Internal distribution code:

(A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

Datasheet for the decision of 24 April 2018

Case Number: T 2193/15 - 3.5.06

Application Number: 08726684.7

Publication Number: 2137609

IPC: G06F9/06, G06F9/22, G06F21/00, G06F21/24

Language of the proceedings: EN

Title of invention:
TRUSTED COMPONENT UPDATE SYSTEM AND METHOD

Applicant:
Hewlett-Packard Development Company, L.P.

Headword:
Trusted component update/HEWLETT-PACKARD

Relevant legal provisions:
EPC Art. 56

Keyword:
Inventive step - (no)

Decisions cited:

This datasheet is not part of the Decision.
It can be changed at any time and without notice.

Catchword:

This datasheet is not part of the Decision.
It can be changed at any time and without notice.

Europäisches Patentamt
European Patent Office
Office européen des brevets

Beschwerdekammern

Boards of Appeal

Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 2193/15 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 24 April 2018

Appellant: (Applicant)	Hewlett-Packard Development Company, L.P. 11445 Compaq Center Drive West Houston, TX 77070 (US)
Representative:	Zimmermann, Tankred Klaus Schoppe, Zimmermann, Stöckeler Zinkler, Schenk & Partner mbB Patentanwälte Radlkoferstrasse 2 81373 München (DE)
Decision under appeal:	**Decision of the Examining Division of the European Patent Office posted on 9 June 2015 refusing European patent application No. 08726684.7 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman	W. Sekretaruk
Members:	M. Müller
	A. Teale

Summary of Facts and Submissions

I. The appeal is against the decision of the examining division dated 9 June 2015 to refuse European patent application No. 08726684 for lack of inventive step in view, in particular, of the document

D1: US 2006/0101310 A1.

II. Notice of appeal was filed on 27 July 2015, the appeal fee being paid on the same day. A statement of grounds of appeal was received on 29 September 2015. The appellant requested that the decision be set aside and a patent be granted based on claims 1-6 according to a main request or an auxiliary request, both as filed with the grounds of appeal, in combination with the other application documents on file.

III. In an annex to a summons to oral proceedings, the board informed the appellant of its preliminary opinion that the claimed invention lacked inventive step, Article 56 EPC.

IV. In response to the summons, by letter dated 8 March 2018, the appellant filed arguments and, at the same time, indicated that it would not attend the scheduled oral proceedings. Furthermore, by letter dated 21 March 2018, it withdrew its request for oral proceedings, which were then cancelled.

V. Claim 1 of the main request reads as follows:

"A trusted component update system (10), comprising:
a computing device (121) comprising a processing unit (120), a first memory (110) and a trusted

component (90) comprising a trusted memory (100) separate from the first memory (110);

verify logic (111) stored in the first memory (110) and configured to validate integrity of an update (112) to the trusted component (90) of the computing device (121); and

logic (104) disposed in a boot block (102) stored in the trusted memory (100) of the trusted component (90) and configured to validate integrity of the verify logic (111),

wherein the update is applied to modify the contents of the boot block (102) if the validation of the integrity of the verify logic (111) is successful and if the update (112) is successfully verified by the verify logic (111),

wherein the logic (104) disposed in the boot block (102) is configured to hash at least a portion of the verify logic (111) and compare a result of the hashing with a predetermined hash value (105) stored in the boot block (102)."

Claim 1 of the auxiliary request is identical to claim 1 of the main request, except that the penultimate paragraph relating to the "update" now reads as follows:

"... wherein the update is applied to modify the logic (104) configured to validate integrity of the verify logic (111) if the validation of the integrity of the verify logic (111) is successful and if the update (112) is successfully verified by the verify logic (111), ..."

Both requests also contain a corresponding independent method claim 4.

Reasons for the Decision

The invention

1. The application relates to ensuring that the contents of a "trusted component" in a computer system, for instance of a BIOS flash memory, can be updated in a safe manner (see paragraph 1 of the application as originally filed).

1.1 The system being considered (see figure 1) contains two "sections of memory", a "trusted" first (firmware) memory and less trusted second memory (see nos. 90, 100, 110, and page 2, lines 1-3 and 12-17). The firmware memory comprises a boot block, which contains "initialization logic" to be executed when the computing system is "powered on, restarted, and/or reset" (see nos. 102 and 103, and paragraph 8). The second memory holds the update for the content of the boot block (nos. 112, 115, 116).

1.2 During the boot-up procedure, a check is made whether an update exists in the memory 110 or in another storage medium (see paragraph 15 and no. 112) and, if so, the update is verified and installed (see paragraph 16).

