the verification structure accommodating data that includes at least one license record.

verifying the program using at least the verification structure from the erasable non-volatile memory of the BIOS, and

acting on the program according to the verification.

EX1001, 6:59-7:4

The ’941 Patent – Summary of Invention

SUMMARY OF THE INVENTION

The present invention relates to a method of restricting software operation within a license limitation. This method strongly relies on the use of a key and of a record, which have been written into the non-volatile memory of a computer.

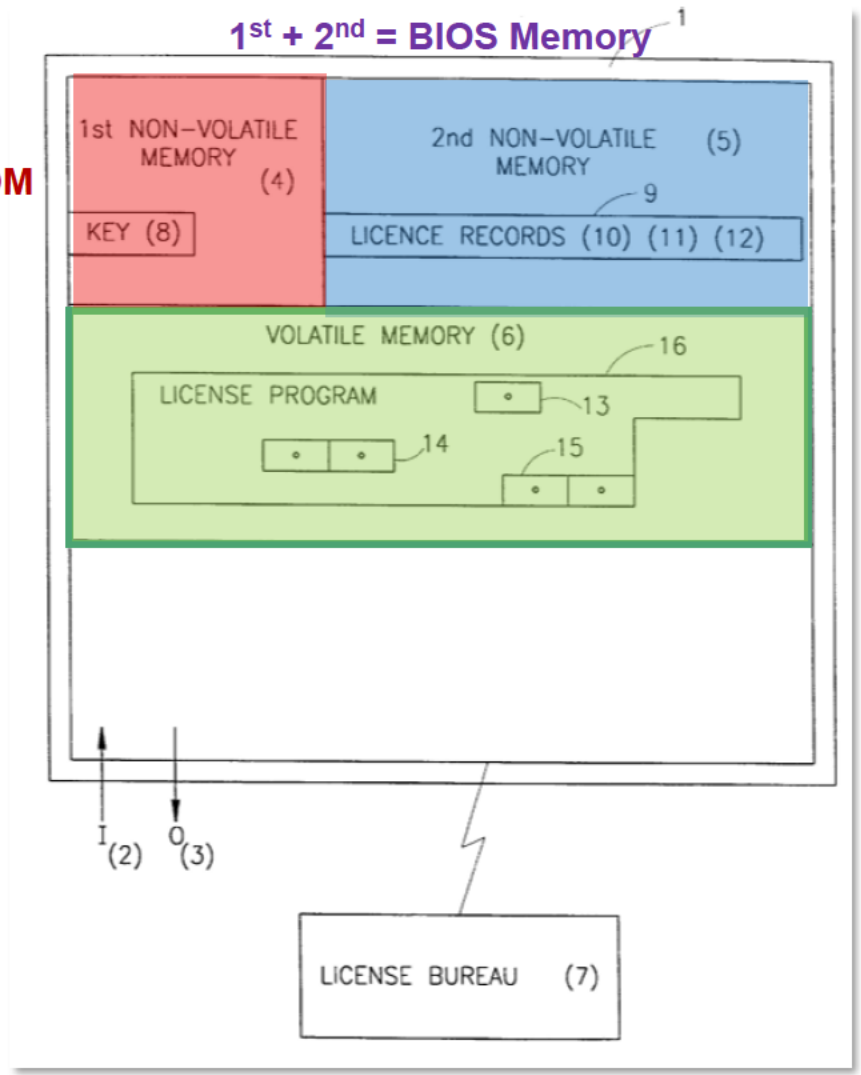
Pet. 11; EX1001, 1:37-43

Now, there commences an initial license establishment procedure, where a verification structure is set in the BIOS so as to indicate that the specified program is licensed to run on the specified computer. This is implemented by encrypting the license record (or portion thereof) using said key (or portion thereof) exclusively or in conjunction with other identification information) as an encryption key. The resulting encrypted license record is stored in another (second) non-volatile section of the BIOS, e.g. E^2PROM (or the

ROM). It should be noted that unlike the first non-volatile section, the data in the second non-volatile memory may optionally be erased or modified (using E^2PROM manipulation commands), so as to enable to add, modify or remove licenses. The actual format of the license may include a string of terms that correspond to a license registration entry (e.g. lookup table entry or entries) at a license registration bureau (which will be further described as part of the preferred embodiment of the present invention).

EX1001, 1:59-2:9; Pet, 11-12; POR, 35

The ’941 Patent - Structure



Pet. 14; EX1001, FIG. 1

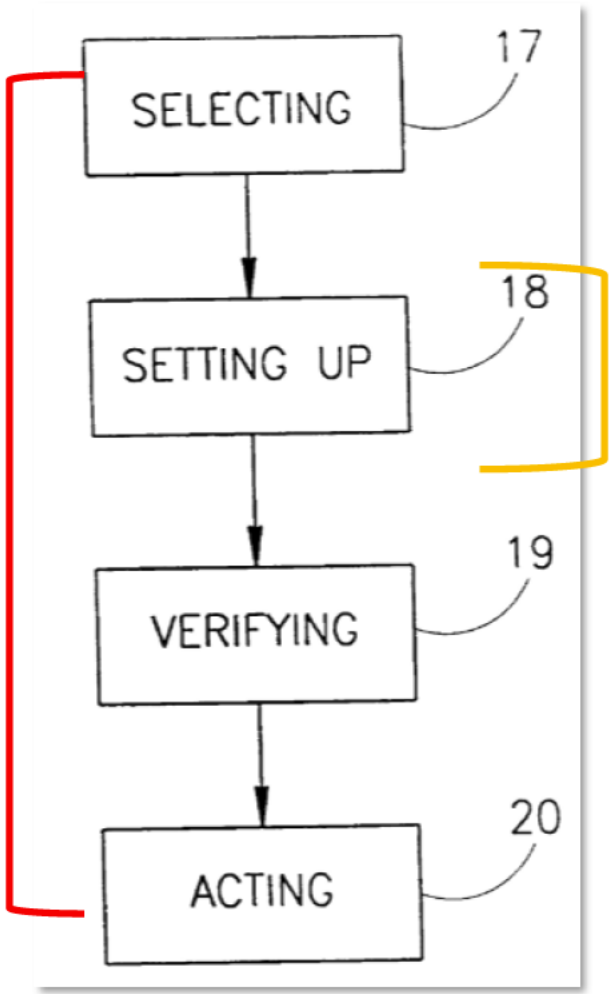
DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

A schematic diagram of a computer and a license bureau is shown in FIG. **1**. Thus, a computer processor (**1**) is associated with input operations (**2**) and with output operations (**3**). This computer (processor) internally contains a first non-volatile memory area (**4**) (e.g. the ROM section of the BIOS), a second non-volatile memory area (**5**) (e.g. the E^2PROM section of the BIOS), and a volatile memory area (**6**) (e.g. the internal RAM memory of the computer).

The second non-volatile memory includes a license-record-area (**9**) e.g. which contains at least one encrypted license-record (e.g. three records **10–12**). The volatile memory accommodates a license program (**16**) having license record fields (**13–15**) appended thereto. By way of example said fields stand for Application names (e.g. Lotus 123), Vendor name (Lotus inc.), and number of licensed copies (1 for stand alone usage, >1 for number of licensed users for a network application).

Pet. 13; EX1001, 5:9-34

The ’941 Patent - Method



Verifying Software

Using an “agent”

Pet. 15; EX1001, FIG. 2

using an agent to set up a verification structure in the erasable, non-volatile memory of the BIOS, the verification structure accommodating data that includes at least one license record,

EX1001, Claim 1

Setting up (**18**) the verification structure includes the steps of: establishing or certifying the existence of a pseudo-unique key in the first non-volatile memory area; and establishing at least one license-record location in the first or the second nonvolatile memory area.

Establishing a license-record includes the steps of: forming a license-record by encrypting of the contents used to form a license-record with other predetermined data contents, using the key; and establishing the encrypted license-record in one of the at least one established license-record locations (e.g. **10–12** in FIG. **1**).

