US 20060259950A1

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2006/0259950 A1**

Mattsson (43) **Pub. Date: Nov. 16, 2006**

(54) **MULTI-LAYER SYSTEM FOR PRIVACY ENFORCEMENT AND MONITORING OF SUSPICIOUS DATA ACCESS BEHAVIOR**

(76) Inventor: **Ulf Mattsson**, Cos Cob, CT (US)

Correspondence Address:
FISH & RICHARDSON PC
P.O. BOX 1022
MINNEAPOLIS, MN 55440-1022 (US)

(21) Appl. No.: **11/357,741**

(22) Filed: **Feb. 17, 2006**

Related U.S. Application Data

(60) Provisional application No. 60/654,181, filed on Feb. 18, 2005.

Publication Classification

(51) **Int. Cl.**
G06F 17/00 (2006.01)

(52) **U.S. Cl.** **726/1**

(57) **ABSTRACT**

A method for controlling data access in a data-at-rest system includes executing a link intrusion prevention analysis between multiple layers of the data-at-rest system, introducing a privacy policy at enforcement points that span multiple system layers, and dynamically altering the privacy policy.

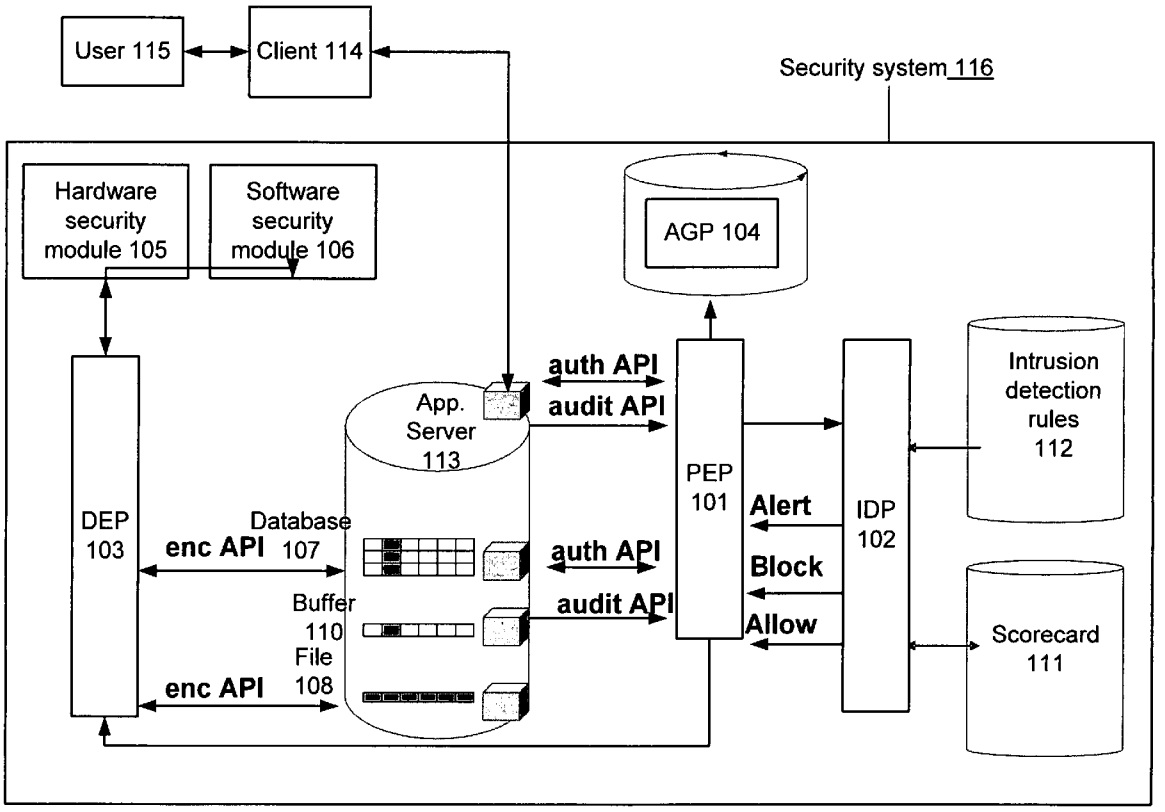


FIG. 1

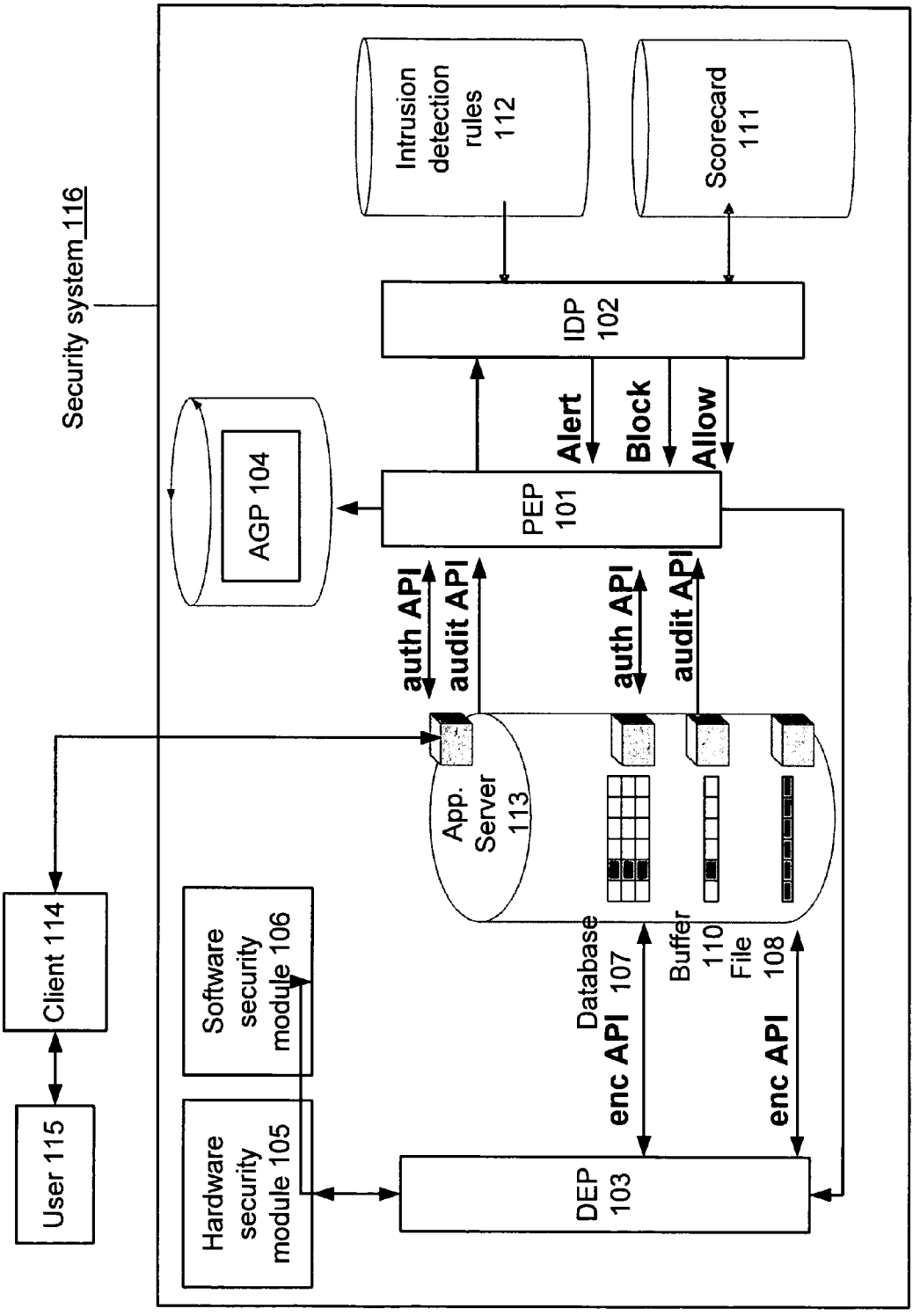

Security system 116
Intrusion detection rules 112
Scorecard 111
IDP 102
Alert
Block
Allow
AGP 104
PEP 101
auth API
audit API
auth API
audit API
App. Server 113
Client 114
User 115
Software security module 106
Hardware security module 105
Database 107
Buffer 110
File 108
enc API
enc API
DEP 103

