Internal distribution code:

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

Datasheet for the decision of 17 September 2014

Case Number: T 0741/11 - 3.5.06

Application Number: 04749735.9

Publication Number: 1631874

IPC: G06F1/00

Language of the proceedings: EN

Title of invention:
METHOD AND APPARATUS FOR ENCRYPTING DATABASE COLUMNS

Applicant:
Oracle International Corporation

Headword:
Transparent access to encrypted database columns/ORACLE

Relevant legal provisions:
EPC 1973 Art. 56
EPC Art. 123(2)

Keyword:
Inventive step -
main request, first and third auxiliary requests (no)
Added subject matter - second auxiliary request (yes)

Decisions cited:
T 1539/09

Catchword:

Beschwerdekammern
Boards of Appeal
Chambres de recours

European Patent Office
D-80298 MUNICH
GERMANY
Tel. +49 (0) 89 2399-0
Fax +49 (0) 89 2399-4465

Case Number: T 0741/11 - 3.5.06

DECISION
of Technical Board of Appeal 3.5.06
of 17 September 2014

Appellant: Oracle International Corporation
(Applicant)
500 Oracle Parkway
Redwood Shores, CA 94065 (US)

Representative: Skone James, Robert Edmund
Gill Jennings & Every LLP
The Broadgate Tower
20 Primrose Street
London EC2A 2ES (GB)

Decision under appeal: **Decision of the Examining Division of the European Patent Office posted on 18 November 2010 refusing European patent application No. 04749735.9 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman	D. Rees
Members:	M. Müller
	C. Heath

Summary of Facts and Submissions

I. The appeal lies against the decision of the examining division to refuse European patent application no. 04749735.9. The decision issued on 18 November 2010 and, for its reasons, referred to the communication dated 5 November 2010. This communication cited the following documents:

D2: WO 02/29577 and
D3: Garcia-Molina *et al.*, "Database Systems: The Complete Book", pages 788-795, Prentice Hall, 2001,

and argued that the claimed invention lacked an inventive step over D2 in view of common knowledge exemplified by D3, Article 56 EPC 1973.

II. A notice of appeal against this decision was received on 12 January 2011, the appeal fee was paid on 17 January 2011, and a statement of grounds of appeal was filed on 18 March 2011. The appellant requested that the decision be set aside and a patent be granted on the basis of two sets of claims according to a main or an auxiliary request filed with the grounds of appeal. It asked that the adaptation of the specification over the application documents on file be postponed until after an allowable set of claims was agreed.

III. With a summons to oral proceedings, the board informed the appellant about its preliminary opinion according to which the claimed invention lacked an inventive step, Article 56 EPC 1973, over D2 in view of common knowledge in the art, as illustrated partly by D3. Objections under Articles 123 (2) EPC and 84 EPC 1973 were also made.

IV. In response to the summons, with letter dated 20 May 2014, the appellant filed three sets of claims according to a main request and first and second auxiliary requests and corresponding description pages 3a and 3b for each request, and gave arguments in favour of inventive step.

V. On 17 September 2014, the oral proceedings took place as scheduled. During the oral proceedings, the appellant filed new method claims as a second auxiliary request 1-8 and made the previous "second auxiliary request" the third one. It was stated that the system claims lacking from amended second auxiliary request might be filed in correspondence to the method claims if and once the method claims were found allowable by the board. When the board had doubts about original disclosure of some of the new features, the appellant declared itself willing to delete these features from the pertinent auxiliary request. A so-amended further request was however not formally filed.

VI. The final application documents, pending adaptation of the description and addition of the system claims of the second auxiliary request, thus were the following:

claims, no.

1-17	main request, filed with letter of 20 May 2014,
1-15	first auxiliary request, filed with letter of 20 May 2014,
1-8	second auxiliary request, filed during oral proceedings, and
1-13	third auxiliary request filed as "second auxiliary request" with letter of 20 May 2014,

description pages

1, 2, 4, 6-14	as published,
5	received with letter of 2 November 2006,

3 received with letter of 15 May 2008,
3a, 3b received, respectively, for the main, first and third auxiliary requests with letter of 20 May 2014, and
drawings, sheets
1-4 as published.

