

EXHIBIT A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re reexam of: U.S. Patent 5,490,216 to RICHARDSON, III Reexam Control No.: 90/010,831 Filed: January 22, 2010 For: System for Software Registration	Confirmation No.: 2214 Art Unit: 3992 Examiner: HENEGHAN, Matthew E. Atty. Docket: 2914.001REX0
---	--

Declaration of William R. Rosenblatt Under 37 C.F.R. § 1.132

Mail Stop *Ex Parte* Reexam
Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

I, William R. Rosenblatt, declare as follows:

Retention:

1. I have been retained as a technical expert witness on behalf of Uniloc USA, Inc. and Uniloc Singapore Private Limited (collectively "Uniloc" herein) for the above-captioned reexamination. I understand that this reexamination involves U.S. Patent No. 5,490,216 ("the '216 patent") which resulted from Application No. 08/124,718, filed on September 21, 1993 on behalf of Frederic B. Richardson, III. I further understand that the '216 patent is currently assigned to Uniloc Singapore Private Limited.

Introduction:

2. Over the past seven years I have been retained as a technical expert witness in nine cases and have testified in that capacity in Federal Court.

3. I have reviewed and am familiar with the Office action dated September 28, 2010 issued by the U.S. Patent and Trademark Office (“USPTO”) for the ‘216 patent (“the ‘216 Office action”). I understand that claims 1-20 of the ‘216 patent are subject to reexamination. Claims 1, 12, 17, 19 and 20 are independent claims. Claims 2-11 depend, directly or indirectly, from claim 1; claims 13-16 depend, directly or indirectly, from claim 12; and claim 18 depends from claim 17.

4. I have reviewed and am familiar with U.S. Patent No. 4,658,093 to Martin E. Hellman (hereinafter “Hellman”) and with U.S. Patent No. 5,291,598 to Gregory Grundy (hereinafter “Grundy”), in combination, as used in the rejection under 35 U.S.C. § 103(a).

5. I am familiar with the level of ordinary skill in the art relevant to the ‘216 patent as of the September 21, 1993 filing date of that patent. In my opinion a person of ordinary skill in the art at the time of the invention (hereinafter “POSA”) would have held a Bachelor of Science degree in Computer Science or Electrical Engineering with an emphasis on computer systems, in addition to one to two years of programming experience.

6. I have been asked to consider the Hellman and Grundy references in view of the claims under reexamination in the ‘216 patent. Specifically, I have been asked to determine facts that would be relevant to the legal conclusion of whether a POSA would have found of the claims under reexamination to be obvious over Hellman in view of Grundy.

7. I am being compensated at my standard hourly rate of \$450.00 for my work on this declaration and my participation in the oral interview with the

reexamination panel on November 17, 2010. My compensation is not dependent on the outcome of this reexamination and in no way affects the substance of my statements during the oral interview or in this declaration.

Qualifications

8. I have more than 27 years of professional experience and familiarity with the art of computer and software architecture and design. I have had over 15 years of particular experience in the field of digital rights management (DRM) and antipiracy technology. I am the lead author of the book *Digital Rights Management: Business and Technology* (Wiley, 2001) and several other book chapters, white papers, and articles on related subject matter. I have also chaired conferences since 2004 and spoken at conferences since 1999 on related subject matter worldwide.

9. I earned a Bachelor of Science in Electrical Engineering and Computer Science *cum laude* from Princeton University in 1983 and a Master of Science degree in Computer and Information Science from the University of Massachusetts at Amherst in 1990. I did research and completed coursework towards a PhD in Computer and Information Science from the same university; my field of research was the intersection of software engineering, databases, and object-oriented systems. I have had business education from New York University, Harvard University, and the University of Southern California.