FIG. 2A

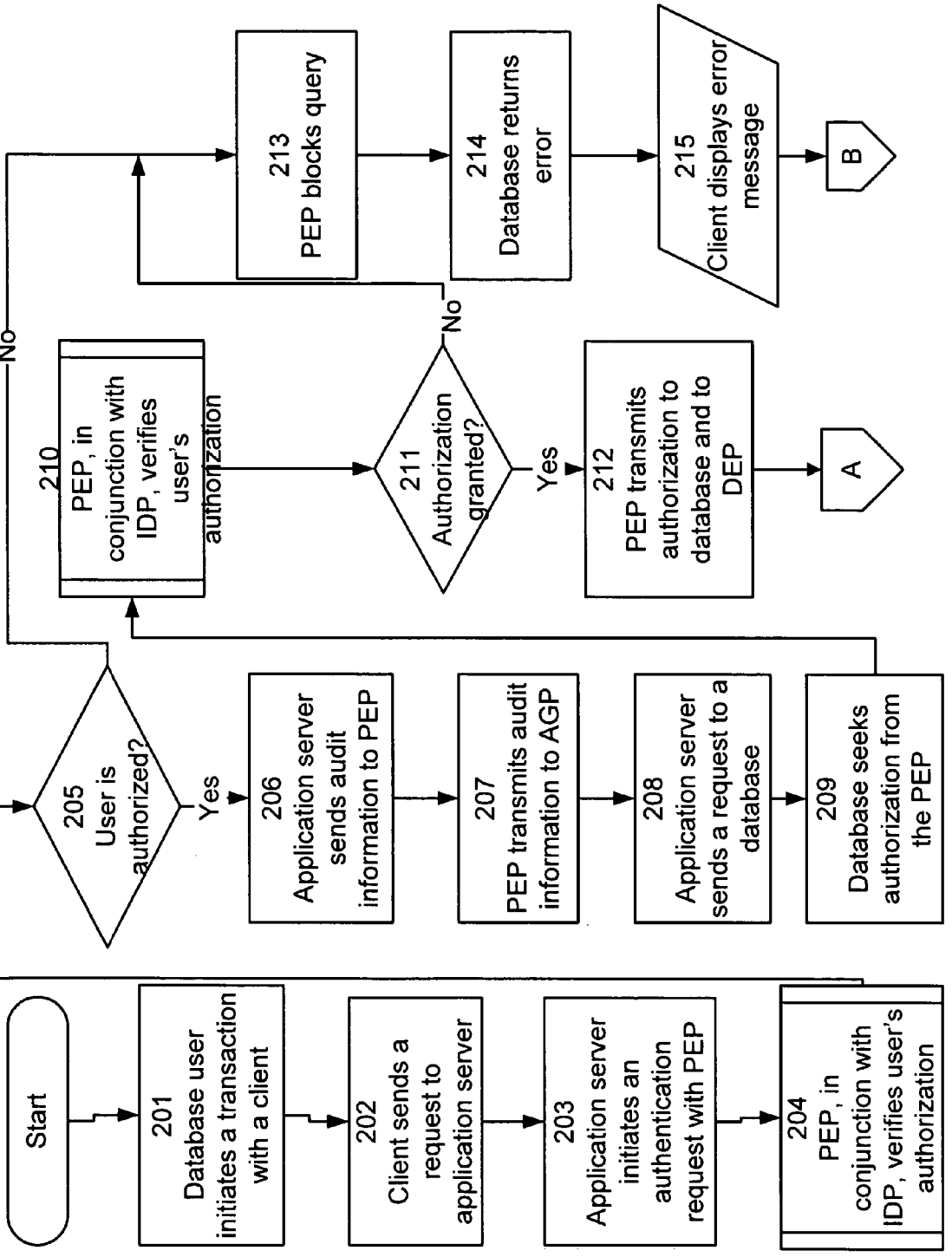

Start
201 Database user initiates a transaction with a client
202 Client sends a request to application server
203 Application server initiates an authentication request with PEP
204 PEP, in conjunction with IDP, verifies user's authorization
205 User is authorized?
Yes
No
206 Application server sends audit information to PEP
207 PEP transmits audit information to AGP
208 Application server sends a request to a database
209 Database seeks authorization from the PEP
210 PEP, in conjunction with IDP, verifies user's authorization
211 Authorization granted?
Yes
No
212 PEP transmits authorization to database and to DEP
213 PEP blocks query
214 Database returns error
215 Client displays error message
A
B