Pet. 15; EX1001, 6:17-27

CLAIM CONSTRUCTION

Claim Construction

"agent"	
Patent Owner's Construction	**Petitioner's Construction**
"agent" excludes hardware and requires an "OS-level software program or routine" *POR 32-36*	"agent" should be given its plain and ordinary meaning, which can be software, hardware, or a combination thereof *Reply 1-7*

- The Board preliminarily (and correctly) rejected Patent Owner's narrow construction in the institution decision. *DI 10, 23.*
- The Board should (again) reject Patent Owner's attempt to rewrite a claim term that appears nowhere in the '941 patent specification. Patent Owner has not met the high burden to show prosecution history disclaimer.

’941 Patent Claims

- “It is a bedrock principle of patent law that the claims of a patent define the invention to which the patentee is entitled the right to exclude.” *Phillips v. AWH Corp*., 415 F.3d 1303, 1312 (Fed. Cir. 2005) (cleaned up).
- The claims themselves put no limitations on the term “agent.”
- Structurally, claim 1’s “agent” is only used in the “setting up” step.

’941 Patent – Independent Claim 1

1. A method of restricting software operation within a license for use with a computer including an erasable, non-volatile memory area of a BIOS of the computer, and a volatile memory area; the method comprising the steps of:

selecting a program residing in the volatile memory,

using an agent to set up a verification structure in the erasable, non-volatile memory of the BIOS, the verification structure accommodating data that includes at least one license record.

verifying the program using at least the verification structure from the erasable non-volatile memory of the BIOS, and

acting on the program according to the verification.

EX1001, 6:59-7:4

’941 Patent Specification

- “The specification is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1315 (Fed. Cir. 2005) (cleaned up).
- The ’941 patent specification does not use the term “agent.” It was added to the claims during prosecution. Reply, 2.
- The specification also does not use the term “operating system” or “OS-level software.” Reply, 2.
- The only evidence Patent Owner relies on in the ’941 patent specification is disclosure of E2PROM manipulation commands, (POR, 35; Sur-Reply,10), which purportedly describes OS-level activity.

’941 Patent Specification

Now, there commences an initial license establishment procedure, where a verification structure is set in the BIOS so as to indicate that the specified program is licensed to run on the specified computer. This is implemented by encrypting the license record (or portion thereof) using said key (or portion thereof) exclusively or in conjunction with other identification information) as an encryption key. The resulting encrypted license record is stored in another (second) non-volatile section of the BIOS, e.g. E^2PROM (or the

ROM). It should be noted that unlike the first non-volatile section, the data in the second non-volatile memory may optionally be erased or modified (using E^2PROM manipulation commands), so as to enable to add, modify or remove licenses. The actual format of the license may include a string of terms that correspond to a license registration entry (e.g. lookup table entry or entries) at a license registration bureau (which will be further described as part of the preferred embodiment of the present invention).

POR, 35; EX1001, 1:59-2:9

'941 Prosecution History

- Patent Owner relies **primarily** on a theory of prosecution history disclaimer. *Sur-Reply at 6.*
  - Prosecution history disclaimers require, in the Federal Circuit's words, "clear and unequivocal evidence that the claimed invention includes or does not include a particular feature." *Poly-America, L.P. v. API Industries, Inc.*, 839 F.3d 1131, 1136 (Fed. Cir. 2016) (citations omitted).
  - "Ambiguous language cannot support disavowal." *Id.*
- Nowhere in its briefing does Patent Owner actually acknowledge the high bar the Federal Circuit has set for prosecution history disclaimer.
- Its arguments fall well short of that high bar since the fairest reading of the prosecution history puts no limits on the term "agent."

Prosecution History – Chron.

- **Examiner rejects claim 1 under Sec. 112 for failure to recite a separate entity that performs the setting up step.** (Reply, 3)

5. Claim 1 is rejected under 35 U.S.C. 112, first paragraph, as based on a disclosure which is not enabling. A device to write to an EEPROM and a method taking into account said device are critical or essential to the practice of the invention, but not included in the claim(s) is not enabled by the disclosure. See *In re Mayhew*, 527 F.2d 1229, 188 USPQ 356 (CCPA 1976). The Applicants do not teach the device necessary to edit an EEPROM nor have they made it clear to the Examiner how their system would be implemented in light of the non-trivial processing required to write and erase its data.

EX1002, 117

Prosecution History – Chron.

- **Claim amended in response to a 112 rejection (Reply, 3-4; EX1002, 137)**

1. (Twice Amended) A method of restricting software operation within a license for use with a computer including an ~~first, non-erasable, non-volatile memory area, a second, non-~~erasable, non-volatile memory area of a (BIOS) of the computer, and a volatile memory area; ~~the first non-volatile memory accomodates data that includes unique key;~~ the method comprising the steps of:

selecting a program residing in the volatile memory,

using an agent to ~~setting~~ up verification structure in the ~~second~~ erasable, non-volatile memory of the BIOS, the ~~verfication~~ verification structure accommodatinges data that includes at least one license record,

verifying the program using at least ~~said~~ the verification structure from the erasable non-volatile memory of the BIOS, and

acting on the program according to the verification.

Applicant's representative appreciates the Examiner's courtesy in conducting a personnel interview in this case. The claims have been amended as agreed upon during the interview and it is respectfully submitted that this application is now in condition for allowance.

Specifically, claim 1 has been amended to recite that the verification structure is stored in an erasable, non-volatile memory area of the BIOS. This claim amendment overcomes the rejections under 35 U.S.C. 112, first paragraph in sections 3, 4 and 5 of the Final Office Action, as well as the rejection under 35 U.S.C. 112, second paragraph in section 7 of the Final Office Action.

- **Amendment adds "agent" as the entity responsible for setting up the verification structure—i.e., writing to the EEPROM. (Reply, 4; EX1002, 135)**
- **Applicant introduces "a description of a specific embodiment of the invention" in the form of the Beeble White paper. (EX1002, 136)**

DELIBERATELY BLANK

Prosecution History – Chron.

- **Post amendment, claims rejected over Misra, Goldman, and Ewertz.** **EX1002, 187.**
- **Applicant Responds:**

Claims 1-23 have been rejected under 35 U.S.C. 103(a) as being unpatentable over Misra et al. in view of U.S. Patent No. 5,684,951 to Goldman et al. and U.S. Patent No. 5,479,639 Ewertz et al.

The cited references do not render the present invention obvious as they do not teach or suggest, among other things, storing a verification structure, such as a software license information, in the BIOS of a computer as is recited in the present claims. EX1002, 197.

Additionally, Misra teaches away from using the BIOS as a storage area by making a statement about client computers that do not have a persistent non-volatile area.

> "The license cache 136 is kept in persisted (non-volatile) storage. Clients that do not have persistent storage can be issued licenses as long as they can generate a unique client ID and can respond to the client platform challenge protocol" (Misra, Col. 12, lines 15-18)

Since all computers must have a BIOS, it is clear Misra teaches away from using the BIOS as a local storage area for licenses. *EX1002, 201*

Furthermore, there is no suggestion or motivation to combine Misra and Ewertz in the manner suggested in the Office Action. BIOS is a configuration utility. Software license management applications, such as the one of the present invention, are operating system (OS) level programs. Therefore, BIOS programs and software licensing management applications do not ordinarily interact or communicate because when BIOS is running, the computer is in a configuration mode, hence OS is not running. Thus, BIOS and OS level programs are normally mutually exclusive.

Ewertz teaches that writing to the BIOS area is performed by the BIOS routines:

> "Referring to Fig. 8, processing logic for updating the flash memory device with configuration data, such as EISA information, is illustrated… The processing logic shown in Fig. 8 resides in the system BIOS of the preferred embodiment" Col 10, lines 20-28

Misra teaches a licensing system that is OS level based:

> "The license generator 26, license server 28 and intermediate server 32 are preferably implemented as computer servers, such as Windows NT servers that run Windows NT server operating systems from Microsoft corporation or UNIX-based servers" Col 5, lines 3-7

Thus, the systems described in Misra and Ewertz are an OS program and a BIOS program, respectively, that cannot run at the same time. Therefore, there is no teaching or suggestion to combine these programs. In fact such a combination would change the operation *EX1002, 199*

Prosecution History – Chron.