VII. Claim 1 of the main request reads as follows.

"A method of operating a server (104) for facilitating, for a database client (102), transparent encryption and decryption of data on a column-by-column basis within a database (106) accessed by the server (104), the method characterised by:

receiving (302), at a server (104) from client (102), a command statement in a database language to perform a database operation;

before executing the command statement, the server (104):

parsing (304) the command statement to create a parse tree, the parse tree having elements comprising operators and column attributes;

examining (306) the parse tree to determine (310) if a column attribute references in the parse tree refers to an encrypted column; and

if so

automatically transforming (312) elements of the parse tree to include one or more cryptographic operators from the group of a decrypt operator, an encrypt operator, and a key retrieval operator, the server (104) being configured to execute a database operation in dependence of the parsed command statement with transformed elements of the parse tree to facilitate accessing the encrypted column while performing (314) the database operation in a way transparent to the client (102)."

Claim 1 of the first auxiliary request is identical to claim 1 of the main request except for the following passage added to its end:

"... wherein examining the parse tree further comprises the server (104):

determining if the command statement includes an explicit command to change an encryption algorithm for the column; and

if so

decrypting the column using a previous encryption algorithm, and encrypting the column using a new encryption algorithm."

Claim 1 of the second auxiliary request is based on claim 1 of the first auxiliary request by seven insertions and one omission which, in the following, are marked by underlining and strikeout, respectively.

"A method of operating a server (104) for facilitating, for a database client (102), transparent encryption and decryption of data on a column-by-column basis within a database (106) accessed by the server (104), the method characterised by:

receiving (302), at a client interface of [sic] server (104) from client (102), a command statement in a database language to perform a database operation, wherein the client is operable as a source of commands that includes commands for performing reference operations on the database and update operations on the database, the update operation including an operation to update the values within a column of the database by multiplying the values within the column by a constant value;

before executing the command statement, the server (104):

sending the command statement to a command parser and parsing (304), by the command parser, the command statement to create a parse tree, the parse tree having elements comprising operators and column attributes;

examining (306) the parse tree to determine (310) if a column attribute references in the parse tree refers to an encrypted column; and

if so

automatically transforming (312), by a command transformer of server (104), elements of the parse tree to include one or more cryptographic operators from the group of a decrypt operator, an encrypt operator, and a key retrieval operator; and sending the transformed command to a database interface of the server that is~~, the server (104) being~~ configured to execute a database operation in dependence of the parsed command statement with transformed elements of the parse tree to facilitate accessing the encrypted column while performing (314) the database operation in a way transparent to the client (102).

wherein examining the parse tree further comprises the server (104):

determining if the command statement includes an explicit command to change an encryption algorithm for the column; and

if so the database interface using the transformed command to perform the operations of:

decrypting the column using a previous encryption algorithm, and encrypting the column using a new encryption algorithm."

Claim 1 of the third auxiliary request is identical to claim 1 of the first auxiliary request except for the following passage added to its end:

"... wherein if the database operation includes a reference operation from the encrypted column, the method further comprises the server (104) transforming (312) the database operation to decrypt data retrieved from the encrypted column during the reference operation."

The main request and the first and second requests contain system claims which correspond closely to the respective method claims.

VIII. At the end of the oral proceedings, the chairman announced the decision of the board.

Reasons for the Decision

The invention

1. The application generally addresses the problem of facilitating the handling of database systems with column-wise encrypted data (see original application, p. 2, lines 8-10). In such databases systems individual columns may or may not be encrypted and, if they are, different encryption parameters (*e.g.* hashing and encryption algorithms, encryption key) may be used for different columns (see original application, *e.g.* pars. 0042-0045).

1.1 Encryption and decryption are handled in a "transparent" manner with respect to "the application developer" or "the user" (par. 0006), to "applications that access" the database (par. 0041) or to "the client" (par. 0027, present claim 1 of all requests). Transparency is specifically disclosed to mean that a command accessing a database column need not reflect whether the column is encrypted or not and, even if so, need not contain explicit encryption and decryption commands (see also par. 0005). When executing a command, the database server will determine the need for encryption and/or decryption and perform the necessary operations automatically.