FIG. 2B

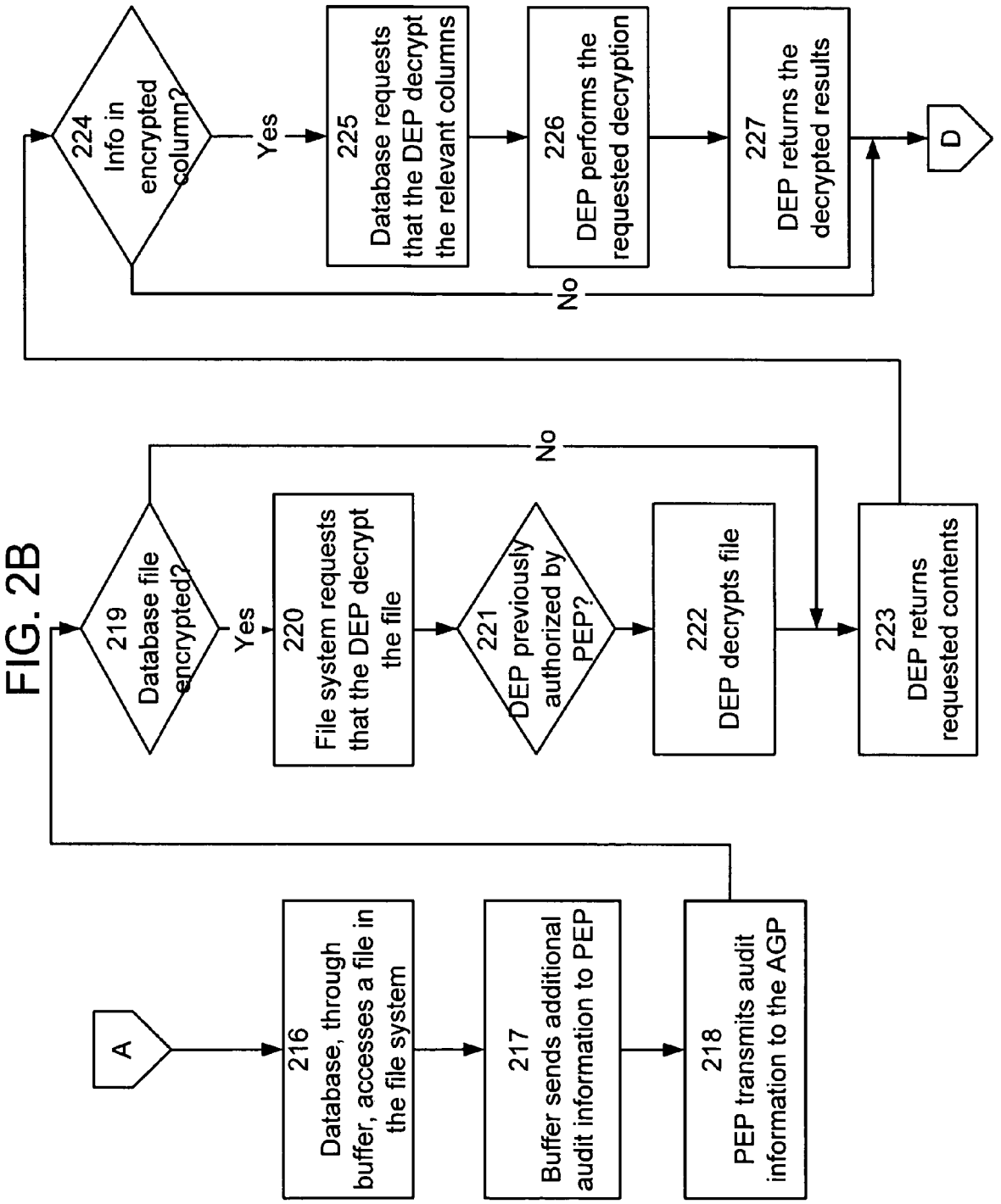

A
216
Database, through buffer, accesses a file in the file system
217
Buffer sends additional audit information to PEP
218
PEP transmits audit information to the AGP
219
Database file encrypted?
Yes
No
220
File system requests that the DEP decrypt the file
221
DEP previously authorized by PEP?
222
DEP decrypts file
223
DEP returns requested contents
224
Info in encrypted column?
Yes
No
225
Database requests that the DEP decrypt the relevant columns
226
DEP performs the requested decryption
227
DEP returns the decrypted results
D

FIG. 2C

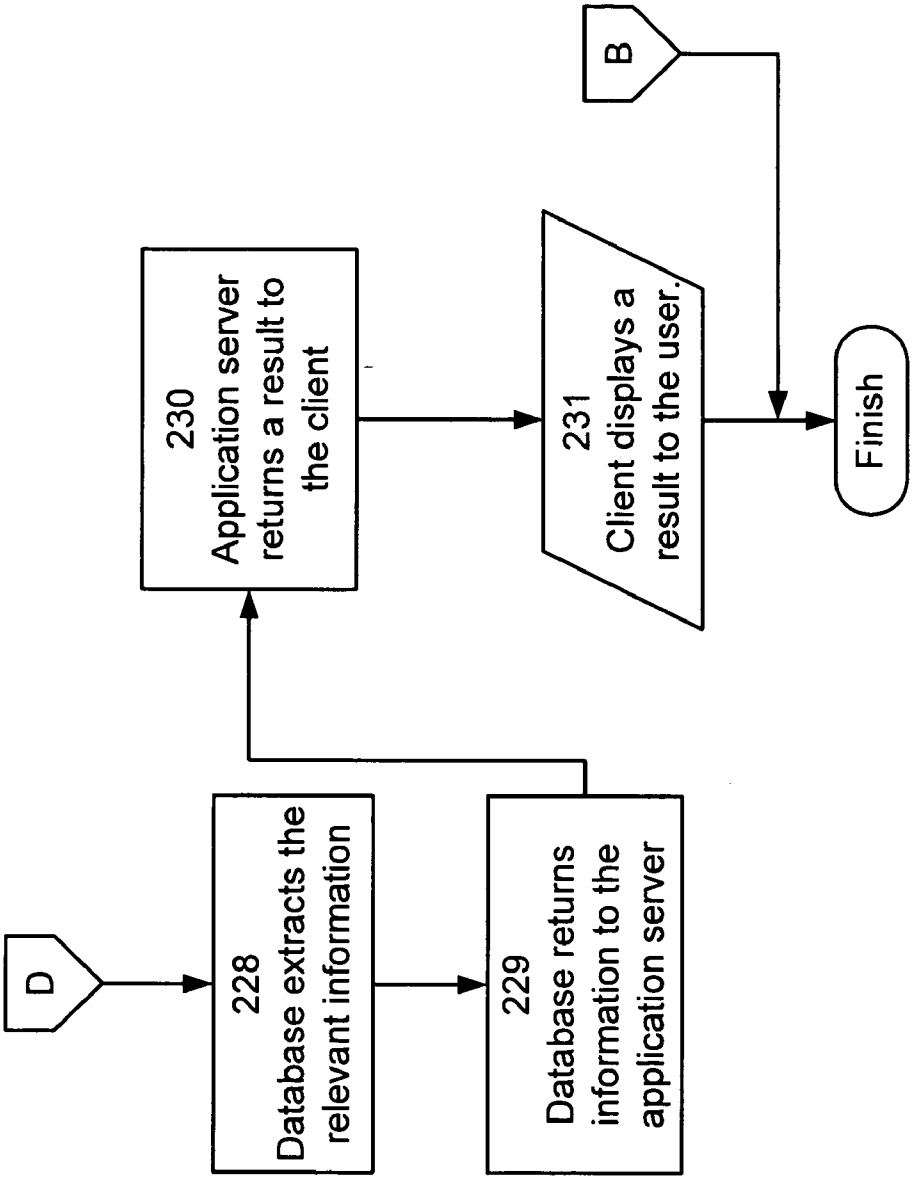

D
228
Database extracts the relevant information
229
Database returns information to the application server
230
Application server returns a result to the client
231
Client displays a result to the user.
B
Finish