- **Focus on setting up the verification structure *in BIOS***
- **No mention of "agent"**
- **No restrictions on "agent"**

Moreover, the present invention proceeds against conventional wisdom in the art. Using BIOS to store application data such as that stored in Misra's local cache for licenses is not obvious. The BIOS area is not considered a storage area for computer applications. An ordinary skilled artisan would not consider the BIOS as a storage medium to preserve application data for at least two reasons.

First, OS does not support this functionality and is not recognized as a hardware device like other peripherals. Every OS provides a set of application program interfaces (APIs) for applications to access storage devices such as hard drives, removable devices, etc. An ordinary person skilled in the art makes use of OS features to write date to storage mediums. There is no OS support whatsoever to write data to the system BIOS. Therefore, an ordinary person skilled in the art would not consider the BIOS as a possible storage medium. Furthermore, it is common that all peripheral devices in the PC are listed and recognized by the OS except for the BIOS. This supports the fact that the BIOS is not considered a peripheral device. Accordingly, an ordinary person skilled in the art would not consider the BIOS for any operation, including writing to the BIOS.

Second, no file system is associated with the BIOS. Every writable device connected to the PC is associated with an OS file system to arrange and manage data structures. An example for such a file system would be FAT, FAT32, NTFS, HPFS, etc. that suggests writing data to the writable device. No such file system is associated with the BIOS. This is further evidence that OS level application programmers would not consider the BIOS as a storage medium for license data.

Pet. 16-17; Reply, 4; EX1002, 200.

Prosecution History – Chron.

- **Examiner allows claims:**

"[R]emarks in the examiner's statement of reasons for allowance" are "insufficient to limit claim scope." *Ancora Techs., Inc. v. Apple, Inc.*, 744 F.3d 732, 737 (Fed. Cir. 2014) (cleaned up).

licensing numbers. Hence, it appears initially, that to one of ordinary skill of the art, the combination of Ewertz et al. with either Ginter et al. and/or Misra et al., would render the present invention obvious. However, the key distinction between the present invention and the closest prior art, is that the Misra et al., and Ginter et al. systems and the Ewertz et al. system run at the operating system level and BIOS level, respectively. More specifically, the closest prior art systems, singly or collectively, do not teach licensed programs running at the OS level interacting with a program verification structure stored in the BIOS to verify the program using the verification structure and having a user act on the program according to the verification. Further, it is well known to those of ordinary skill of the art that a computer BIOS is not setup to manage a software license verification structure. The present invention overcomes this difficulty by using an agent to set up a verification structure in the erasable, non-volatile memory of the BIOS.

Reply, 6-7; EX1002, 213

Key Takeaways

- Applicant/Patent Owner all along emphasized that the crux of the claimed invention was storing a license record in BIOS. *Pet. 16-17; Reply 4-5; EX1002, 197-201; EX1033, ¶12.*
  - POPR: "[s]toring the verification structure in BIOS memory was a 'key distinction'" over the art, and that the BIOS limitation "was significant to the 941 Patent's innovation." Reply, 4; POPR, 35-36.
  - Inventor: Setting up verification structure in BIOS was "the key highlight of this technology." Reply, 4-5; EX1034, 74:4-75:16.
  - Courts: "In sum, the prosecution history demonstrates that the focus of the claims is that the verification structure is in the erasable portion of the non-volatile memory and uses the key in the separate non-erasable portion." Reply, 5; EX1020, 18.
- "Agent" was added to overcome a § 112 rejection and to recite a separate entity for performing the claimed setting up step. *Reply 3-4; EX1002, 116-17, 135, 137; EX1033, ¶13.*
- Applicant never mentioned the significance of the agent at all, let alone distinguished prior art on the basis that it lacks an agent. *Reply 4-6.*
- Examiner's use of "agent" does not restrict the term – it is whatever is used to interface with non-volatile memory to set up a verification structure.

ALLEGED MISSING LIMITATIONS

"using an agent to set up a verification structure"

- "**Agent**"
  - Patent Owner's alleged missing-limitation argument rises and falls with its too narrow construction of "agent" as excluding hardware only and hardware-software implementations.
  - With no alternative arguments, there is no dispute that Hellman discloses an "agent" under the plain and ordinary meaning of that term, which allows hardware and hardware-software implementations.
  - In any event, the prior art combinations render obvious a software-only agent.
- "**Verification Structure**"
  - Hellman's memory structure (i.e., table) of M values defined by H values is the claimed "verification structure."
  - Patent Owner's contrary arguments are based on an implicit construction that a verification structure must be some (unspecified) specific type of structure.

Prior Art – Hellman

However, there is no copy protection, so that one dishonest customer can make as may copies as he wants of the regular version and give or sell them to acquaintances with similar base units (computers). These acquaintances can in turn give or sell generation copies to their acquaintances, etc.

EX1004, 2:24-29; Pet. 22; Reply, 10.

The user at base unit **12** obtains software package **17** by purchasing it at a store, over telephone line, or in some similar manner. The cost for software package **17** can be set low because additional revenue will be obtained by the software manufacturer when issuing additional authorizations for use of the software package.

EX1004, 5:51-56; Pet. 23.

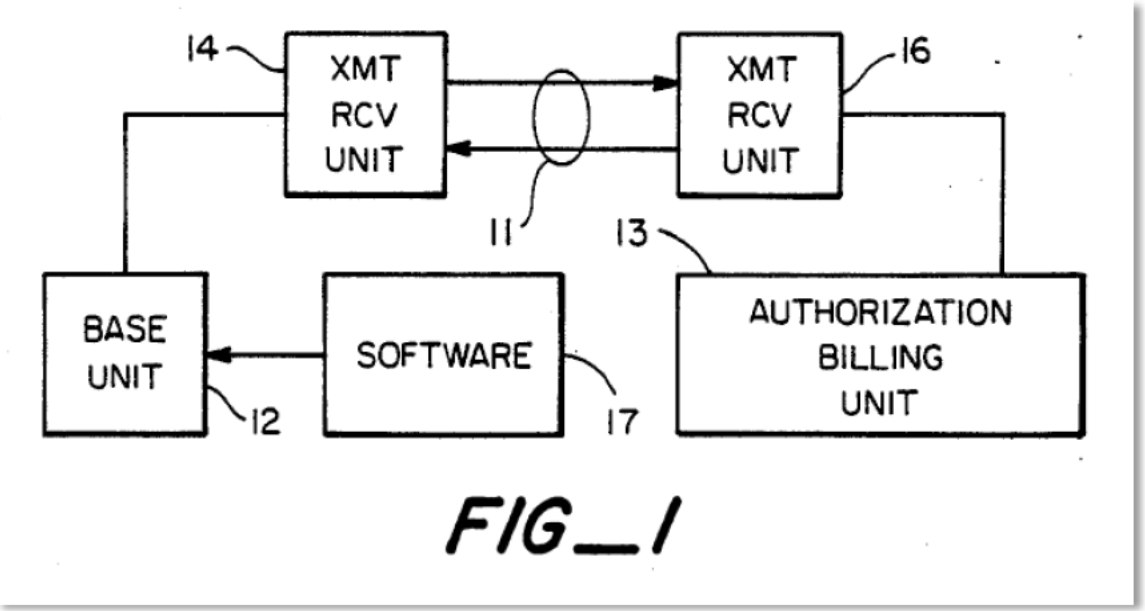


FIG_1

EX1004, FIG. 1; Pet. 23.