1.2 The claimed invention according to all requests refers, in particular, to a "command statement" which the server receives from a client (fig. 1) and parses to create a parse tree. The server examines the parse tree to determine whether it refers to an encrypted database column and, if so, automatically "transform[s] ... elements of the parse tree to include ... cryptographic operators"; this transformation effectively determines the database operation to be executed (see par. 0053 and fig. 2).

The prior art

2. The application discusses a prior art solution to the problem of providing transparent access to a column-wise encrypted database which is based on "views" and "triggers" (par. 0006). This solution is based on the idea of providing an unencrypted database "view" to "hide the cryptographic functions" from the user and the use of "triggers" so that an update to this view causes the data in the base table to be encrypted implicitly. Disadvantages of this solution are discussed.

3. D2 refers to the problem of dealing with sensitive information in a "[m]odern database system" (p. 1, line 16).

3.1 As a solution, it discloses a database system in which encryption is handled "automatically and transparently to a user" (p. 2, lines 20-21). Specifically, if it is requested to *store data* in a database column which has been "designated ... as an encrypted column", "the system" - *i.e.* the database server (p. 5, lines 7-9, fig. 1) - automatically encrypts the data", using the appropriate key retrieved from a keyfile in the database system (p. 2, lines 21-26; p. 9, lines 7-18) and possibly based on further encryption parameters such as encryption mode, key length, and integrity type retrieved from column "metadata" (p. 3, lines 25-31). If it is requested to *retrieve data* from an encrypted database column, "the system allows the ... user to decrypt the encrypted data" using the appropriate key, provided the user is authorized accordingly (p. 2, lines 27-31; p. 9, line 20 - p. 10, line 9). D2 refers to "requests" which the database server "receives" from the clients but does not disclose their specific form or formats.

3.2 The focus in D2 lies on the protection of sensitive data against a malicious database administrator by distributing administration tasks across three distinct administrator "roles" (see p. 6, lines 1-3). Specifically, it is disclosed that a "security administrator ... manages the encryption system through database server" by, *inter alia*, "specifying which columns in the database are encrypted" (see p. 5, lines 26-28 and fig. 1) and "select[ing] the mode of encryption" and "establishing encryption parameters" (p. 3, lines 3-4 and 25-28). It is disclosed that the administrators are not "authorized users" and thus "prevented from decrypting and receiving encrypted data" (p. 10, lines 6-9).

4. D3 is an excerpt of a standard textbook on databases relating to "query compilation" (p. 788, sec. 16.1, 1st sentence): It is disclosed that a "text written in a language such as SQL" (sec. 16.1.1, 1st sentence), *i.e.* a database command, is parsed into a parse tree and then transformed into an "expression in relational al-

gebra" (see par. below fig. 16.1). This expression, the "initial logical query plan", is further transformed so as to yield an "improve[d]" or "preferred logical query plan" (*loc. cit.* and fig. 16.1).

D2 as a starting point for assessing inventive step.

5. The appellant argued that D2 was fundamentally different from the claimed invention. These differences were, in fact, so significant that D2 should be considered as an accidental anticipation from a different field than that of the invention. D2 thus was unsuitable as a starting point for assessing inventive step of the present invention and, if used nonetheless, taught away from it.

5.1 Specifically, the appellant argued that D2 was "not a command-based system", whereas it was central for the invention to operate and transform database commands. The system of D2, so the argument, was a "simple request based system in which a user [could] only store and retrieve data from a database" (see letter of 20 May 2014, point 3.8). The requests of D2 were not "commands" but only means to trigger one of two preprogrammed processes. In support for this interpretation, the appellant referred, in particular, to figures 1, 6, and 7 of D2.

5.2 The appellant also argued that the "use of parsable commands [was] only known in systems directed towards providing user operating through a client with a high level of functionality" (see letter of 20 May 2014, point 3.9) whereas "[t]he purpose of D2 [was] to increase the security of the user's data" which came "at the expense of reduced functionality" (point 3.12). D2 thus directly taught the skilled person away from providing increased functionality using parsable commands.