FIG. 3

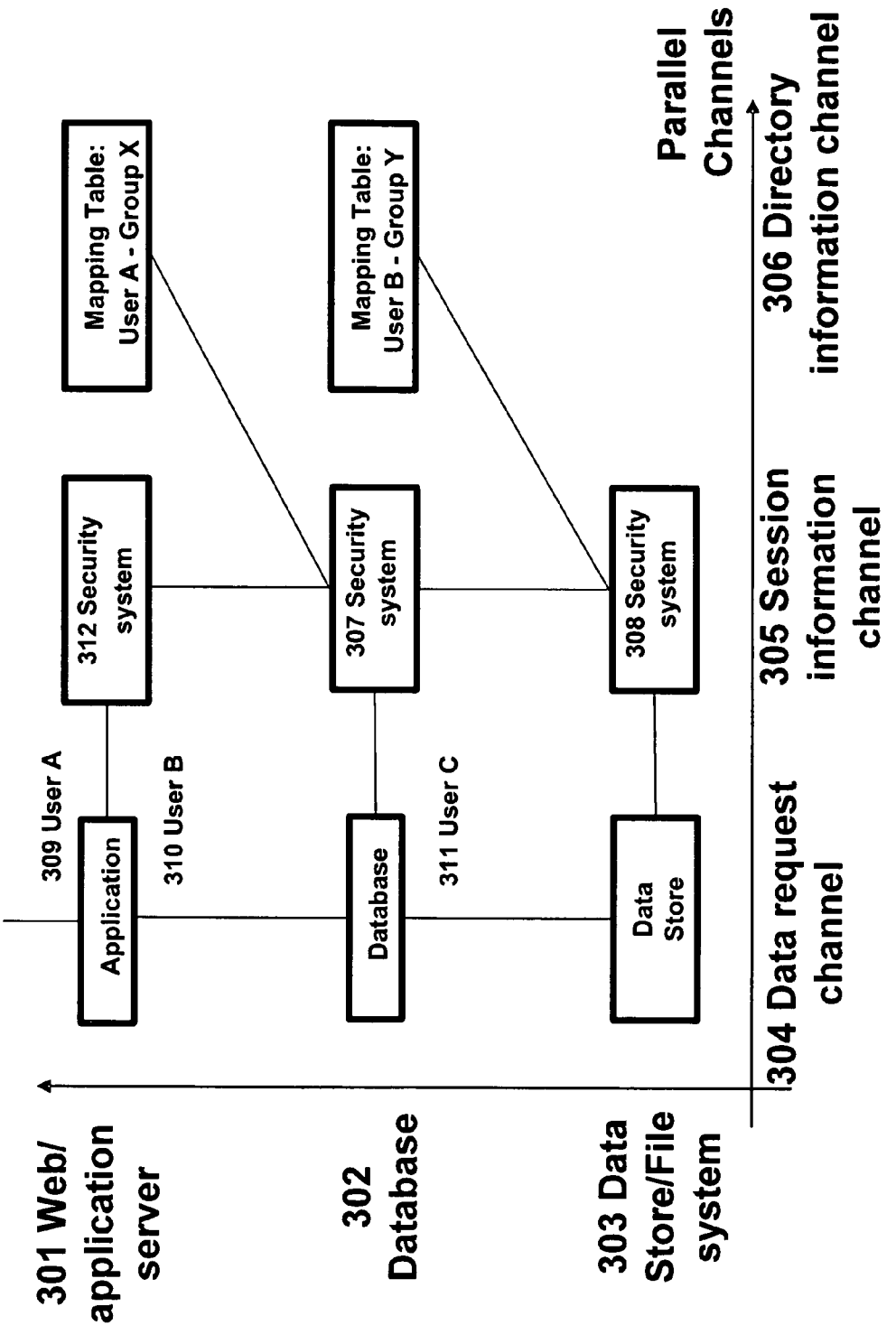

Parallel Channels
306 Directory information channel
305 Session information channel
304 Data request channel
301 Web/ application server
302 Database
303 Data Store/File system
Mapping Table: User A - Group X
Mapping Table: User B - Group Y
312 Security system
307 Security system
308 Security system
309 User A
310 User B
311 User C
Application
Database
Data Store

FIG. 4A

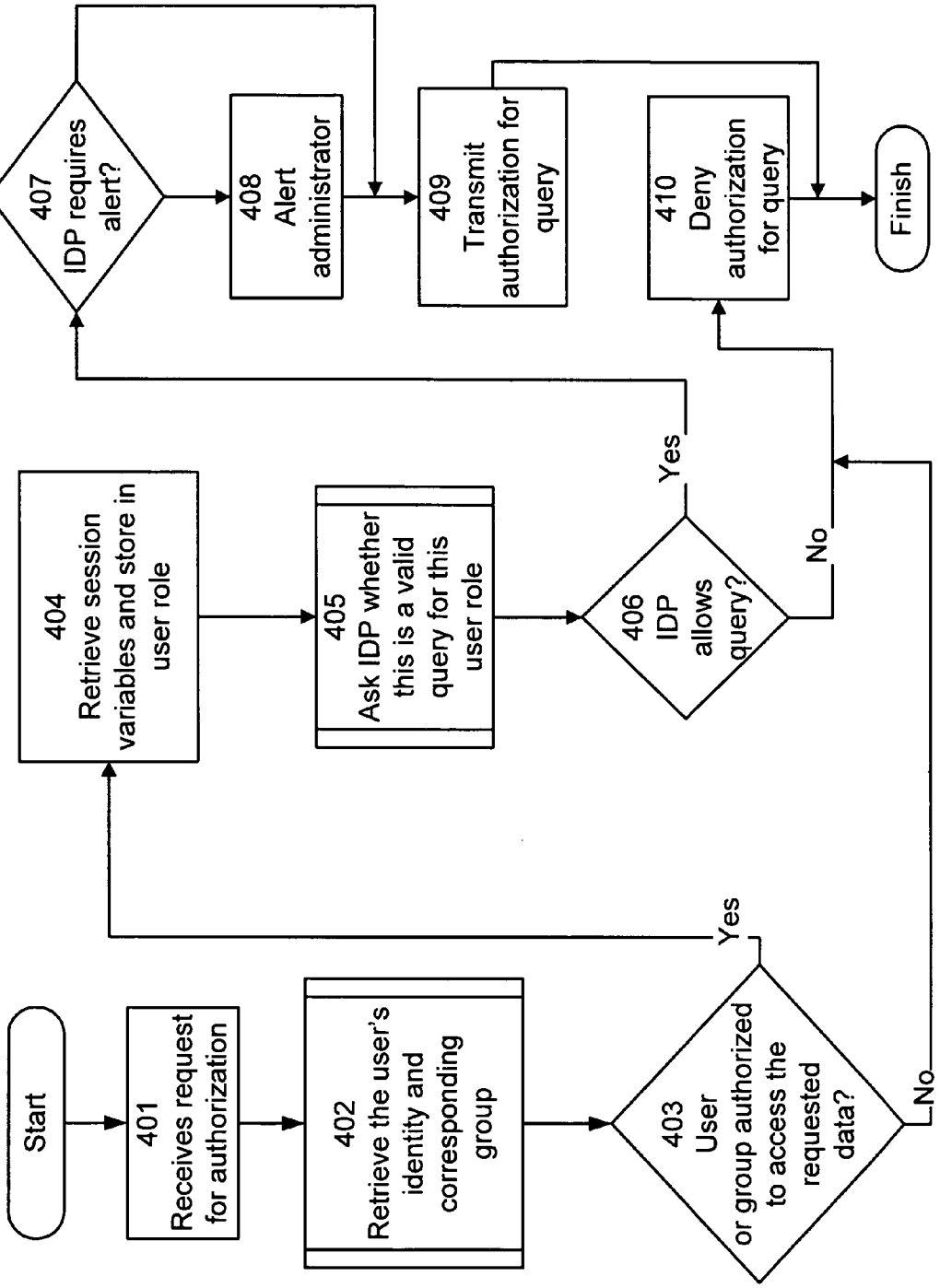

Start
401 Receives request for authorization
402 Retrieve the user's identity and corresponding group
403 User or group authorized to access the requested data?
Yes
No
404 Retrieve session variables and store in user role
405 Ask IDP whether this is a valid query for this user role
406 IDP allows query?
Yes
No
407 IDP requires alert?
408 Alert administrator
409 Transmit authorization for query
410 Deny authorization for query
Finish