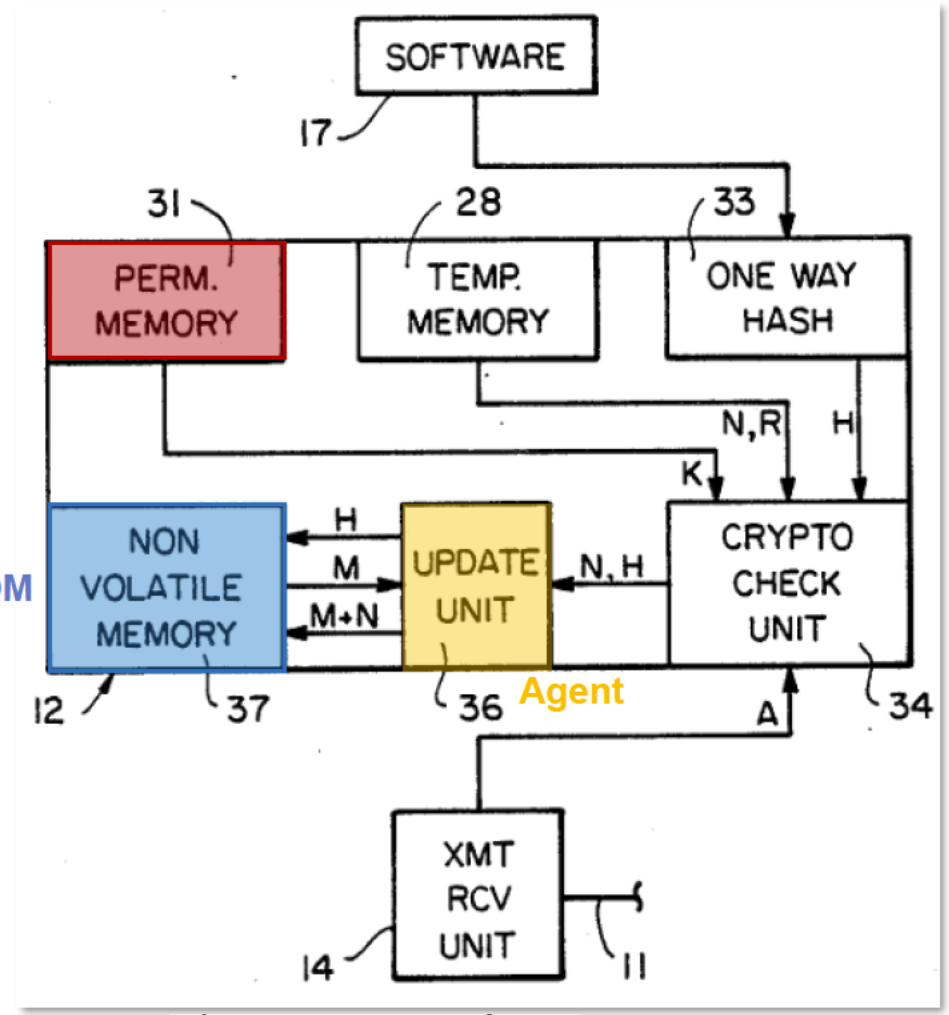
Prior Art – Hellman

software being used. Update unit **36** applies to interrogatory signal representing **H** to a non-volatile memory **37**, for example an EEPROM or a CMOS memory with battery backup. The non-volatile memory **37** applies a signal to the update unit **36**, said signal representing **M**, the number of authorized uses of the software package with hash value **H** which still remain unused prior to this new authorization. The update unit **36** adds **M** and **N** and applies a signal representing **M+N** to the non-volatile memory **37**, so that **M+N** replaces the old number **M** in the non-volatile memory **37** as the number of uses of the software package which have been paid for.

Pet. 25; EX1004, 10:1-13.

package is being used. Update unit **36** uses **H** as an address to non-volatile memory **37**, which responds with a signal representing **M**, the number of uses of software package **17** which are still available.

Pet. 26; EX1004, 10:40-43



EX1004, FIGs. 1, 6 (excerpted)

Prior Art – Hellman

FIG. 8 depicts an implemenation of the base unit 12 during use of a software package. Software package 17 is connected to the base unit 12 and a signal representing said software package is operated on by the one-way hash function generator 33 to produce an output signal which represents the hash value H. The signal H is transmitted to update unit 36 to indicate which software package is being used. Update unit 36 uses H as an address to non-volatile memory 37, which responds with a signal representing M, the number of uses of software package 17 which are still available.

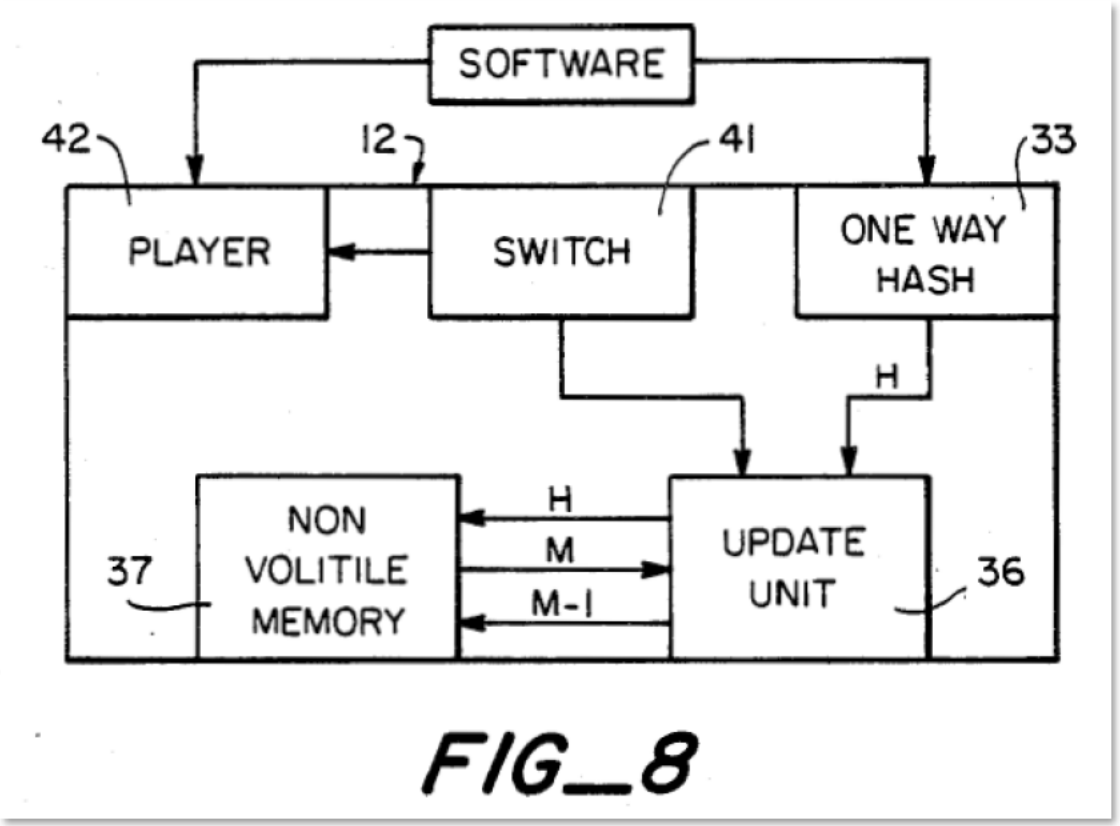
Pet. 26; EX1004, 10:40-43

Software player 42 will vary from application to application. For example, if the software is recorded music then software player 42 would be a record player;

11

if the software is a computer program, then software player 42 would be a microprocessor or central processing unit (CPU).

Pet. 33-34, 37; EX1003, 99; EX1004, 10:66-11:3



EX1004, FIG. 8

Hellman Renders Obvious a Software "Agent"

- "[U]pdate unit would have been implemented by a software routine, potentially along with a hardware module." *Pet. 39; EX1003, ¶137; Reply, 9.*

- Hellman's disclosed verification process is, without modification, suitable for software implementation. *EX1003, ¶¶137-137B; EX1033, ¶18; Reply, 9.*

- A software implementation would allow the provider of the base unit and authorization and billing unit "to change the implementation logic" of the units over time, "without having to physically disassemble, modify, and reassemble" them. *Pet. 39; Reply 9; EX1003, ¶¶ 137B, 138B; EX2026, 34:17-19, 35:9-18; EX1033, ¶¶ 17-18; DI, 23-24.*

Hellman Renders Obvious a Software "Agent"

- Claims do not preclude hardware from working with software to set up the claimed verification structure. *Reply 10; EX1033, ¶ 26; EX1035, 122:12-123:10, 129:9-130:22, 131:14-19.*
- Both experts agree:

Dr. Martin: Q. Can other components or software be used along with the agent to set up the verification structure?
MR. GOSSE: Object to the form.
THE WITNESS: The way I read this limitation in Claim 1 of using an agent to set up a verification structure, it requires the use of the agent to set up the verification structure as described in the limitations. But it does not exclude the possibility of using additional other entities or operations in service of using an agent to set up a verification structure.
EX1035, 129:9-130:22, see also 122:12-123:10, 131:14-19.