1.3 For this purpose, the memory 110 contains "verify logic" (no. 111) which checks the integrity of the update. If the update is digitally signed, a method such as RSA digital signature verification may be used. To ensure that the verify logic itself has not been tampered with, a "hash logic" (no. 104) is provided in the boot block for verifying the integrity of the

verify logic (see paragraphs 11 and 16). This arrangement establishes a "chain of trust" from the manufacturer, who has stored the initial hash logic in the boot block, via the verify logic to the update code or data (see paragraph 11, last sentence).

1.4 The update could concern the hash logic itself (see paragraph 14).

The prior art

2. D1 relates to verifying the integrity of software before it is loaded or executed (see abstract). The situation considered in D1 is that of a computer system loading firmware or other data from an "attachable memory device" (see figure 1, nos. 16 and 20, paragraph 28). The computer system contains an "integrity checking algorithm" (no. 22) for confirming that the firmware has not been tampered with (see paragraphs 2 and 12). In order to validate the integrity checking algorithm itself, the computer system provides a further "integrity checker" which is stored in a ROM (nos. 14 and 24) or other "suitable memory device that is capable of storing a program or other code that is not intended to be erased or written over" (see paragraphs 10, 11 and 15). The integrity checker may execute a secure hashing algorithm (see paragraph 19) and it may "include" its parameters such as a checksum or a hash value (see paragraph 24).

Claim construction

3. The claims refer to a "trusted" component and a "trusted" memory.

3.1 In the board's view, the meaning of "trust" in this context has to be understood with reference to other system components.

3.2 The description states that "'trust' or 'trusted' means the expectation of consistent operation within a predefined set of rules that is enforced by computing hardware and/or software". For example, the "boot block 102 of firmware memory 100" may be trusted because it is ensured that it "contains only information produced by a previously-identified source" (paragraph 6, last sentence). Elsewhere, it is disclosed that the boot block is trusted because it "is locked within firmware memory 100 and is protected from updating during normal computing system 121 operation" and updated only "by trusted methods from a trusted source which can be validated" (see paragraph 8). The appellant has argued in this regard that this "definition" of trust is "in conformity with with the other claim features" (see the letter of 8 March 2018, page 2, paragraph 3).

3.3 The board takes the appellant's argument to be that any "trust" in the claimed "trusted component" and "trusted memory" is due to the integrity checks also claimed. In other words, the word "trusted" reflects an effect achieved by the other claimed features but does not, in itself, represent a limitation of the claims. This corresponds to the board's preliminary opinion (see the summons, point 6, in particular, lines 7-9).

3.4 The board concludes that the word "trusted" does not render the claims unclear but also that it need not be considered for the purposes of inventive step.

Inventive step

Main request

4. The examining division found (see the decision, reasons 3.2) that claim 1 of the then main request differed from D1 in that

(a) the logic (104) configured to validate integrity of the verify logic was disposed in a boot block (102) stored in the trusted memory, and
(b) the update was applied to modify the content of the book block (102) stored in the trusted memory.

4.1 These features were said to solve the objective technical problem of "providing a secure and trusted update to the content of the book block, and indirectly to logic (104)" (*loc. cit.*).

4.2 The appellant has not challenged this analysis (see the grounds of appeal, page 2, paragraphs 3-5). By way of amendment, however (see *loc. cit.*, paragraphs 6-9), claim 1 of both requests now also contains a further feature not known from D1, namely that

(c) the logic (104) is based on hashing the verify logic and uses a hash value that is also stored in the boot block (102).

5. The board agrees with the examining division and the appellant that D1 is a very relevant piece of prior art. However, the board tends to prefer not to *start* its analysis from D1.

5.1 The decision argued that, starting from D1, the invention solved the problem of "providing a secure and

trusted update to the content of the book block". The board is not persuaded that the skilled person starting from D1 would concern him/herself with this problem.

5.2 However, the board takes the view that in any prior art computer with a boot block, the need naturally arose to update its content in "a secure and trusted" manner. Hence the skilled person would have addressed the problem of updating the BIOS.

5.3 The board takes the view that the problem of updating the contents of a boot block, even though it contains feature (b), is not based on hindsight. No other technical problem is solved by updating the contents of the boot block, as opposed to other memory regions, than that the boot block contains content that needs updating. The board therefore considers it appropriate to consider the reference to the boot block as part of the objective technical problem rather than its solution. The board points out that the examining division also implicitly adopted this approach, and the appellant has not challenged it, not even in view of the corresponding explicit statement in the board's preliminary opinion (see the summons, point 8.3; the decision, reasons 3.2; the grounds of appeal, page 2, penultimate paragraph; and the appellant's letter of 8 March 2018, page 2, paragraph 2).