FIG. 4B

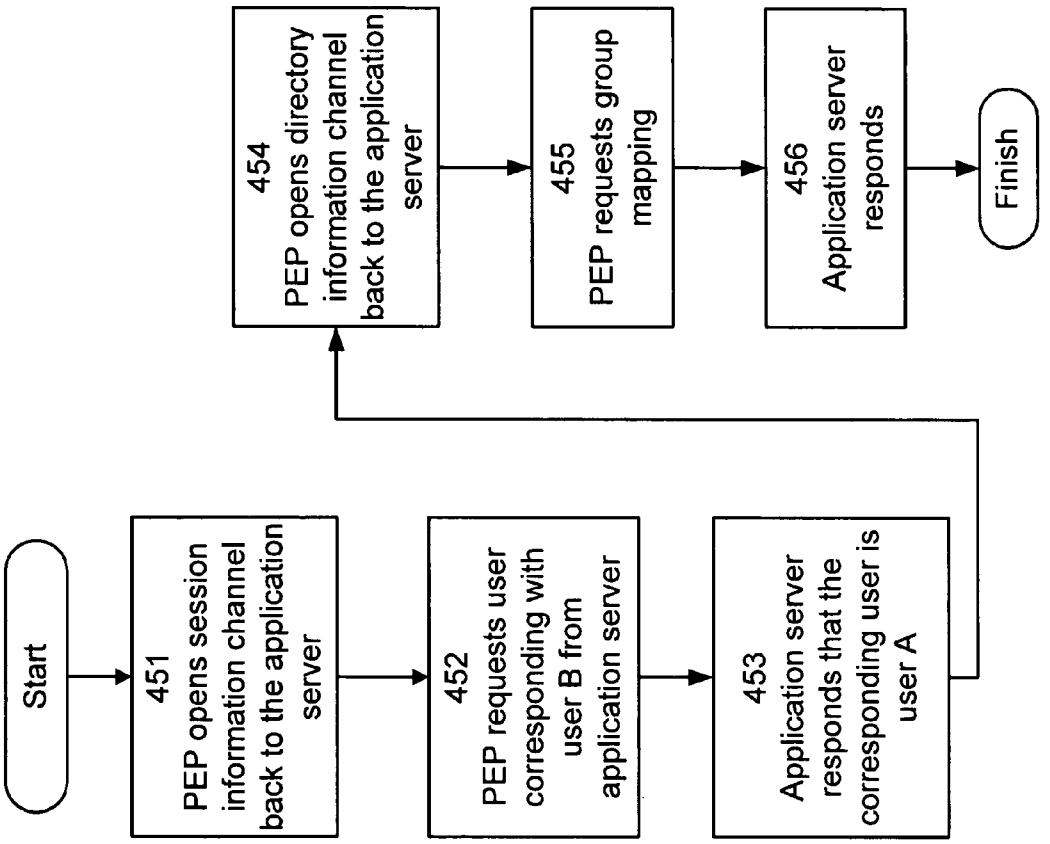

Start
451
PEP opens session information channel back to the application server
452
PEP requests user corresponding with user B from application server
453
Application server responds that the corresponding user is user A
454
PEP opens directory information channel back to the application server
455
PEP requests group mapping
456
Application server responds
Finish

FIG. 5

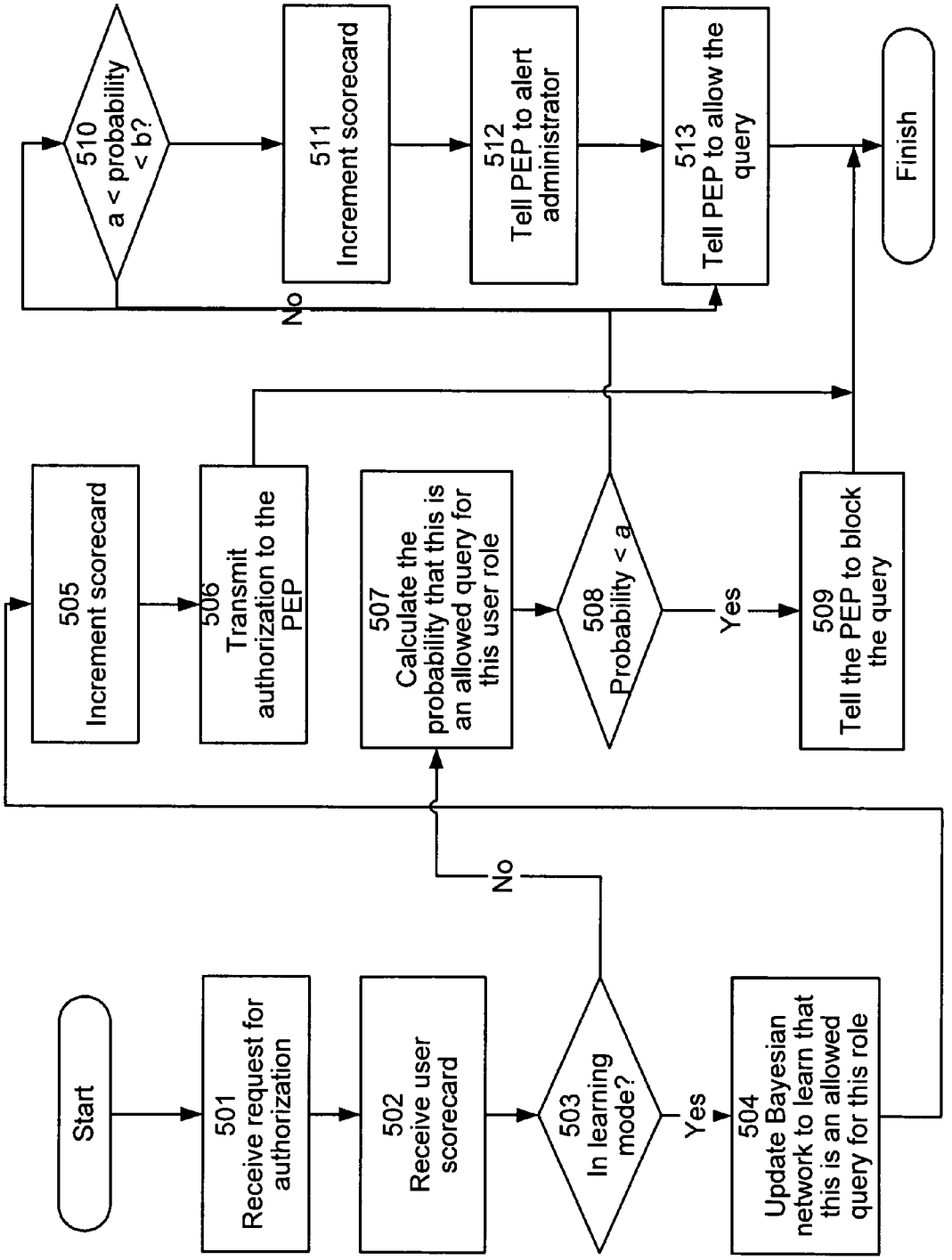
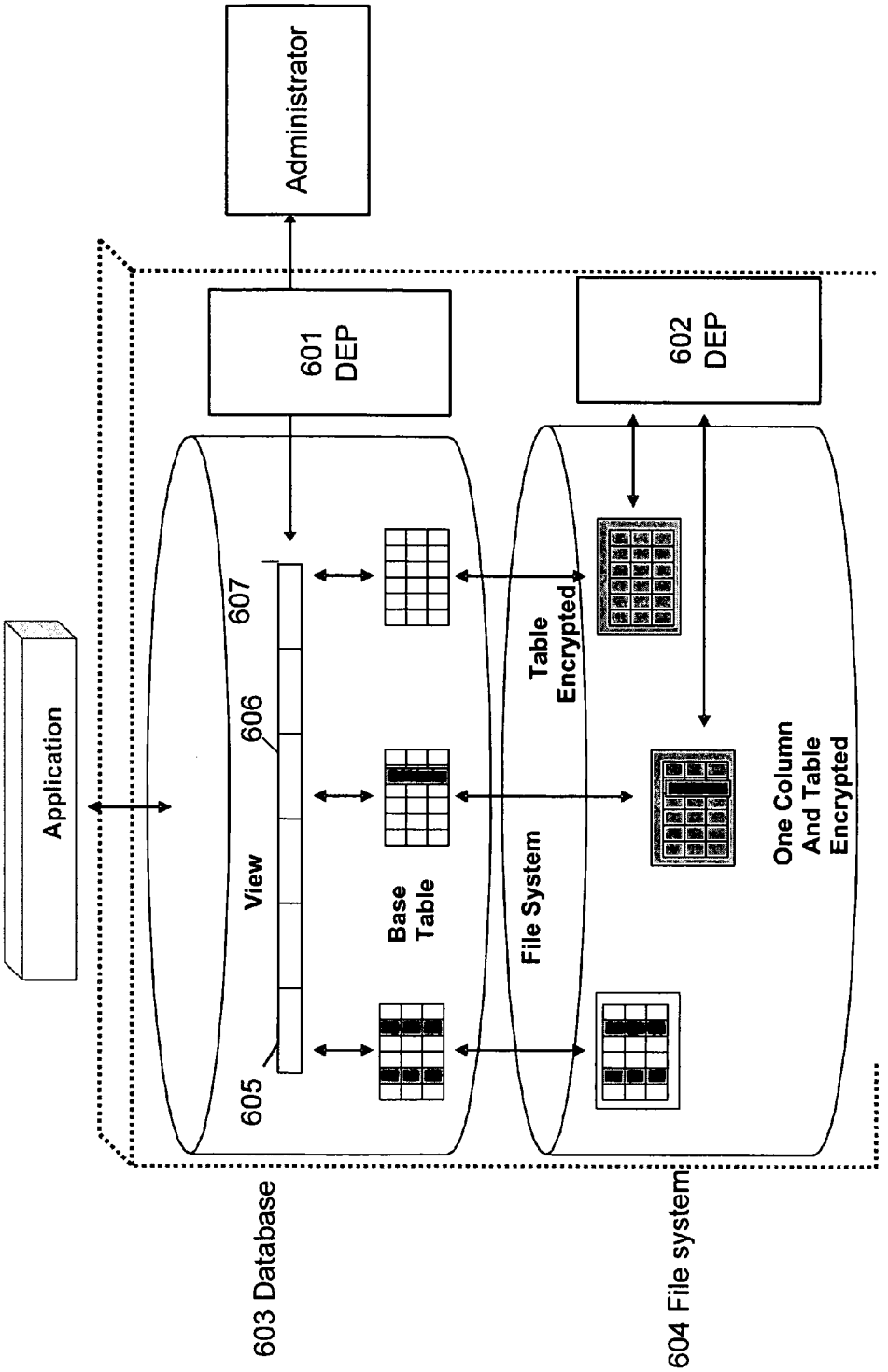