Dr. Wolfe: "[E]ven if the claimed agent is limited to software, nothing in the claims precludes the software from working with hardware to set up the claimed verification structure."
EX1033, ¶26.

Hellman Renders Obvious a Software "Agent"

- Dr. Wolfe testified that a POSA would have known how to address any security risks that a software-implementation might create. *Reply 11; EX1033, ¶¶ 22-23.* Patent Owner did not depose Dr. Wolfe and this technical point is unrebutted.
  - "[W]e do not ignore the modifications that one skilled in the art would make to a device borrowed from the prior art." *In re ICON Health & Fitness, Inc.*, 496 F.3d 1374, 1382 (Fed. Cir. 2007).
- That Hellman's alleged hardware implementation may also be effective does not negate Dr. Wolfe's rationale for using a software implementation. *Reply 11.*
  - "The normal desire of artisans to improve upon what is already generally known can provide the motivation to optimize variables such as the percentage of a known polymer for use in a known device." *In re Ethicon, Inc.*, 844 F.3d 1344, 1351 (Fed. Cir. 2017).
- Dr. Wolfe unambiguously opined as to what a POSA "would do," not what she "could do." *Reply 11-12; EX1003, ¶¶ 137, 137A, 138-138B; EX2026, 34:17-19; EX1033, ¶¶ 24-25.*

> 137A. A POSA would have recognized that the update unit 36 would have been implemented by software, hardware, or some combination of the two.

EX1003, ¶ 137A

Hellman Renders Obvious an OS-Level Software "Agent" Under Patent Owner's Construction

- Dr. Martin's criteria for determining whether a program operates at the OS level (Reply, 13):
  - OS-level software "relates to programs that are running that use the running operating system services, as part of their operation." *EX1035, 100:8-22.*
  - "OS-level software can be thought of as running through the operating system." *Id. 101:19-102:4.*
  - OS-level software "rel[ies] on operating system services and is doing so after the operating system is running." Id., 102:5-9, 105:4-10.
  - "[T]here is, in that sense, a moment of transition when the operating system first starts running, its services become available, and then an application could rely on those services. When they do so, they're relying on the OS level services, not the BIOS configuration utility … ." *Id. 103:9-104:2.*

Hellman Renders Obvious an OS-Level Software "Agent" Under Patent Owner's Construction

- Hellman's base unit is a computer and computers in the early 1980s and throughout the 1990s used operating systems. *Reply 12-13; EX2026, 31:21-23; 34:1-2; EX1033, ¶¶ 28-29; EX2018, ¶ 73; EX1035, 99:17-100:1, 109:9-17.*

Dr. Wolfe

28. As I explained above, a POSA would have understood that Hellman's system uses a computer as the base unit (which includes the update unit). *Supra* ¶19. And it was well understood at the time of the alleged invention that computers used operating systems. Indeed, I made clear during my deposition that "[a] general purpose desktop computer, like an ordinary PC, would usually have an operating system." EX2026, 34:1-2; *see also id.*, 31:21-32:23 (noting that Hellman "talks about a computer," and that a POSA "would assume that a computer has an operating system"), 33:16-18 ("If you bought a computer, a desktop computer, for home use, that would almost always have an operating system.").

EX1033, ¶ 28

Dr. Martin

Q. You would agree that operating systems were well known as of 1998, right?

A. Yes, I do.

MR. GOSSE: Object to form.

THE WITNESS: I do agree that operating systems generally were well known to people of ordinary skill in the art in 1998.

* * *

Q. Dr. Martin, would you agree that OS level programs, as you understand that term, were known in the art as of 1998?

A. Yes, I do agree that in 1998, it was known that programs could run that would rely on operating system services that are available because the operating system has been started. And so in that sense, yes, OS level programs were known at that time.

EX1035, 99:17-100:1, 109:9-17

Hellman Renders Obvious an OS-Level Software "Agent" Under Patent Owner's Construction

- Patent Owner's own arguments support Petitioner's position. *Reply 13-14; POR 35; EX2018, 128; EX1033, ¶32; EX1035, 148:6-22, 152:8-153:22, 156:7-157:1, 163:18-164:2, 174:20-176:9, 178:21-179:14; EX1033, 33-41; EX1029, 5.*

'941 Patent

identification information) as an encryption key. The resulting encrypted license record is stored in another (second) non-volatile section of the BIOS, e.g. E^2PROM (or the ROM). It should be noted that unlike the first non-volatile section, the data in the second non-volatile memory may optionally be erased or modified (using E^2PROM manipulation commands), so as to enable to add, modify or remove licenses. The actual format of the license may include a string of terms that correspond to a license registration entry (e.g. lookup table entry or entries) at a license registration bureau (which will be further described as part of the preferred embodiment of the present invention).

EX1001, 1:65-2:9 (cited by POR 35)

Hellman

software being used. Update unit **36** applies to interrogatory signal representing H to a non-volatile memory **37**, for example an EEPROM or a CMOS memory with battery backup. The non-volatile memory **37** applies a signal to the update unit **36**, said signal representing M, the number of authorized uses of the software package with hash value H which still remain unused prior to this new authorization. The update unit **36** adds M and N and applies a signal representing M+N to the non-volatile memory **37**, so that M+N replaces the old number M in the non-volatile memory **37** as the number of uses of the software package which have been paid for.

EX1004, 10:1-13 (cited by Reply 13-14; EX1033, ¶33)

Hellman Discloses a "Verification Structure"

- Any data structure established to accommodate a license record qualifies as a "verification structure." *Reply 15; EX1033, ¶¶44-46; DI, 25; POPR, 32; EX1001, 6:17-21.*

- The claimed "verification structure" is Hellman's memory structure (e.g., a table of M values) defined by hash values (H). *Pet. 37-39; Reply 14-15; EX1004, 10:38-49; EX1003, ¶¶135-36; EX1033, 42-47; EX1026, 30:1-22.*

Memory Address	M Value
Address Defined by (H1)	M1
Address Defined by (H2)	M2
Address Defined by (H3)	M3

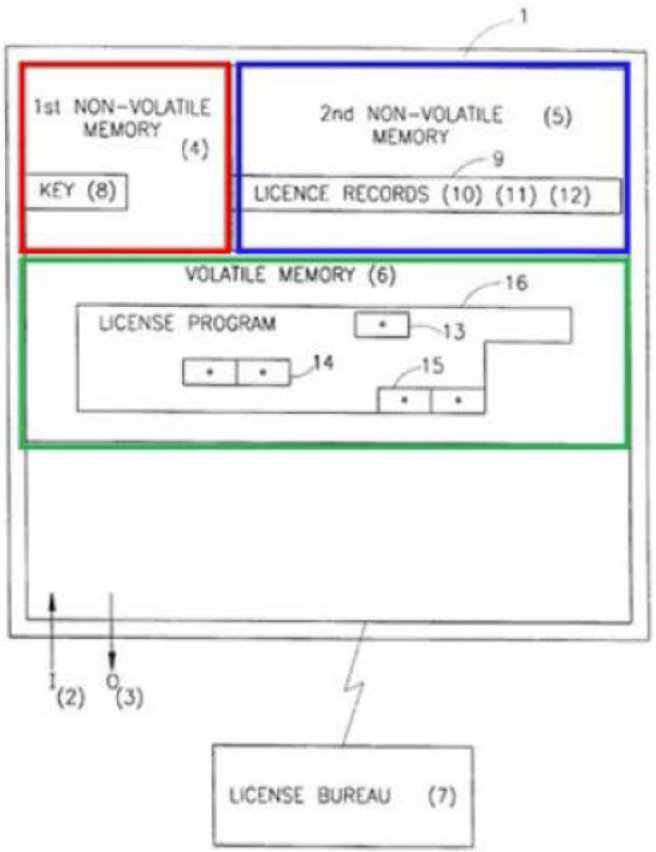
FIG. 8 depicts an implemenation of the base unit **12** during use of a software package. Software package **17** is connected to the base unit **12** and a signal representing said software package is operated on by the one-way hash function generator **33** to produce an output signal which represents the hash value H. The signal H is transmitted to update unit **36** to indicate which software package is being used. **Update unit 36 uses H as an address to non-volatile memory 37, which responds with a signal representing M, the number of uses of software package 17 which are still available.**

If M is greater than 0 then update unit **36** sends a control signal to switch **41** which activates software player **42**, allowing it to use software package **17**. Update unit **36** also decrements M to M−1 and stores this as the new value in address H in non-volatile memory **37**.