10. My work experience includes periods as a software engineer at Motorola, Intermetrics (now Titan Corp.), and The Rustin Group, as well as Information Technology management and Research and Development roles at Moody's Investors Service, Times Mirror Co., the McGraw-Hill Companies, and Fathom Knowledge

Network (an Internet e-learning startup at Columbia University). I have managed the consulting firm GiantSteps Media Technology Strategies since June 2000. GiantSteps' clients have included several vendors of DRM, antipiracy, and related technologies. I have also performed patent infringement and valuation analysis for several patent holders and potential patent acquirers.

My Understanding of Obviousness:

11. It is my understanding that a claimed invention is unpatentable if the differences between the invention 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 the subject matter pertains.

12. It is my understanding that "obviousness" is a question of law based on underlying factual issues including the content of the prior art and the level of skill in the art. I understand that for a single reference or a combination of references to render the claimed invention obvious, a person of ordinary skill in the art must have been able to arrive at the claims by altering or combining the applied references in an obvious manner.

13. I also understand that when considering the obviousness of a patent claim, one should consider whether a teaching, suggestion or motivation to combine the references exists so as to avoid impermissibly applying hindsight when considering the prior art. I understand this test should not be rigidly applied, but that the test can be important to avoiding such hindsight.

14. In addition, it is my understanding that one must consider whether or not there is objective evidence of non-obviousness, which is also referred to as the “secondary considerations of non-obviousness.”

Claim Construction:

15. I understand that claim construction is a matter of law. But to understand the invention, I necessarily have to arrive at my own conclusions as to certain terms that appear in the ‘216 patent claims. I understand that claim terms are to be read in view of the patent specification. If a term is not explicitly defined there, the default definition of the term is that which a POSA would use or infer from context. If the term in question is not a term of art, then I would naturally use the ordinary English definition of the term.

“Licensee unique ID,” “Security Key,” “Registration Key,” and “Enabling Key”:

16. In my view, these terms refer to the same thing. The ‘216 patent at 13:4-10, describing an embodiment of the invention, discloses:

Remote summer 89 combines these signals by addition (thereby acting as a remote licensee unique ID generating means) so as to provide a summed output, here termed X, which represents a *licensee unique ID* or *enabling key* which should match identically with the local licensee unique ID or *registration key* or *registration number* Y if inputs S, C and P to summers 85 and 89 are identical. (‘216 patent at 13:4-10, emphasis added)

17. From this language the “licensee unique ID,” “enabling key,” and “registration key” are the same, because they are either synonymous or of equal value. In addition, this language defines “registration number” as the same as the aforementioned terms.

18. As for “security key,” it is defined in the patent specification with the language “registration number or security key” several times, e.g., ‘216 patent at 7:29-30. I therefore understand all of the terms to have the same meaning.

19. The term “licensee unique ID” is the most common of these terms in the specification and I will focus on it for that reason.

20. The patent specification indicates that the unique ID is associated with information about the user:

The registration dialogue box C ... prompts the user for details unique to that user (including, for example, name, company, address, state, contact number) together with financial details for payment for the purpose of becoming a registered user of the software protected by the registration routine... This information, unique to the user, is passed through a registration algorithm ... which generates a registration number or security from the information unique to the user together with the serial number previously generated. (*216 patent at 7:8-19, parenthetical statements and figure references omitted)

21. I understand “user” and “licensee” to be synonymous: the user is the entity intending to take the license to the software. The specification does not disclose the registration algorithm in sufficient detail to determine whether or not the unique ID can be said to be “derived from” the user information; therefore it suffices to say that the unique ID is “associated with” the user information.

22. “Derived from” suggests a calculation or algorithm that takes the user information as input. The invention leaves open possibilities for other methods of generating unique IDs from user information -- table lookups, for example. Therefore a looser relationship between the user information and the ID is warranted, and the unique ID is “associated with” the licensee.

23. Each of the above terms therefore requires a unique identifier that is somehow associated with a licensee.

“Registration system”:

24. This term is explained in two places in the ‘216 patent. First, the specification states that “A registration system 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.” (‘216 patent, Abstract).