Start
501 Receive request for authorization
502 Receive user scorecard
503 In learning mode?
Yes
504 Update Bayesian network to learn that this is an allowed query for this role
No
505 Increment scorecard
506 Transmit authorization to the PEP
507 Calculate the probability that this is an allowed query for this user role
508 Probability < a
Yes
509 Tell the PEP to block the query
510 a < probability < b?
No
511 Increment scorecard
512 Tell PEP to alert administrator
513 Tell PEP to allow the query
Finish

FIG. 6


Administrator
601 DEP
602 DEP
Application
607
606
605
View
Base Table
File System
Table Encrypted
One Column And Table Encrypted
603 Database
604 File system

MULTI-LAYER SYSTEM FOR PRIVACY ENFORCEMENT AND MONITORING OF SUSPICIOUS DATA ACCESS BEHAVIOR

RELATED APPLICATION

[0001] This application claims priority from co-pending provisional U.S. Application Ser. No. 60/654,181, filed Feb. 18, 2005. This application also claims priority from copending U.S. application Ser. No. 10/034,996, filed Dec. 28, 2001.

FIELD OF DISCLOSURE

[0002] The disclosure is directed to software for interacting with a database, and in particular, to software for interacting with databases that include encrypted data.

BACKGROUND

[0003] It is difficult to detect advanced attacks on data and data misuse by monitoring only one system layer. It is likewise difficult to attain acceptable performance in using external policy driven encryption systems.

[0004] One difficulty arises because large amounts of encrypted data are exposed when providing an efficient search on encrypted data. In addition, large amounts of sensitive data are exposed when using effective performance optimization to offload cryptographic operations. This results in exposure of data in memory or disk, outside of the control of the security/encryption system.

SUMMARY

[0005] In general, in some aspects, a method for controlling data access in a data-at-rest system includes executing a link intrusion prevention analysis between multiple layers of the data-at-rest system, introducing a privacy policy at enforcement points that span multiple system layers, and dynamically altering the privacy policy.

[0006] In some implementations, the method includes one or more of the following features. The data-at-rest system is a database system. The method further includes modifying the protection of data at one of the multiple system layers. The step of modifying is performed based on a result of the link intrusion prevention analysis. The privacy policy includes access control information. The privacy policy includes intrusion detection information. The privacy policy includes cryptographic information.

[0007] In general, in some aspects, a method for controlling access to a database system includes assigning a first access criterion and a second access criterion to a user role, receiving a query from a user, the user having an access history, determining that the user matches the user role, comparing, in a first system layer, the access history to the first access criterion, and comparing, in a second system layer that differs from the first system layer, the access history to the second access criterion.

[0008] In some implementations, the method includes one or more of the following features. The first access criterion comprises a privacy policy. The method further includes learning a value for the first access criterion. The method further includes selecting a response to the query, wherein the response is selected from the group consisting of blocking the query, alerting a system administrator and allowing the query, and allowing the query. Selecting a response to the query comprises selecting a response to the query based on a result of the step of comparing in a first system layer.

[0009] In general, in some aspects, a method for accessing data includes in a first system layer, receiving a first request from a user, the user having an access history, the access history including a counter, in the first system layer, comparing the counter to a first threshold, transmitting a second request to a second system layer, the second request being based on the first request.

[0010] In some implementations, the method includes one or more of the following features. The method further includes comparing the counter to a second threshold. The counter includes a scorecard. The method further includes determining that the counter exceeds a third threshold, and alerting a system administrator. The method further includes, in the first system layer, transmitting a notification to the second system layer to deny the second request.

[0011] Other general aspects include other combinations of the aspects and features described above and other aspects and features expressed as methods, apparatus, systems, program products, and in other ways.

[0012] Advantages and features will become apparent from the following description and claims.

DESCRIPTION OF DRAWINGS

[0013] **FIGS. 1, 3** and **6** are block diagrams of database systems.

[0014] **FIGS. 2A, 2B**, 2C, **4**A, **4**B, and **5** are flow charts.

DETAILED DESCRIPTION

[0015] The system described herein is intended to be integrated into that described in copending U.S. application Ser. No. 10/034,996, filed Dec. 28, 2001, and entitled METHOD FOR INTRUSION DETECTION IN A DATABASE SYSTEM, the contents of which are herein incorporated by reference.

[0016] A method and system for overcoming the foregoing difficulties provides for the introduction of a privacy policy with enforcement points that span multiple system layers.

[0017] The privacy policy is coupled with link intrusion prevention analysis between multiple system layers. The scope, both in data and in time, for enforcing data privacy and encryption is then dynamically optimized between multiple system layers.

[0018] As used herein, multiple system layers includes application database sessions, table data access, table space access, and database file level access. The term "transaction" is intended to include queries. The term "data at rest" is intended to include all forms of stored data. A "data-at-rest system" includes any system for storing data.

[0019] In a system for overcoming the foregoing difficulties, selected rules control the amount of data that is exposed, and the time window for exposure of unencrypted data. A policy underlying the selected rules defines the extent to which data privacy is to be enforced for particular data. This extent, which includes the extent of the particular data exposed and the duration of such exposure, is determined on the basis of the sensitivity of the particular data.

[0020] Dynamic control over the extent and duration of unencrypted-data exposure required to satisfy a user transaction is provided by linking the intrusion detection point ("IDP"), the policy enforcement point ("PEP"), the audit generation point ("AGP"), and the data-at-rest encryption point ("DEP"). These scopes are controlled by an operational sensitivity class defined in the policy. The operational sensitivity class defines what rules to check and when to do so by linking the IDP, the PEP, the AGP, and the DEP.

[0021] At the intrusion detection point, a scorecard is provided to accumulate violation attempts. On the basis of the number of violation attempts, session statistics, and data access statistics spanning multiple system layers, one can determine whether a threshold indicative of an attack has been reached.

[0022] A system as described above enhances the ability to detect advanced attacks on data as well as instances of data misuse. The system also reduces the extent to which data is exposed and outside the control of the security/encryption system, both in terms of the amount of data being exposed and the duration of such exposure. In addition, the system enables effective performance optimization and offloading of cryptographic operations.