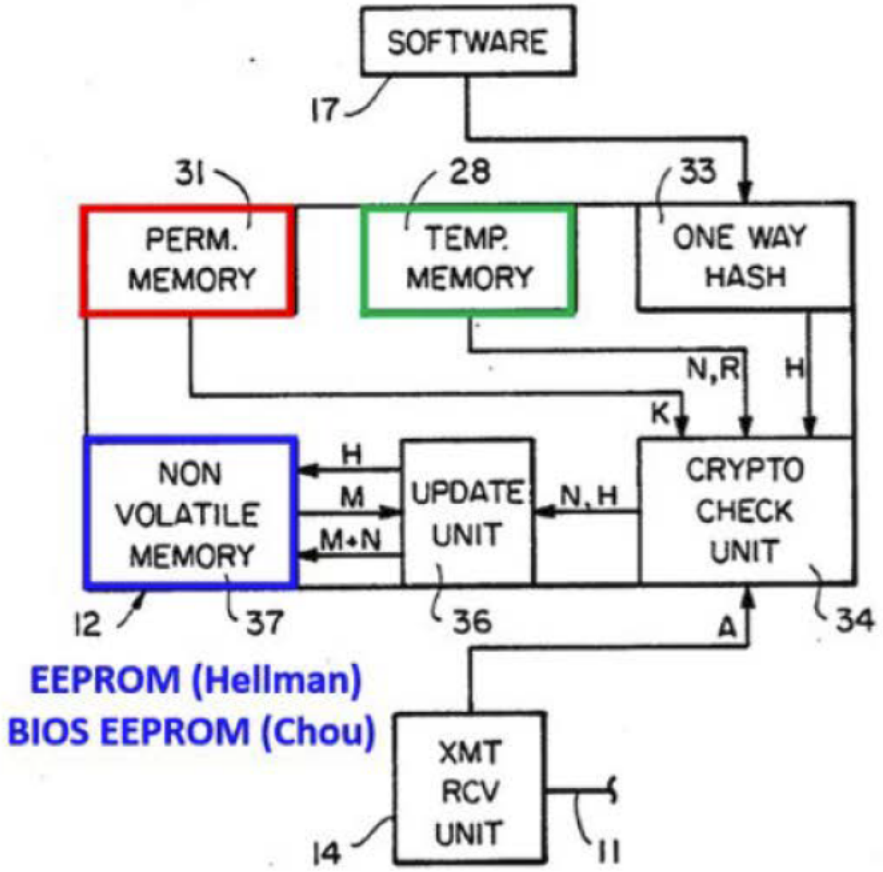
- Hash values are used to interrogate the non-volatile memory and thus in fact exist. *Reply 15; ID 25; EX1033 47; EX1035, 152:8-153:22.*

MOTIVATION TO COMBINE

Motivation to Combine



EX1001, FIG. 1 (annotated).



EX1004, FIG. 6 (annotated).

Pet. at 29

Prior Art – Chou

United States Patent [19]

Chou et al.

[11] **Patent Number: 5,892,906**

[45] **Date of Patent: Apr. 6, 1999**

[54] **APPARATUS AND METHOD FOR PREVENTING THEFT OF COMPUTER DEVICES**

[76] Inventors: **Wayne W. Chou**, 25 Hauley Pl., Ridgefield, Conn. 06877; **Laszlo Elteto**, 86 Snow Crystal La., Stamford, Conn. 06905; **Joseph M. Kulinets**, 40 Meredith La., Stamford, Conn. 06903; **Joseph LaRussa**, 43 Lowell St., Hicksville, N.Y. 11801

[21] Appl. No.: **684,659**

[22] Filed: **Jul. 19, 1996**

[51] **Int. Cl.**[6] **G06F 7/00**

[52] **U.S. Cl.** **395/188.01**; 395/652

[58] **Field of Search** 395/186, 188.01, 395/187.01, 183.12, 652; 380/3, 4, 23, 25

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,634,807	1/1987	Chorley et al.	178/22.08
4,757,533	7/1988	Allen et al.	380/25
4,864,494	9/1989	Kobus, Jr.	395/186
4,866,769	9/1989	Karp	380/4
4,937,861	6/1990	Cummins	380/2
5,007,082	4/1991	Cummins	380/4
5,097,504	3/1992	Camion et al.	380/23
5,146,499	9/1992	Geffrotin	380/23
5,214,695	5/1993	Arnold et al.	380/4
5,222,135	6/1993	Hardy et al.	380/4
5,325,430	6/1994	Smyth et al.	380/4
5,363,446	11/1994	Ruppertuz et al.	380/4
5,369,707	11/1994	Follendore, III	380/25
5,377,269	12/1994	Heptig et al.	380/25
5,402,492	3/1995	Goodman et al.	380/25
5,410,699	4/1995	Bealkowski et al.	395/700
5,421,006	5/1995	Jablon et al.	395/183.14
5,432,939	7/1995	Blackledge, Jr. et al.	395/700
5,448,045	9/1995	Clark	380/4
5,483,596	1/1996	Rosenow et al.	380/25
5,483,649	1/1996	Kuznetsov et al.	395/186
5,497,421	3/1996	Kaufman et al.	380/23
5,535,409	7/1996	Larvoire et al.	395/188.01
5,586,301	12/1996	Fisherman et al.	395/186
5,615,263	3/1997	Takahashi	380/4
5,707,777	1/1998	Sloan et al.	395/188.01

Primary Examiner—Joseph E. Palys

Attorney, Agent, or Firm—Pollock, Vande Sande & Amernick

[57] **ABSTRACT**

Apparatus and method for discouraging computer theft. The apparatus and method requires that a password or other unique information be supplied to the computer before the computer BIOS routines can be completely executed. A BIOS memory storing the BIOS routines includes a security routine which will determine whether or not the required password entered by the user, or a known quantity read from an externally connected memory device is present. The security function stored within the BIOS memory also includes an administration function which permits the computer to be either placed in a locked state, thereby requiring password or the known quantity read from an externally connected memory device to be present each time the computer is booted up. The administration function also permits an unlock state which permits the computer boot up process to complete without entering any password or externally supplied quantity. The external memory location is consulted during each boot up sequence, to determine whether the computer has been placed in the locked or in the unlocked state. If the security depends upon the supply of the known quantity from an externally connected memory device, the computer will be inoperable to anyone not in possession of the external memory device. In the event that the external memory location bearing the locked or unlocked code is removed, the security function assumes the computer to be in the locked state, thus frustrating avoidance of the locked state by tampering with the external memory.

16 Claims, 5 Drawing Sheets

[57] **ABSTRACT**

Apparatus and method for discouraging computer theft. The apparatus and method requires that a password or other unique information be supplied to the computer before the computer BIOS routines can be completely executed. A BIOS memory storing the BIOS routines includes a security routine which will determine whether or not the required password entered by the user, or a known quantity read from an externally connected memory device is present. The security function stored within the BIOS memory also includes an administration function which permits the computer to be either placed in a locked state, thereby requiring password or the known quantity read from an externally connected memory device to be present each time the computer is booted up. The administration function also

Pet, 27; EX1005, Abstract

Petitioner's Demonstrative Exh bit
Not Evidence

Prior Art – Chou

- Chou makes important observations relative to BIOS memory. Reply, 27-28.
  - Using EEPROM for BIOS allows one to write data to BIOS memory.
  - This allows BIOS memory, stored in EEPROM, to be used for security features like password protection, and to store security routines.

Many computer manufacturers have implemented password protection in the computer BIOS (Basic Input/Output System) which is integral to the operation of a personal computer. The password protection in the BIOS halts the system boot up unless the user enters a password which is also stored in the foregoing CMOS RAM. As noted, if the power is removed from the CMOS RAM, the password is cleared and the system will boot up without requiring the user to enter the required password.