25. Second, one of the disclosed embodiments states that “...the registration system is implemented such that the digital data and, more particularly, the use code portion of that digital data can be executed on the platform in a use mode only if the registration procedure to which reference has been made in respect of previous embodiments has been performed.” (‘216 patent at 11:8-13, figure citations omitted)

26. The “registration procedure” in the above is defined in the disclosure of the first embodiment (‘216 patent at 8:12-13) as the procedure described in Figure 2b.

27. The term “registration system” is used in disclosures of subsequent embodiments with reference to the definition in the first embodiment; see for example ‘216 patent at 10:42-44 (fifth embodiment: “This registration procedure is as previously described with reference to the first embodiment.”) and 11:8-13 as above (sixth embodiment).

28. For these reasons, the definition of “registration system” provided in the Abstract is consistent with its use in the remainder of the specification—namely, “A

registration system 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.”

“Wherein said registration system is replicated at the registration authority”:

29. This phrase is an element of claim 12 of the ‘216 patent. The “registration authority” is the remote system on which the remote licensee unique ID generating means is located.

30. Otherwise, the word “replicated” appears only once in the specification, in language that is substantially the same as that recited in claim 12. (‘216 patent at 4:13-17, the only difference being “means” in the latter instead of “system” in claim 12).

31. “Replicate” or “replicated” is a term of art in the software field, but in a subfield that is irrelevant to the subject matter of the ‘216 patent. It refers to a technique for duplicating data stored in a database onto another database and synchronizing updates to the data so that they are kept current in both databases. This has nothing to do with the procedures referred to in the ‘216 patent.

32. Turning to the English dictionary definition of “replicated”: the Merriam-Webster Third New International Dictionary defines “replicate” (verb) as “duplicate, repeat¹.”

33. Therefore I understand “replication” of a registration system in claim 12 to refer to duplication of the registration system at the registration authority.

¹ Merriam-Webster *Third New International Dictionary*, p. 1925. Springfield, Massachusetts: Merriam-Webster, 1986.

Analysis of Claim 1 Over Hellman in View of Grundy:

34. Claim 1 of the '216 patent stands rejected on grounds of obviousness over Hellman in view of Grundy. Claim 1 recites local and remote "licensee unique ID" generation by the same algorithm on both local and remote systems.

35. The Office action asserts that "Hellman discloses several algorithms for local licensee unique ID and remote licensee unique ID generation" and then relies on Grundy to supply an algorithm for unique ID generation, noting that "it would be obvious to one of ordinary skill in the art at the time the invention was made to modify Hellman to use Grundy's checksum for ID generation, because it is easier to implement" than the summer disclosed in the '216 patent. ('216 Office action, p. 7).

Neither Hellman Nor Grundy Teach a Local or Remote Licensee Unique ID

36. To form an opinion on the above assertion in the Office action, first I consider whether either Hellman or Grundy teaches the generation of "licensee unique ID" (hereinafter "LUID") in any form. My opinion is that they do not. In support of the rejection, the '216 Office action cites to Hellman at 10:14-18, which refers to the "base unit" in Hellman's Figure 7, and equates that to the "local platform" in the '216 patent. The Examiner also cites to Hellman at 6:62-7:2, which refers to the "authorization and billing unit" in Hellman's Figure 2, and equates that to the "remote platform" in the '216 patent.

37. My showing that Hellman in combination with Grundy does not teach LUID generation follows three steps: First, I show that Hellman does not teach "*licensee* ID generation." Second, I show that Hellman does not teach "*unique* ID generation" (at

either local or remote locations). Third, I show that Grundy does not cure the deficiencies of Hellman such that the resulting combination teaches local or remote LUIDs.

38. First, regarding Hellman and “*licensee* ID generation”: the process in Hellman cited by the examiner above shows that the “check value C” (Hellman at 10:17), which the Examiner equates to LUID, is generated from four inputs, designated as K, N, R, and H. Of these: K is a *key* to a cryptographic function, which is an indicium of the *computer* on which the software is intended to be run. N is the *number of usages* of the software requested by the user (see Hellman at 5:65-66); R is a *random number* (Hellman at 5:66); H is an indicium of the *software package* being authorized for use (Hellman at 6:31-35), which is computed by means of a hash function. A hash function produces a value that serves as a mathematical “shorthand” for some data that has properties described appropriately in Hellman at 6:38-61, including that it is efficient to calculate and store.