[0023] In an exemplary security system **116**, depicted in **FIG. 1**, a user **115** communicates through a client **114**, which interacts with an application server **113**. The application server **113** communicates with a PEP **101** to request authorization and to transmit auditing data. The application server **113** also passes queries along to a database **107**, which itself communicates with the PEP **101** to request authorization. The database process **107** also communicates with a DEP **103** to request decryption. The DEP **103**, in turn, utilizes a hardware security module ("HSM") **105** and a software security module ("SSM") **106**.

[0024] The database process **107** transmits requests to its buffer **110**, which (through a process overseeing the buffer) sends audit information to the PEP **101**. The buffer process then transmits requests to a file system file **108**, which also communicates with the DEP **103** to request that a file be decrypted.

[0025] The PEP **101** communicates directly with the DEP **103** to provide authorization for other components to decrypt files. The PEP **101** interacts with the AGP **104** to store audit information. The PEP **101** calls the IDP **102** to provide information used in determining whether a given query should be allowed. In response, the IDP **102** tells the PEP **101** that a given query should either be allowed, blocked, or allowed but with an alert sent to the system administrator. It performs this task with the aid of intrusion detection rules **112** and a scorecard **111** associated with each user.

[0026] The DEP **103** can be optimized to perform at a layer that allows granularity (e.g., operations on a table cell vs. a table vs. an entire database vs. an entire file system) in compliance with a privacy policy. The DEP **103** can then dynamically dispatch an operation to be performed, either by a hardware security module ("HSM") **105** or by a software encryption engine **106**, or a combination thereof. The DEP **103** can operate on an in-memory database or on a disk.

[0027] Operations on different levels of granularity may be achieved, in the depicted example, by associating the DEP **103** with multiple layers of the database hierarchy. The DEP **103** is connected to The database process **107** and the data store (file system) layer **108**. An encryption request originating at the database layer **302** permits the DEP **103** to encrypt data in an individual row, column, or cell. (It might also, however, permit a database administrator to decrypt data for which the administrator lacks authorization.) In some embodiments, an encryption request originating at the file system layer **108** permits the DEP **103** to encrypt data in an individual file system file, thereby preventing a database administrator from accessing sensitive data.

[0028] If permitted by the privacy policy, the DEP **103** can, under certain conditions, dynamically re-route a decryption request from a software security module **106** to a hardware security module **105**. Exemplary conditions include having a message size larger than a predetermined size. This dynamic re-routing optimizes performance and offloads cryptographic operations.

[0029] Upon detecting an attack, the PEP **101** can carry out any combination of the following options: issuing a security alert, blocking access to selected data, disabling one or more users, and disabling a request.

[0030] **FIGS. 1 and 2**A-2C illustrate one example of the operation of a system including a PEP **101**, an IDP **102**, a DEP **103**, and an AGP **104**. In this example, a database user **115** initiates a transaction through a client **114**, such as a web browser (step **201**). The client **114** sends a request to an application server **113**, e.g., a web server (step **202**).

[0031] The application server **113** initiates an authentication request with the PEP **101** (step **203**). The PEP **101**, in conjunction with the IDP **102**, verifies the user's authorization, as described in more detail below in connection with **FIGS. 5A and 5B** (step **204**). If the user is not authorized (step **205**), the PEP blocks the query (step **213**). If the user is authorized, then the application server **113** sends auditing information to the PEP **101** (step **206**), which the PEP **101** transmits to the AGP **104** (step **207**). Audit information includes the database user ID, the date and time, the SQL query and other action details, the originating machine name or IP address, and the database name.

[0032] The application server **113** then sends a request to a database **107** (step **208**). The database process **107** again seeks authorization from the PEP **101** (step **209**). The PEP **101** again in conjunction with the IDP **102** verifies the user's authorization as described in connection with **FIGS. 5A and 5B** (step **210**). If authorization is granted (step **211**), the PEP **101** transmits the authorization to the database process **107** as well as to the DEP **103** (step **212**). The authorization to the DEP **103** indicates that The database process **107** is permitted to access decryption keys associated with columns to which the user **115** has access. If authorization is refused, the PEP **101** blocks the query (step **213**) and The database process **107** returns an error (step **214**), which the client **114** propagates to the user **115** (step **215**).

[0033] If the PEP **101** grants authorization, a database process **107** accesses a file **108** in the file system, through the database's buffer **110** to read the relevant data (step **216**). A computer process overseeing the buffer **110** sends additional audit information to the PEP **101** (step **217**), which the PEP **101** transmits to the AGP **104** (step **218**). If the database file **108** is encrypted (step **219**), the file system requests that the

DEP **103** decrypt the file (step **220**). If the DEP **103** has been previously authorized by the PEP **101** in step **212** (step **221**), then the DEP **103** decrypts the file using a hardware security module ("HSM") **105** and/or a software decryption engine **106** (step **222**), and returns the requested contents (step **223**).

[**0034**] The database process **107** then checks to see if any of the requested information is in an encrypted column (step **224**). If so, the process overseeing the database process **107** requests that the DEP **103** decrypt the relevant columns (step **225**). The DEP performs the requested decryption using the HSM **105** and/or the software decryption engine **106** (step **226**), and returns the decrypted results (step **227**).

[**0035**] The database process **107** extracts the relevant information (step **228**) and returns it to the application server **113** (step **229**). The application server **113** returns a result to the client **114** (step **230**), which displays a result to the user **115** (step **231**).

[**0036**] Often, many application users will share a single database user. In some examples, a PEP **101** connected to the database server utilizes the identity of the application user (in addition to or instead of the database user) as a factor in determining whether a given request is authorized. The PEP **101** does this by communicating with an application user mapping table located within the application server's security system **312**. The table contains a mapping associating the application user with the database user. Real time mapping data provides information about which application user is using the database connection at any given time. In some examples, the mapping table is stored in a database table. In other examples, the mapping table is stored in a file. In still other examples, the table, or simply the identity of the application user, is transmitted by the application server to the database server during the session.

[**0037**] As illustrated in **FIG. 3**, in some examples, security components are connected to three levels: the web or application server **301**; the database **302**; and the data store or file system **303**. Services on these three levels communicate across multiple channels: the data request channel **304**; the session information channel **305**; and the directory information channel **306**. In the depicted example, user A **309** is logged in to an application. The application requests data, as user B **310**, over the data request channel **304**, from the database. The database is running as user C **311** on a server, and requests data, over another data request channel, from the data store (e.g., the file system).

[**0038**] Meanwhile, on the application server **301**, a mapping table, which associates user A **309** (the application user) with user B **310** (the database user), is maintained. This information may be communicated, via a separate session information channel **305**, to the database's security system **307**.

[**0039**] Examples of further details of the operation of the PEP **101**, the IDP **102**, and the DEP **103** are provided below. The boundaries of the functions performed by the IDP **102**, the PEP **101**, the AGP **104**, and the DEP **103** are not fixed; some functions may be combined in a single component, or allocated differently between components.

[**0040**] A. PEP

[**0041**] **FIG. 4A** explains in further detail an example of operation of the PEP **101** (see **FIG. 2**, steps **204** and **210**). First, the PEP **101** receives a request for authorization (step **401**). The PEP **101** then retrieves the user's identity and corresponding group, as described below in connection with **FIG. 4B** (step **402**). The PEP **101** then determines whether the user or group is authorized to access the requested data, for example, by consulting a privacy policy or access control list (step **403**). If the user or group is unauthorized to access the requested data, the PEP **101** skips to step **410**.