6. The board does not share this interpretation of D2.

6.1 It is conceded that the main focus of D2 is on database security. However, the security problem addressed in D2 is formulated in the context of unspecified "[m]odern database systems". Also, it is not disclosed that the proposed solution requires any changes in the database architecture beyond, obviously, the distribution and separation of privileges amongst the roles of the administrators and users. Nor does D2, in the board's view, imply that such changes were required.

6.2 D2 discusses database access only in generic terms by talking about requests to "store" and "receive data". In the board's understanding this does not, however, limit the ways in which requests may be expressed: after all, storing and receiving data are the fundamental operations on any database (*i.e.* writing or reading). A more complex operation such as updating the values in a column by multiplying them by a constant value can easily be reduced to ("receiving") reading data from the database, processing it, and writing it back to ("storing" it in) the database. The brevity of D2 regarding the form of the requests and the interface with which they are issued are, in the board's understanding, due to the fact that they are not relevant in D2 for the security issue at stake and for presenting the proposed solution. While this brevity obviously leaves undefined many features of the database system, the board does not agree that it establishes a prejudice against specific forms of requests or interfaces.

6.3 Specifically, the board disagrees that D2 teaches away from using the proposed security architecture in a database system using SQL, *i.e.* an expressive "command-based" system in the appellant's terms.

7. The appellant argues that other documents cited in the European or the International phase should have been used instead of D2 as a starting point for assessing the inventive step. These documents corresponded to the prior art based on "triggers" and "views" as discussed in the application (and summarized above, point 2) and on which the invention is meant to improve. These documents were neither specifically discussed during the appeal procedure, nor does the board consider this to be necessary: since the board deems D2 to be a suitable starting point for assessing inventive step and is in a position to come to a conclusion on inventive step in view of D2, it may be left open whether there are other, even possibly more suitable starting points for this assessment.

Inventive step, Article 56 EPC 1973

8. The independent claims refers to "commands" which are "parsed". The skilled person would understand this to imply that the commands are expressions in some sort of database query language. D2 refers to "requests" to store and to retrieve data but leaves open how these requests are generated and in which form. Moreover, as the appellant points out, D2 is silent as to "whether or not the client has to explicitly specify the cryptographic functions of the server on storing or retrieving data from the database" (see grounds of appeal, reasons 6.7).

8.1 Claim 1 of the main request thus differs from D2 by the following features:

a) Database requests are expressed as "commands" which can be - and are - parsed, and

b) the parse trees (or rather: elements thereof) obtained from a database command are "automatically transform[ed] ... to include one or more cryptographic operators" such as "a decrypt operator [or] an encrypt operator".

8.2 The board agrees with the appellant that these features can be seen to solve the problem of "how to facilitate client interaction with a column-by-column encrypted database" (see grounds of appeal, point 7.7).

Re. difference a)

9. The board considers that it was common practice well before the present priority date to interact with databases via "requests" in the form of "commands" in some database query language such as SQL. It was further commonly known that such commands had to be parsed (see also the textbook excerpt D3 which establishes this). During oral proceedings, the appellant confirmed that such command-based database systems were conventional at the time. As argued above, however, the board does not share the appellant's opinion that D2 is incompatible with such a command-based system. To the contrary, the board considers it to be an entirely obvious option for the database in D2 to be command-based.

Re. difference b)

10. The board notes that D3 also discloses that the parse tree is checked and that, in that process, "each attri-

bute" is "resolve[d]" by "attaching it to the relation to which it refers" (see p. 794, point 2, lines 6-9). In the board's view, this does not unambiguously disclose a "transformation" of the parse tree to include that additional information. The transformations actually disclosed are from parse trees into expressions of relational algebra and between such expressions (p. 795, 16.2). D3 thus does not disclose difference b).