39. None of these four inputs is an indicium of the *licensee* of the software, i.e., the user intending to run the software on the computer. Therefore Hellman does not teach “licensee ID generation.”

40. I have reviewed Hellman’s sworn testimony at trial. It reinforces my conclusion. The following is an excerpt from examination of Hellman at trial:

[Attorney] Question: If you wanted to indicate that information associated with the user, unique information was input into the cryptographic function, you certainly had the ability to disclose that in the figures, if you so chose.

[Hellman] Answer: Correct.

[Attorney] Question: And you didn’t?

[Hellman] Answer: Correct.

[Attorney] Question: And you also had the ability to describe in the patent, if you so chose?

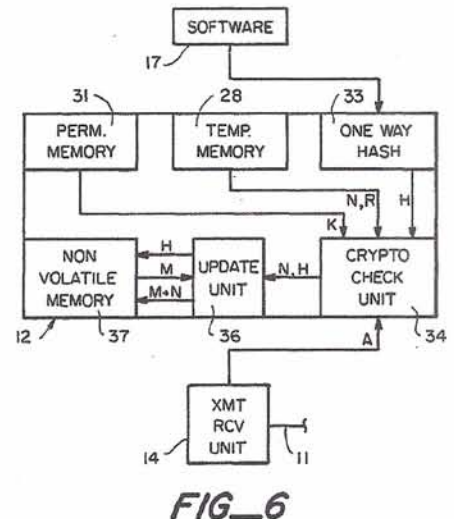
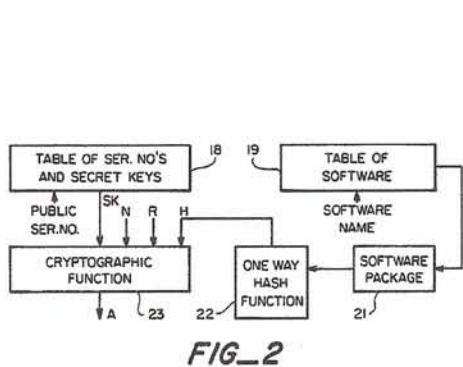
[Hellman] Answer: In the specification? Yes.

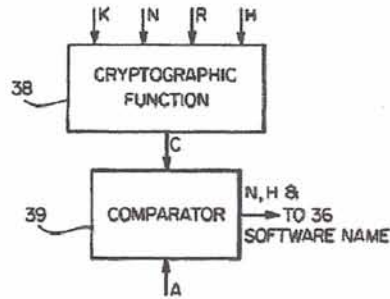
[Attorney] Question: And you didn't?

[Hellman] Answer: Correct.

(March 31, 2009 Trial Transcript: p. 61, ll 17 - p. 62, ll 4, Uniloc USA, Inc. et al. v. Microsoft Corp., C.A. No. 03-440 (D.R.I.))

41. Second, I show that Hellman does not teach “*unique* ID generation” at either local or remote locations. Three figures in Hellman depict the generation of the ID that is to be checked for validity: Figures 2, 6, and 7 (shown below). Figure 2 shows a Cryptographic Function with inputs SK, N, R, and H that is used on the base unit (equivalent to the local system in the ‘216 patent); its output is marked A and labeled 23. Figure 6 shows a more complete picture of the base unit. Figure 7 shows a Cryptographic Function with inputs K, N, R, and H and output C, which is part of the authorization and billing unit (equivalent to remote system in the ‘216 patent). These have outputs designated A and C respectively. Because these are intended to be equal if the software is authorized to run, I use the designation C hereinafter to refer to either of the outputs. The Examiner equates C to the LUID in the ‘216 patent.