Recent changes in the computer BIOS memory storage devices permit writing data to the BIOS memory, offering the opportunity to provide password protection within the same memory which stores the BIOS routines. Thus, any attempt to delete the protection will result in the BIOS

routine being disabled, disabling the boot up process. EEPROM flash devices may be programmed with BIOS routines which permit the user to enter data without requiring the computer to be returned to the manufacture. The present invention makes use of these new BIOS memory devices for effecting security measures which discourage theft.

Pet., 27-28; EX1005, 1:54-2:7

Prior Art – Chou

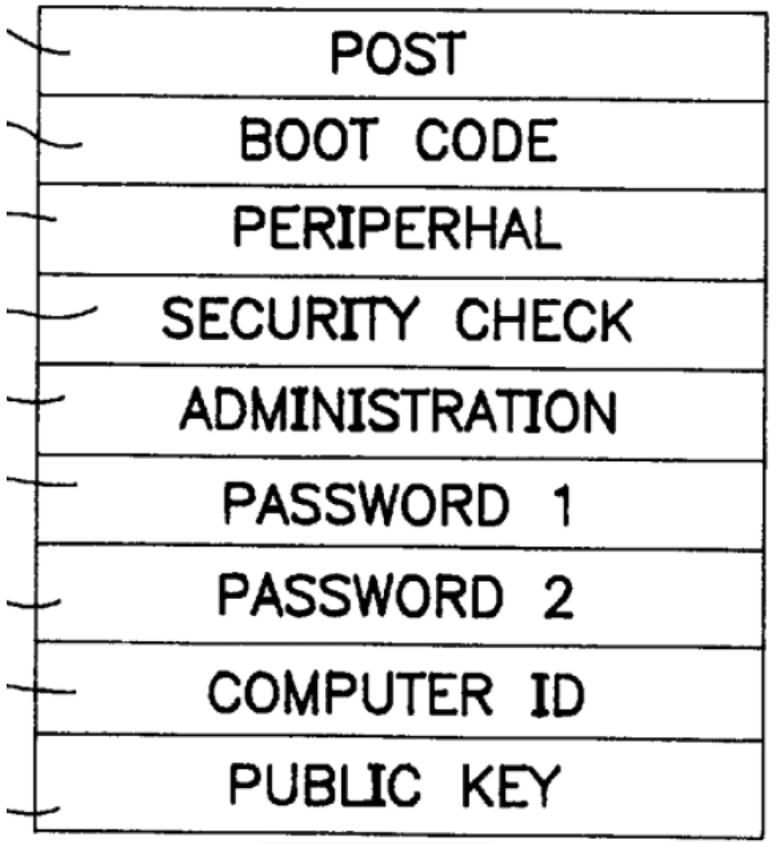
- Chou discloses storing security routines in BIOS EEPROM to discourage piracy.
  Pet. 27-28.

Referring now to FIG. **1**, a general organization of a personal computer **10** is shown which includes a security function stored as a programming routine within the BIOS EEPROM **15**. As will be evident with respect to the description of this embodiment, the BIOS routines which provide for the basic input/output system cannot be completely executed unless the security function is successfully executed.

EX1005, 3:21-29

user entered password for protection. FIG. 7 illustrates the configuration of the BIOS EEPROM **15**(*a*) in a system which relies on a user entered password instead of an externally connected key to enable complete execution of the BIOS routines. First and second passwords are entered

EX1005, 3:21-29

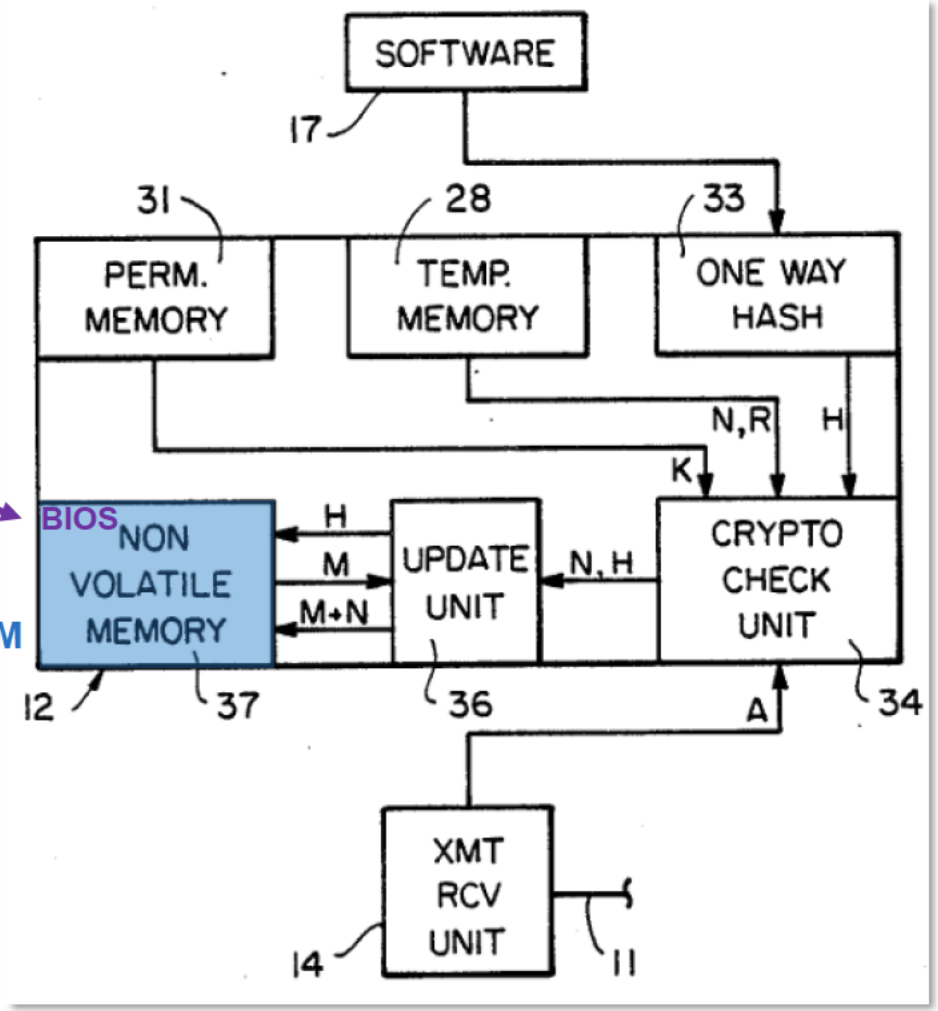


EX1005, FIG. 7

Hellman-Chou Combination

BIOS motivated by Chou

EEPROM



Pet. 29, 33, 35-36; Reply 16-17; EX1004, FIG. 6 (annotated)

Hellman

software being used. Update unit 36 applies to interrogatory signal representing H to a non-volatile memory 37, for example an EEPROM or a CMOS memory with battery backup. The non-volatile memory 37 applies a signal to the update unit 36, said signal representing M, the number of authorized uses of the software package with hash value H which still remain unused prior to this new authorization. The update unit 36 adds M and N and applies a signal representing M+N to the non-volatile memory 37, so that M+N replaces the old number M in the non-volatile memory 37 as the number of uses of the software package which have been paid for.

EX1004, 10:1-13

Chou

Recent changes in the computer BIOS memory storage devices permit writing data to the BIOS memory, offering the opportunity to provide password protection within the same memory which stores the BIOS routines. Thus, any attempt to delete the protection will result in the BIOS routine being disabled, disabling the boot up process. EEPROM flash devices may be programmed with BIOS routines which permit the user to enter data without requiring the computer to be returned to the manufacture. The present invention makes use of these new BIOS memory devices for effecting security measures which discourage theft.

EX1005, 1:63-2:7

DEPENDENT CLAIMS

Prior Art – Schneck

United States Patent [19]
Schneck et al.

[11] **Patent Number: 5,933,498**
[45] **Date of Patent: Aug. 3, 1999**

[54] **SYSTEM FOR CONTROLLING ACCESS AND DISTRIBUTION OF DIGITAL PROPERTY**

[75] Inventors: **Paul B. Schneck**, Potomac; **Marshall D. Abrams**, Silver Spring, both of Md.