10.1 With regard to D2, the board is not convinced by the appellant's argument that "from reading D2 the skilled person would undoubtedly think that the cryptographic functions should be explicitly included in the client ... requests". Specifically, the passage cited by the appellant referring to "a user ha[ving] designated the column as an encrypted column" does not imply that the client request has to "include a designation of encryption" (see grounds of appeal, point 6.8). As argued in the summons (point 11), the board tends to consider that the skilled person would understand D2 to mean that cryptographic functions are not part of the database storage and retrieval requests issued by the client.

10.2 However, *arguendo*, let it be assumed to the appellant's benefit, that D2 taught or suggested that the cryptographic functions were explicitly specified in the database requests. In this case the user would have to keep track of which database columns are encrypted and how, and which are not. Moreover, the required commands would be complex to write and difficult to read: See, for instance, the command disclosed in the application (p. 9, lines 10-13). In this situation the board considers it to be an obvious desirable to simplify the users' task by relieving them from having to specify the cryptographic operators explicitly.

10.3 An obvious and common solution to this type of problem is to change the *semantics* of the commands in question by leaving certain parameters implicit. In the present case one would, for example, define a command reading "store v in column c" to *mean* "if c is an encrypted column then encrypt v and store the result in c, otherwise store v directly". The board considers that modifying the semantics of commands in itself does not solve any technical problem (see T 1539/09, headnote).

10.4 Beyond that, the board deems it to be common practice in the art of programming languages to simplify commands by leaving certain parameters implicit and have the compiler add the missing information. For illustration note that in C the required type conversion from an integer (say, 1) to a floating point number (say, 2.5) in a mixed-type addition such as 1+2.5 is left implicit and generated "transparently" by the compiler ("implicit type conversion").

10.5 If, as is the case according to D2, the cryptographic parameters are known to the server - in a keyfile or column metadata - it would have been obvious to the skilled person that they *can* be retrieved automatically if needed and thus that they need not be specified explicitly in commands.

11. It remains to be considered whether it would have been obvious for the skilled person to implement, in the system of D2, the handling of commands which did not specify the cryptographic operations but left it for the server to add, in the manner claimed.

11.1 In a database command the column names are what is called "identifiers" in programming language terminology. The parser performing a *syntactic analysis* of the given command recognizes identifier names. However, further information about an identifier often cannot be determined during parsing: for instance the type of an identifier may have been declared in a different comman d. The same applies to the names of database columns: different databases may have columns with the same name (see also D3, p. 793 ff., sec. 16.1.3, esp. point 2, lines 6-9), and whether a column is encrypted is part of the database definition rather than the command. As a consequence, identifiers are commonly processed *after parsing* in a phase referred to as *semantic analysis*. During this phase, the parse tree is commonly annotated ("attributed", "decorated") with the derived semantic information. The board considers that this "automatically transform[s] elements of the parse tree" as claimed.

11.2 Whether or not a database column identifier refers to an encrypted database column or not, and if so, what cryptographic parameters are to be used, are, in the board's view, obvious semantic "attributes" of column identifiers which can, as the skilled person would have noted, naturally be determined during the semantic analysis just described.

12. The appellant argued that "although these techniques may be generally known for the given example of an arithmetic compiler, there is no teaching that would lead the skilled person to implement such techniques in the specific field of encryption" (see letter of 20 May 2014, point 3.11). The board points out, however, that the example was expressly given as a mere illustration for a technique which the board deems to belong to the general knowledge in compiler technology. Parsing and semantic analysis of commands is largely a matter of command and language structure and is independent of

whether the operations represented by the commands relate to arithmetic, database management or encryption. During oral proceedings, the board stressed that it considered the claimed technique of transforming a parse tree to belong to the common knowledge in the art of parsing and compiling, and the appellant did not challenge the board on this point.

13. In summary, the board comes to the conclusion that the subject matter of claim 1 of the main request lacks an inventive step, Article 56 EPC 1973, over D2 in view of common knowledge in the field of parsing and compilation.

First auxiliary request

14. The independent claims of the auxiliary request comprise the additional features that the server determines, based on the parse tree, whether the command "includes an explicit command to change an encryption algorithm for the column" and, if so, decrypts the column using "a previous algorithm" and encrypts it using the new encryption algorithm.