FIG_7

42. Hellman does not teach that C is unique. Here are descriptions of C (or A) as disclosed in the patent specification: "Generator 23 has the property that new authorizations cannot be inferred from old authorizations." (Hellman at 7:3-4 describing Figure 2.) This property is not the same thing as "authorizations must be unique." Authorizations need not be unique to satisfy this property, nor would a POSA infer uniqueness from this property.

43. For Figure 6, Hellman refers the reader to the ensuing description of Figure 7 for a disclosure of the output of the Cryptographic Function: "In a manner to be described shortly, the cryptographic check unit 34 determines if the authorization A is valid for the software package 17 being presented to the base unit 12 for the current authorization request." (Hellman at 9:54-57)

44. For Figure 7, Hellman discloses: "FIG. 7 depicts an implementation of the cryptographic check unit 34. Signals representing K, N, R and H are applied as inputs to a cryptographic function generator 38 which generates a check value C as an output signal." (Hellman at 10:14-18)

45. These are the only disclosures in Hellman that describe any properties of C. None of them teach that C must be unique, nor would they suggest such teaching to a POSA.

46. Therefore Hellman does not teach “unique ID generation.”

47. Because Hellman teaches neither “licensee ID generation” or “unique ID generation,” Hellman does not teach local LUID generation, nor does Hellman teach remote LUID, and thus cannot by itself satisfy the “licensee unique ID generating means” element of claim 1 of the ‘216 patent.

Neither Hellman Nor Grundy Disclose or Teach Any Means for Generating Unique IDs:

48. The Office action claims that the combination of Hellman with Grundy teaches “local or remote licensee unique ID generation.” The Office action relies on Hellman to teach ID generation, and introduces Grundy in order to add Grundy’s checksum algorithm for generating unique IDs. As I will show, Grundy’s checksum algorithm does not do this; therefore Grundy does not cure the acknowledged deficiency in Hellman. In short, neither Hellman nor Grundy disclose or teach any “local or remote licensee unique ID generation.”

49. In support, I will now show that Grundy does not supply an adequate algorithm for generating unique IDs. The Examiner cites to Grundy, including “the checksum originally calculated ... from the owner data as entered by the user during the registration process” (See, Grundy 15:11-13, figure citation omitted) and “a checksum of the user data” (See, Grundy 18:25-26). (‘216 Office action p. 7.)

50. First, it is worth noting that the Order granting reexamination of the '216 patent characterizes checksums as non-unique: "*Checksums are not unique fields*, even if there [sic] are at least in part derived from unique data." (Order Granting/Denying Request for Ex Parte Reexamination, p. 9, emphasis added)

51. I concur with that characterization of checksums. A checksum is not usable as a generator of unique IDs. Grundy does not define "checksum" in the specification in the context of "a checksum of the user data," but rather suggests definitions elsewhere in his specification that are consistent with both dictionary definitions and a definition that a POSA would use, as I will now show.

52. A POSA would understand "checksum" to mean a small number of check digits that are typically appended to data in order to ensure the data's integrity when it is copied, entered by a user, or transmitted. To calculate a checksum of some data, the data can be added up (e.g., broken up into C-byte chunks, where C is a small number such as 1, 2, 4, or 8, and summed); the sum is chopped to a fixed length (e.g., C bytes) and appended to the data before storage or transmission. Checksum algorithms used in practice are variations on this scheme. When the data is received or retrieved later, the checksum is re-calculated to ensure that the result is the same as the original checksum; if the result differs then the data must have been corrupted.

53. The Grundy patent was filed in April 1992 and issued in March 1994. Definitions of "checksum" from dictionaries of computer terms from that era concur with the POSA's definition that I have provided above.

54. For example: *The Computer Glossary: The Complete Illustrated Desk Reference*, 6th Edition, 1993 defines "checksum" as a "Value used to ensure data is

transmitted without error. It is created by adding the binary value of each alphanumeric character in a block of data and sending it with the data. At the receiving end, a new checksum is computed and matched against the transmitted checksum. A non-match indicates an error.”