[73] Assignee: **MRJ, Inc.**, Fairfax, Va.

[21] Appl. No.: **08/968,887**

[22] Filed: **Nov. 5, 1997**

Related U.S. Application Data

[63] Continuation of application No. 08/584,493, Jan. 11, 1996, abandoned.

[51] **Int. Cl.**[6] **H04L 9/00**

[52] **U.S. Cl.** **380/4**; 380/9; 380/23; 380/25; 380/49; 380/50

[58] **Field of Search** 380/4, 9, 21, 23, 380/24, 25, 49, 50, 51, 55

[56] **References Cited**

U.S. PATENT DOCUMENTS

3,504,132	3/1970	Wallace, Jr. .	
3,764,742	10/1973	Abbott et al. .	
3,798,359	3/1974	Feistel .	
3,878,331	4/1975	Morgan et al. .	
3,906,460	9/1975	Halpern .	
3,911,216	10/1975	Bartek et al. .	
3,944,976	3/1976	France .	
3,958,081	5/1976	Ehrsam et al. .	
3,996,449	12/1976	Attanasio et al. .	
4,004,089	1/1977	Richard et al. .	
4,028,678	6/1977	Moran .	
4,037,215	7/1977	Birney et al. .	

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

0332707	9/1989	European Pat. Off. .
9500355	8/1996	Sweden .
2236604	4/1991	United Kingdom .
2236604	10/1991	United Kingdom .
WO92/20022	11/1992	WIPO .
WO9220022	11/1992	WIPO .
9301550	1/1993	WIPO G06F 11/34
WO93/01550	1/1993	WIPO .
96/27155	9/1996	WIPO .

OTHER PUBLICATIONS

Abrams, M. D. et al, "Cyptography", Information Security–An Integrated Collection of Essays, Abrams, M.D. et al eds., IEEE Computer Society Press 1995, pp. 350–384.
Choudhury, A. K. et al, "Copyright Protection for Electronic Publishing Over Computer Networks", IEEE Network, May/Jun. 1995, pp. 12–20.
Ciciora, W. S., "Inside the Set–Top Box", IEEE Spectrum, Apr. 1995, vol. 32, No. 4, pp. 70–75.

(List continued on next page.)

Primary Examiner—Bernarr E. Gregory
Attorney, Agent, or Firm—Pillsbury Madison & Sutro LLP

[57] **ABSTRACT**

A method and device are provided for controlling access to data. Portions of the data are protected and rules concerning access rights to the data are determined. Access to the protected portions of the data is prevented, other than in a non-useable form; and users are provided access to the data only in accordance with the rules as enforced by a mechanism protected by tamper detection. A method is also provided for distributing data for subsequent controlled use of those data. The method includes protecting portions of the data; preventing access to the protected portions of the data other than in a non-useable form; determining rules concerning access rights to the data; protecting the rules; and providing a package including: the protected portions of the data and the protected rules. A user is provided controlled access to the distributed data only in accordance with the rules as enforced by a mechanism protected by tamper protection. A device is provided for controlling access to data having protected data portions and rules concerning access rights to the data. The device includes means for storing the rules; and means for accessing the protected data portions only in accordance with the rules, whereby user access to the protected data portions is permitted only if the rules indicate that the user is allowed to access the portions of the data.

88 Claims, 26 Drawing Sheets

In general, within the system, the data are encrypted on any non-volatile storage devices so that they remain unavailable in the case of tampering. Unencrypted data are only present within the access mechanism **114** inside the security boundary **167** in components where the data can be destroyed when tampering with the access mechanism **114** is detected.

EX1006, 17:6-12; Pet. 42.

Since all storage of data on internal non-volatile memory devices (for example, disks, flash memory, and the like) is encrypted, this ensures that a physical attack on the system will not result in compromise of plaintext.

EX1006, 25:64-67; Pet. 42.

Dependent Claims 3, 8, 9, 14 (encrypting the LR)

- Schneck protects against "secondary distribution" of software. Pet. 42; EX1006, 6:57-62; 2:40:67.

- Schneck achieves this, in part, by encrypting data "on any non-volatile storage devices so that they remain unavailable in the case of tampering." Pet. 42; EX1006, 17:6-12; *see also* EX1006, 25:64-67; EX1003, 144-150.

- The skilled artisan would have found it obvious to store Hellman's licensing information, in non-volatile memory 37, in *encrypted* form.
  - Specifically, where Hellman's license is for an unlimited number of uses (M is unlimited; EX1004, 10:55-65), the skilled artisan would have stored Hellman's authorization A in encrypted form on Hellman's EEPROM. Pet. 45; EX1003, ¶¶144-50; Reply, 21-22; EX1033, ¶¶62-63.

Dependent Claims 3, 8, 9, 14 (encrypting the LR)

- Contra Sur-Reply at 20, Petitioner did not "cherry-pick" Hellman's "unlimited use" embodiment in its Reply.
  - Petitioner unambiguously identified Hellman's unlimited use embodiment in the Petition as "*especially useful* where M was the default value representing 'unlimited number of uses of a software package' … given that M would not need to be incremented or decremented." Pet. 45.
- Patent Owner's claim of improper hindsight is unsupportable.
  - Schneck discloses storing data in non-volatile flash memory in encrypted form (EX1006, 25:64-67). The skilled artisan would have applied that technique to Hellman's non-volatile memory 37 to "further[] Hellman's goal of preventing a license authorization from being improperly duplicated." Pet. 46; EX1003, ¶¶144-50.
- Petitioner's expert Dr. Wolfe addressed (at EX1033, ¶¶62-64) Patent Owner's complaints about authorization A allegedly not including number of uses M (POR 65). His technical testimony on this point stands unrebutted since Patent Owner did not depose Dr. Wolfe on his Reply declaration.

IPR2021-01406
Appendix Slides

Plain and Ordinary Meaning

- Oxford Dictionary of Computing:
  - "**agent** An autonomous system that receives information from its environment, processes it, and performs actions on that environment. Agents may have different degrees of intelligence or rationality, and may be software, hardware, or both."
  - Contra Sur-Reply at 2, this definition does not "relate[] to 'robots'."
- Contra Sur-Reply at 2, *none* of Ancora's multiple dictionaries "*require*" an agent to be "a software program or routine," let alone an "OS-level" software program or routine.
- Despite its insistence that the "agent" as an "OS-level" software program or routine is a crucial aspect of the purported invention, and its insistence that the Hellman-Chou combination does not disclose an "OS-level" software agent, Patent Owner nonetheless refuses to say what "OS-level" software is or what its defining characteristics might be. It is, according to Patent Owner, a "**non issue**." Sur Reply at 10.

Writing to EEPROM was well-known by March 1998

- Patent Owner's own arguments support Petitioner's position. *Reply 13-14; POR 35; EX2018, 128; EX1033, ¶32; EX1035, 148:6-22, 152:8-153:22, 156:7-157:1, 163:18-164:2, 174:20-176:9, 178:21-179:14; EX1033, ¶¶33-41; EX1029, 5.*

Beeble White Paper

2. EEPROM Writing

Beeble will attempt to write license data by utilizing the DMI Function 52h[2], which is designed to write DMI structures to EEPROM. If BIOS manufactures do not support DMI function 52h, then Beeble will write to a EEPROM generically. Beeble's File System incorporates a driver that contains a library of different EEPROM chips and proper instructions set to write to different chip manufacturers. If the Beeble driver does not recognize the EEPROM, the Beeble driver will attempt to download a new driver from the Beeble License server.

EX2011, ANC000184

Dr. Martin

Q. Is it fair to say that DMI function 52h was known as of March 1998?

MR. GOSSE: Object to the form.

THE WITNESS: Well, assuming that this footnote on ANC 184 has an accurate date associated with this publication of the SM BIOS reference specifications, then I would also expect that the documentation being referred to here was available at that time in 1998.

BY MR. CRUDO:

Q. Is it fair to say as a general matter, writing data structures to EEPROM was known as of March 1998?

A. Generally speaking, yes, it's fair to say that it was known to read and write from EEPROM generally.

EX1035, 178:21-179:14