14.1 In the board's understanding these features primarily express the requirement that a command to change the encryption algorithm for a column is provided at all. The last two lines of claim 1 (or, correspondingly, the last four lines of claim 9) merely state that this command is executed. That prior to execution this command is "determined" by "examining the parse tree" is considered to be common practice in the art.

14.2 The board considers it obvious that the security administrator of D2, responsible for selecting mode and parameters of encryption (p. 3, lines 3-4 and 25-29), may have to change the encryption algorithm for a column for various reasons, for instance if the security of an encryption algorithm has been compromised.

14.3 The appellant argued that D2 disclosed a strict separation of tasks between users and the security administrator and that the security administrator performed its duties directly at the server and not through a client. It would therefore not be obvious from D2 to provide a command for changing the encryption algorithm. Moreover, the appellant argued that the term "client" in the present application was disclosed as synonymous with "user" which would clearly exclude the security administrator. Claim 1 thus had to be construed as equipping the end user with the capability of changing the encryption algorithm which was specifically discouraged in D2 in which the management of encryption was the exclusive task of the security administrator.

14.4 The board disagrees. Firstly, it is noted that the term "client" is explicitly disclosed in the application to be a "node on a network" (par. 0024) and thus does not denote the "user" but a terminal from which the user accesses the system. Secondly, the system administrator according to D2 is also a user: D2 discloses that the security administrator may issue requests like a normal user even though it will be found not to be authorized for reading encrypted data (p. 10, lines 5-9). Thirdly, D2 lacks any detail as to how - *i.e.* via which kind of interface - the security administrator performs its primary duties.

14.5 The board considers it as an obvious option to provide commands also for the tasks of a security administrator and sees nothing in D2 that would prohibit or just discourage this: the separation of powers according to D2 could be implemented by simply not authorizing the execution of the command for changing the encryption algorithm when issued by an end user; a suitable authorization mechanism is already available in D2 (*loc. cit.* and p. 9, lines 25-26).

14.6 Furthermore, the board considers it obvious to enable the security administrator to perform its tasks not only directly at the server but also from a client terminal, independent of whether the terminal is exclusive to the security administrator or shared with end users.

14.7 Thus the board comes to conclusion that the additional feature of the first auxiliary request constitutes the obvious implementation of an obvious new command. Hence, the independent claims of the auxiliary request also lack an inventive step, Article 56 EPC 1973.

Second auxiliary request

15. The appellant argued that the amendments were originally disclosed in the application on page 6, lines 5-7 and 24-25, page 7, line 28 - page 8 line 24 and in figures 1 and 2.

15.1 The board is not convinced that these passages disclose the last two of these amendments, namely the new features

F) "sending the transformed command to a database interface of the server" and

G) "the database interface using the transformed command to perform the operations of" decrypting ... and encrypting, nor is it aware of any other basis in the original application.

15.2 These features are meant to clarify that the command parser and transformer running on the server act as "middle-ware" between two interfaces, a "command interface" and a "database interface" so that the database interface need not be changed in order to make transparent to the user how encryption of database content is handled. This architecture was depicted in figure 2.

15.3 The board considers that the terms "database" and "database interface" are, in themselves, rather broad terms. The database could refer to the mere collection of data or to the data collection in combination with pertinent software for database access and/or administration. Likewise, the database interface could merely enable access to the raw data or also to further support functionality.

15.4 The board notes that the application uses the term "database interface" only in relation to figure 2 which depicts it with reference number 210 (pars. 0028 and 0032). It does not however define the "database interface" nor does it, in particular, disclose what the database interface is arranged to do or how: All it says is that "[d]atabase interface 102 includes mechanisms for accessing database 106", which "accessing operations can include retrieving data from database 106 and storing or updating data within database" (par. 0032). Hence, the board considers that feature G, according to which the database interface performs decryption and encryption, is not disclosed in the application as originally filed.