6. The skilled person addressing this problem, would, in the board's view, find D1.

6.1 The appellant has challenged this assumption, arguing that D1 would not give the skilled person any hint to use its teaching "to solve the objective technical problem of updating a boot block (main request) or a

logic to validate integrity of verify logic (auxiliary request)" (see the letter of 8 March 2018, page 2, paragraph 2).

6.2 The board, however, considers that the skilled person would not need such a hint to be contained in D1. The desire to update the contents of the boot block is, as just explained, considered part of the problem rather than the solution and thus does not require any further explanation. In D1 it is made clear that the disclosed "system and method for verifying integrity of software programs" does not depend on the nature of the software programs in question but is applicable to a large variety of "functional code, instructions, programs, databases or graphics ("firmware") ...", and target memory locations: "... that may [...] be downloaded from a network or otherwise loaded into a memory of a device" (see e.g. paragraphs 1, 12, 22 and 28).

6.3 In the board's judgment, it would therefore have been apparent to the skilled person that the two level integrity checking of D1 might provide a solution to the problem posed.

7. D1 discloses that the integrity checker may be stored in ROM.

7.1 This has the obvious advantage that the integrity checker cannot be tampered with (in a sense, the ROM is more "trusted" than a RAM) but at the price that it also cannot be updated if necessary. From this perspective, the appellant is correct in stating that D1 does not disclose or suggest any "downloading into the ROM 14" (see the letter of 8 March 2018, page 2,

last sentence, and page 3, paragraph 2), the ROM being a Read Only Memory.

7.2 That said, D1 does not disclose that what is referred to as a ROM must be a memory that *cannot* be written to. Rather, D1 discloses that "ROM 14 may be any suitable memory device that is capable of storing a program or other code that *is not intended* to be erased or written over" (see paragraph 11, sentence 1; emphasis by the board).

7.3 Hence D1 does not contain any express disclosure of updating "the integrity checker". However, the board takes the view that the reference to what is "intended to be erased or written over" does not constitute a technical limitation but expresses a design decision made in D1 for security reasons (the implementation of which, of course, requires technical means).

7.4 The board considers that the skilled person would, as a matter of course, have always been prepared to reconsider such design decisions, for instance when reviewing security requirements.

7.5 For instance, the board considers it to be common knowledge for the skilled person that hashing algorithms (as used in D1, see paragraphs 19 and 24) may themselves be cracked and thus may have to be replaced.

7.6 The skilled person would, therefore, have been aware that security might require the integrity checker itself and its parameters to be updated. It would also have been obvious for the skilled person to adapt the available updating method to that end and allow the

"firmware 20" to contain an update of the integrity checker and its parameters.

7.7 Of necessity, the skilled person would have had to store the integrity checker in a memory which could be written to, but would also have selected the memory so that the integrity checker could still be protected against tampering.

7.8 The board recalls at this point that D1 discloses that the integrity checker "include[s]" the required hash value (see paragraphs 19 and 24) and thus, in the board's view, at least suggests that the integrity checker and its hash values are stored in the same place.

8. In view of the foregoing, the present decision thus turns on the question of whether it would have been obvious to the skilled person to store the integrity checker - and the hash value - in the boot block.

8.1 The board agrees with the appellant that the appealed decision did not go into this point (see the grounds of appeal, page 3, paragraph 4, and the decision, reasons 3.3). However, the appellant has also not explained, neither in its grounds of appeal nor its letter of 8 March 2018, what problem is solved by placing the integrity checker in the boot block rather than in another area of memory, instead limiting itself to the observation that D1 did not suggest this option (see grounds of appeal, page 3, paragraphs 2-5).

8.2 In the scenario considered above, the boot block and the updating of its content would, as a central part of the objective technical problem, have been considered by the skilled person. Since, moreover, a boot block is

a well-protected area of memory, the board considers the boot block to have been an obvious choice for storing the integrity checker and its parameters for the skilled person seeking to also make them updatable.

8.3 Therefore, the board comes to the conclusion that features (a) and (c) would be obvious to the skilled person as a solution to the objective technical problem posed (see point 5.2 above) in view of D1 and common knowledge in the art (see point 7.5 above). Hence the subject-matter of claim 1 does not involve an inventive step, Article 56 EPC.

Auxiliary request

9. As set out above for the main request, the desire to update the integrity checker and hash value is a problem that would naturally have arise from common knowledge about the security of integrity checking (see again point 7.5 above). It would moreover have been straightforward to apply the teaching of D1 to them because, as also explained above, the updating method of D1 is applicable to any kind of software (see point 6.2 above). As a consequence, the subject-matter of claim 1 of the auxiliary request also does not involve an inventive step in view of D1 and common knowledge in the art, Article 56 EPC.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

S. Sánchez Chiquero W. Sekretaruk

Decision electronically authenticated