55. As another example, the *IBM Dictionary of Computing*, 10th Edition, 1994 defines “checksum” as: “(1) The sum of a group of data associated with the group and used for checking purposes. (2) In error detection, a function of all bits in a block. If the written and calculated sums do not agree, an error is indicated. (3) On a diskette, data written in a sector for error detection purposes; a calculated checksum that does not match the checksum of data written in the sector indicates a bad sector. The data are either numeric or other character strings regarded as numeric for the purposes of calculating the checksum.”

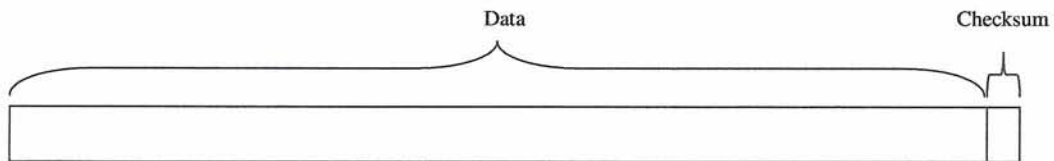
56. In all of these cases, a checksum is much smaller in length than its input data. For example, a 16-bit (2-byte) or 64-bit (8-byte) checksum may be calculated on thousands, millions, or billions of bytes of data. This fulfills the checksum’s intended purpose well, given that most errors in data storage or transmission are small and localized, making it highly likely that the resulting checksum will differ from the one originally calculated, and extremely unlikely that corrupted data will produce the same checksum as the original one. For example, if one or two bits are altered, the checksum will differ.

57. A few examples highlight this point. Most credit card numbers contain check digits that serve to make sure that a credit card holder enters the number properly (e.g., on a website). The check digits are often calculated by means of an

algorithm called the Luhn algorithm, which is a type of checksum algorithm. Because it is a single digit (with values 0-9), it is clearly impossible for the check digit to be unique, given that most credit card numbers are 15 digits long (not counting the check digit) and therefore that there are up to 10^{15} or 1 quadrillion possible credit card numbers.

58. As another example, consider social security numbers, which are 9 digits long. Suppose we wanted to calculate the checksum of a social security number. We can use the most basic checksum algorithm, in which all the digits in the input data are added and the result is the checksum. All checksum algorithms in practical use are derived from this basic one. The social security numbers 123-45-6789 and 987-65-4321 are clearly different, but they both have the same checksum of 45; moreover, it is impossible to reconstruct (preserve the uniqueness of) the original social security number given the checksum.

59. As a final example, electronic devices often have memory chips in them that contain data stored at the factory. These are known as PROMs (Programmable Read-Only Memory). A typical PROM will contain a checksum of 8 bytes in length, even though the capacity of the PROM may be in the thousands or millions of bytes, as illustrated in the diagram below. The data in the PROM can be added up byte by byte, and the lowest 8 bytes of the result can be used as the checksum. The checksum is used to ensure that the data in the PROM was copied correctly during manufacturing. It works well for that purpose, because errors are typically limited to a few bytes, which would result in a different checksum being calculated. However, many different sets of PROM data would lead to the same checksum, meaning that uniqueness of the data is not preserved.



60. Grundy himself suggests the use of checksums as error-checking means in disclosures of checksums in his invention other than the “checksum of the user data” used to generate the code that the Office action alleges satisfies the LUID element of claims in the ‘216 patent. Here are some examples:

- “If there is a match, the *anti-virus checksum* as originally calculated during the Franking process retrieved from the central database is compared with the checksum of the host software product as calculated in the process of generating the registration code. If these two checksums do not match, it indicates that *the host software product has in some way been deliberately or accidentally corrupted.*” (Grundy at 15:28-36, figure citations omitted, emphasis added.)
- “If the user data validity check passes, a checksum of the user data is created. This checksum will be used during the Authorization process as a cross-reference to *validate the user data* as communicated to the Manufacture Control Agency.” (Grundy at 18:26-29, figure citations omitted, emphasis added.)
- “A checksum is then calculated from the bits in the array and added to the packed bit array. This checksum will be used by the Manufacture Control Agency to *avoid operator data-entry errors.*” (Grundy at 19:4-8, figure citations omitted, emphasis added.)