15.5 Moreover, figure 2 contains an arrow pointing from the command transformer 206 to the database interface 102 but the meaning of this arrow is nowhere specifically discussed (see pars. 0028-0032). While it appears to relate to some kind of data flow between the command transformer and the database interface, it remains open whether the entire "transformed command" is actually transferred to the database interface, as feature F requires, or only relevant parts of it. Therefore, also feature F is not, in the board's judgment, disclosed in the application as originally filed.

15.6 As a consequence, claim 1 of the second auxiliary request does not conform with Article 123 (2) EPC.

15.7 In passing, the board notes that the precise separation of tasks between the server and the database interface appears not to be disclosed in the application as filed and for that reason the appellant's "middle-ware" argument (see point 15.2) fails not only for present claim 1 but appears not to have a basis in the entire application as originally filed.

16. In response to this objection, the appellant requested the board to consider, as a potential further request, a claim corresponding to claim 1 of the second auxiliary request without the additional features F and G.

16.1 The board is satisfied that a so-amended claim does not go beyond the application as originally filed, Article 123 (2) EPC.

16.2 However, the remaining additions are insufficient to change the board's assessment of claim 1 as to the inventive step. The board considers it implicitly disclosed in D2 that the client request is received at the server via a suitable interface, *i.e.* a "client interface at" the server. That the request is received in the form of a "command", which is "parsed" and then "transformed" has already been discussed above with regard to the main request and found not to be inventive over D2. Finally, the specifically claimed commands "for performing reference operations on the database and update operations" are considered to be common-place operations which the skilled person would support in a conventional database as a matter of course and which, as argued above, do not conflict with the security architecture of D2.

16.3 Therefore, also claim 1 of the second auxiliary request without feature F and G lacks an inventive step over D2, Article 56 EPC 1973.

Third auxiliary request (filed as "second" on 20 May 2014)

17. Claim 1 of the third auxiliary request incorporates into claim 1 of the first auxiliary request the features of original claim 2.

17.1 The appellant argued that the new features established that both kinds of commands could be executed on request from "the same source" (see letter of 20 May 2014, point 7.1) or indeed, as the appellant clarified during oral proceedings, from the same person. This aspect was relevant, so the argument, because D2 disclosed (*loc. cit.*) that the selection of encryption algorithm and parameters was the exclusive right of the security administrator who, however, was not allowed to access encrypted database content, so that D2 specifically taught away from both commands coming "from the same source".

17.2 The board considers it to be clear - also in view of original claims 1 and 2 - that "the database operation" mentioned by the added features refers to the transformed database access command rather than the command to change the encryption.

17.3 The board also notes that the wording of amended claim 1 does not imply there to be a single complex command which contains subcommands for database access and for changing the encryption, let alone that both these operations may actually be authorized and executed in response to a single such complex command. In response to the board's question during oral proceedings, the appellant confirmed this interpretation.

17.4 The board thus considers that claim 1 only establishes that the server is equipped to handle both kinds of commands but does not exclude that they are issued from different persons at possibly different clients at different points in time. In this respect, the board disagrees with the appellant and considers that the amendment does not add anything substantial to claim 1 of the first auxiliary request.

17.5 As a consequence, claim 1 of the third auxiliary request also lacks an inventive step, Article 56 EPC 1973.

Remark

18. A central argument by the appellant was that the invention contradicted the security architecture of D2 because it allowed end users to access encrypted database content *and* to perform security management functions. This argument already failed in the present case because the claims, in the board's judgment, are consis-

tent with the security architecture of D2, *i.e.* access and security management functions being assigned to different roles. Moreover, the appellant was unable to propose, and the board equally did not see, any potential amendment of the claims which would have a basis in the application as filed and would not be consistent with the security architecture of D2. Therefore, it was not and did not have to be decided what impact on the inventive step analysis the alleged deviation from the security architecture of D2 might have had.

Order

For these reasons it is decided that:

The appeal is dismissed.

The Registrar: The Chairman:

Beschwerdekammern des Europäischen Patentamts
Boards of Appeal of the European Patent Office
Chambres de recours de l'Office européen des brevets

B. Atienza Vivancos D. Rees

Decision electronically authenticated