61. From the foregoing, Grundy understood well the use of checksums according to the POSA's and dictionary definitions described above; therefore one may assume that his understanding of checksums applies to all of the disclosures of that term in his patent, including the "checksum of the user data" as referred to above.

62. Grundy shows the input data to this checksum routine in Fig. 2, 212, "ENTER NEW USER DETAILS." This is "new user data, such as the user's name, address and telephone number" (Grundy at 12:37-38). Such data might take up roughly a hundred bytes of data. As indicated above, a checksum of this data would not preserve its uniqueness; many different sets of user data could produce the same checksum.

63. For the above reasons, a checksum cannot possibly preserve whatever uniqueness the input data may possess. In particular, a POSA would not ascribe any reasonable definition of "unique" to the output of a checksum routine.

64. Therefore the checksum as disclosed in Grundy cannot function as a generator of unique identifiers, which is a required characteristic of the claimed LUID. Grundy thus does not overcome the acknowledged deficiency in Hellman.

65. I therefore disagree with the factual conclusions stated in the Office action with respect to the teachings of Hellman and Grundy—neither of them teach or disclose licensee unique ID generation.

Extension of Arguments to Independent Claim 12:

66. As with claim 1, the '216 Office action also rejects claim 12 of the '216 patent on grounds of obviousness with respect to Hellman in view of Grundy. Claim 12 of the '216 patent recites a "registration system" to generate 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.

67. As discussed above, I understand the term “Security Key” in claim 12 to be synonymous with the “unique identifier associated with a licensee.” As previously discussed with regard to claim 1, Hellman fails to disclose an identifier associated with a licensee. Hellman also fails to disclose an identifier associated with a licensee that is also unique. For these reasons, Hellman fails to disclose the “security key” recited in claim 12.

68. Using the reasoning discussed above (see paragraphs 37-47), Hellman does not teach or disclose the claim 12 element of “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.” By the same reasoning, Hellman does not teach or disclose “wherein said registration system is replicated at a registration authority.”

69. The Office action again relies on Grundy to allegedly satisfy the claim element “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” through the generation of a checksum. The Office action states that Grundy supplies “user data,” but Grundy's use of a checksum cannot “uniquely identif[y] an intended registered user of said software” for reasons given above.

70. Grundy does teach “allocating a unique user code to the user of the information product” (Grundy at 24:60-61), but this is not the same thing as uniquely

identifying a user. If the data input by the user is not unique then the “unique user code” does not confer uniqueness on it. On the contrary, it could result in different unique user codes being assigned to the same user (who has more than one computer on which he intends to use the software).

71. For the above stated reasons, I disagree with the factual conclusions stated in the Office action surrounding the teachings of Hellman and Grundy as they are applied to claim 12.

Extension of Arguments to Independent Claim 17:

72. As with claim 1, the ‘216 Office action also rejects claim 17 of the ‘216 patent on grounds of obviousness with respect to Hellman in view of Grundy. Claim 17 of the ‘216 patent recites a method comprising providing registration key generating means adapted to generate a registration key which is a function of information unique to an intending user of the software.

73. As discussed above, I understand the terms “Registration Key” and “Enabling Key” to be synonymous with “a unique identifier associated with a licensee.” As previously discussed in regards to claim 1, Hellman fails to disclose an identifier associated with a licensee. Hellman also fails to disclose an identifier that is unique and therefore fails to disclose either the registration key or the enabling key recited in independent claim 17.

74. Using the reasoning discussed above (see paragraphs 37-47), Hellman does not teach or disclose a “method further comprising providing registration key generating means adapted to generate a registration key which is a function of

information unique to an intending user of the software.” By the same reasoning, Hellman does not teach or disclose an “enabling key generated by a third party means of operation of a duplicate copy of said registration key generating means.”

75. In supporting his rejection of claim 17, the Office action asserts that Hellman allegedly discloses means for switching “between a fully enabled mode and a partly enabled or demonstration mode” as per claim 17. The Office action to cites Hellman’s mention of such an invention in the prior art (Hellman at 2:14-48).

76. I disagree that Hellman’s prior art reference teaches mode switching. Hellman does not identify his prior art reference, so I rely solely on Hellman’s own description of it as cited above. As Hellman describes it:

“The manufacturer provides each customer for its program with two versions: the regular version is in a sealed envelope, and a ‘degraded’ version, which only allow[s] up to 30 records per file. Within a 30 day trial period, the manufacturer allows the customer to return the regular version in its sealed envelope for a refund. This way, the customer can experiment with the degraded version to determine if he wants the full package.”
(Hellman at 2:15-23)

77. This describes two completely separate software packages, one of which happens to be a “‘degraded’ version” of the “full package.” It does not describe means of switching use modes of a given piece of software from one to the other.

78. The Office action further cites Grundy’s use of “a checksum, which is used as a registration key, that is derived at least in part from the user data.” I assume that the Office action cites this to satisfy the claim 17 element “registration key generating means adapted to generate a registration key which is a function of information unique to an intending user of the software.” I disagree that Grundy’s

checksum meets this claim element for reasons described above under claim 12 (see paragraphs 69-70) as well as claim 1 (see paragraphs 49-64).

79. For the above reasons, I disagree with the factual conclusions stated in the Office action in support of its rejection of claim 17 (and claim 18 which depends from claim 17) over Hellman in view of Grundy.

Extension of Arguments to Independent Claims 19 and 20:

80. As with independent claim 1, the '216 Office action also rejects claims 19 and 20 of the '216 patent on grounds of obviousness with respect to Hellman in view of Grundy. Like claim 1, independent claims 19 and 20 of the '216 patent both recite a "local licensee unique ID" and a "remote licensee unique ID," and "wherein said remote licensee unique ID generation means comprises software executed on a platform which includes the algorithm utilized by said local licensee unique ID generation means to produce said licensee unique ID."

81. As previously discussed in regards to claim 1, (see paragraphs 34-65), neither Hellman nor Grundy, alone or in combination, teach or disclose a "local licensee unique ID," a "remote licensee unique ID," and "wherein said remote licensee unique ID generation comprises software executed on a platform which includes the algorithm utilized by said local licensee unique ID generation to produce said licensee unique ID."

82. Independent claim 19 is similar to claim 1 except disclosed from the perspective of the remote system instead of the user's computer. In rejecting claim 19, the Office action relies on the same arguments expressed in rejecting claim 1, regarding obviousness over Hellman in view of Grundy's checksum for generating unique IDs,

stating that “Hellman discloses a system including local licensee unique ID generation” and that “Grundy discloses an analogous algorithm for unique ID generation, wherein the unique ID, a registration code, is produced by performing a checksum of the user data component fields.” (Office action, p. 12). Accordingly, I disagree with the factual conclusions expressed in the Office action in support of the rejection of independent claim 19 for the same reasons as for claim 1 above; see paragraphs 34-65 above.

83. Independent claim 20 is a method claim that is otherwise similar to claim 1 (a system claim). In rejecting claim 20, the Office action relies on the same arguments expressed to reject claim 1, regarding obviousness over Hellman in view of Grundy’s checksum for generating unique IDs. Accordingly, I disagree with the factual conclusions expressed in the Office action in support of the rejection of claim 20 for the same reasons as for claim 1 above; see paragraphs 34-65 above.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the '216 patent.

Executed this 23rd day of November 2010 in New York, NY.

Respectfully submitted,

A handwritten signature in cursive script, appearing to read "William R. Rosenblatt". The signature is written in black ink and is positioned above the printed name.

William R. Rosenblatt