[**0042**] If the user or group is authorized, the PEP **101** then retrieves session variables (including the time of day, day of week, IP address from which the user is logged in, the user's geographic location, the user's identity, the user's group, the user's client's software, etc.), and stores these variables as the user's "role" (step **404**). The PEP **101** then communicates with the IDP **102** to determine whether the query is valid for the user's role (step **405**). The IDP makes this determination in a process exemplified by **FIG. 5**. The PEP **101** determines whether the IDP **102** allowed the query (step **406**). If the IDP **102** rejects the query, the PEP **101** skips to step **410**.

[**0043**] If the query was allowed, the PEP **101** checks to see whether the IDP **102** indicated that an alert was to be sent to the system administrator (step **407**). If so, the PEP alerts the administrator (step **408**). In either event, the PEP **101** authorizes the query (step **409**). If the query was not allowed, the PEP **101** denies authorization for the query (step **410**).

[**0044**] **FIG. 4B** describes how, in some examples, the database's security system **307** retrieves the user's identity and corresponding group in step **402**. First, the PEP **101** in the database's security system **307** opens a session information channel **305** to the application server **301** (step **451**). Next, the PEP **101** requests, from a mapping table in the application's security system **312**, the application user corresponding to the current database user (for example, user B **310**) (step **452**). The application server responds, for example, that the corresponding user is user A **309** (step **453**).

[**0045**] Next, the PEP **101** opens a separate directory information channel **306** back to the application server **301** (step **454**). On the directory information channel **306**, the PEP **101** requests the group mapping for user A **309** (step **455**). In a typical response, the application server **301** indicates that user A **309** is a member of group X (step **456**).

[**0046**] **FIG. 4B** describes the process by which the PEP **101** in the database's security system **307** retrieves information from the application server **301**. In some examples, the PEP **101** in the data store's security system **308** requests the identity of the database user (i.e., user B **310**) from the PEP **101** in the database security system **307** over the session information channel **305**. The PEP **101** in the data store's security system **308** may also ascertain the database user's group over the directory information channel **306**.

[**0047**] In some examples, the PEP **101** in the data store's security system **308** can identify the application user by requesting the information from the PEP **101** in the database's security system **307**, which then relays the query to the application server's mapping table **311**. Similarly, in some examples, the PEP **101** in the data store's security system **308** can ascertain the application user's group, by requesting the information from the PEP **101** in the data-

base's security system **307**, which then relays the query to the mapping table **311** in the application server.

[**0048**] In some practices, a database's security system **307** (for example, through its PEP **101**) notifies other layers to indicate that a severe attack has occurred. In some practices, the IDP **102** in the application server's security system **312** receives this notification and subsequently blocks all access attempts that would otherwise have only triggered an alert to the system administrator. In some practices, the DEP **103** in the data store's security system **308** receives this notification and blocks all subsequent requests to decrypt data.

[**0049**] One example of these practices is provided where a PEP **101** detects that authorized access to credit card information at the database level exceeds normal usage, but not is not at a critical level. The PEP **101**, in this example, modifies a privacy policy to instruct the application server's security system **312** to block further access attempts. In another example, the PEP **101** in the application server's security system **312** detects multiple hacking attempts from multiple locations. The security system **312** modifies a privacy policy to block requests at the application server **312** level, increase file security at the data store's security system **308**, and prevent the data store's security system **308** as well as the database's security system **307** from decrypting sensitive data.

[**0050**] B. IDP

[**0051**] In some embodiments, the IDP **102** has a learning mode and an enforcement mode. In learning mode, the IDP **102** acquires information about users of the system, including the typical time of day and day of week during which they access the system, the resources they usually access, their physical location or IP address, and the volume of data they usually access. In some examples, the IDP **102** maintains a Bayesian network to associate authorized accesses with these variables. In other examples, other types of learning may be used. When the IDP **102** is in enforcement mode, it denies access to a user when the time or day of access, the resources accessed, the user's location or IP address, or the volume of data requested exceeds a learned threshold or differs from learned values. The IDP **102** optionally alerts a system administrator when any of these criteria exceeds a learned threshold or differs from learned values.

[**0052**] In some embodiments, the IDP **102** accepts user logins only during certain times of day, or only on certain days of the week, or only from certain physical locations. In some examples, the IDP **102** learns how these criteria should be restricted. In other examples, the system administrator manually enters restrictions. In some examples, the system administrator manually changes restrictions, for example, to temporarily allow a particular user to log in from a distant location when the user is on vacation.

[**0053**] In some embodiments, the IDP **102** restricts the volume of data a user may access in a given day. In one example, the IDP **102** permits a user to access only a predetermined number of rows per day from a given table. In another example, the IDP **102** permits a user to issue only a predetermined number of queries per day in a given table. In other examples, the user is restricted to a given volume of data over the entire database, rather than in specific tables. In some examples, the IDP **102** uses a counter to maintain information about the volume of data a user has accessed by means of a counter.

[**0054**] In some examples, an IDP **102** restricts access based on the user's role. A user's role may be based on his or her identity, the time of day, the day of week, the IP address being used, the country or geographic region from which the request originates, etc. In some examples, an IDP **102** located in the database server sets a maximum number of rows per day accessible to users in a given role. Some examples restrict the number of rows a user in a given role may insert, or the number of rows a user in a given role may modify, or the number of rows a user in a given role may delete. In some examples, these values are learned while the IDP **102** is in learning mode.

[**0055**] Some examples permit the IDP **102** and/or the PEP **101** to communicate with a trusted component running on an authorized client to further assist in user authentication.

[**0056**] In some examples, the IDP **102** utilizes one or more of the following criteria to decide whether to permit access, block access, or alert the administrator: session authorization (i.e., the user's identity); session authentication (i.e., the resources a user is entitled to access); session encryption; password integrity; database software integrity; application data integrity; database metadata integrity; security software integrity; time of day of access; and signature rules (i.e., pattern matching and content analysis to detect any known attack signature using, e.g., Snort® network intrusion detection software). To verify data or software integrity, a hash value is stored and verified against periodically.

[**0057**] In some examples, the IDP **102** triggers an alert whenever a particular user accesses an abnormally high volume of data. When this alert is triggered, the PEP **101** analyzes an audit log to ascertain whether unusual activity is occurring. If so, the IDP **102** can disallow further accesses by the current user, and/or send an alert to a system administrator.

[**0058**] An example of the operation of the IDP **102** will now be described with reference to **FIG. 5**. First, the IDP **102** receives a request for authorization from the PEP **101** (step **501**). The request includes information about the user's role (see **FIG. 4A**, step **404**). The request also includes a query the user seeks to execute. The IDP **102** next retrieves the user's scorecard **111** (including variables tracking a user's total volume of access in a given time period, e.g., total number of rows accessed in a day, total kilobytes of data downloaded in a day, etc.) (step **502**).

[**0059**] Next, the IDP **102** checks to see if it is in learning mode (step **503**). If it is, then the IDP **102** updates a Bayesian network to learn that the query is authorized for this user role (step **504**). Next, the IDP **102** adds data from the current query to the user's scorecard **111** (step **505**). Finally, the IDP **102** transmits authorization to the PEP **101** (step **506**).

[**0060**] If the IDP is not in learning mode, then the IDP **102** calculates the probability that the query is allowed for the user's role (step **507**). If this probability is below a predetermined threshold a (step **508**), then the IDP **102** tells the PEP **101** to block the query (step **509**). If, instead, the probability is between the threshold a and a predetermined threshold b, where b>a (step **510**), then the IDP **102** adds the data from the current query to the scorecard **111** (step **511**), allows the query, and tells the PEP **101** to send an alert to a system administrator (step **512**). If the probability is greater than the threshold b, the IDP **102** simply allows the query (step **513**).

[0061] More generally, in some examples, the IDP **102** maintains a set of i thresholds t_i. For each interval $[t_i, t_{i+1})$, a different action k_i is defined. If the probability that the query is allowed is within the interval $[t_i, t_{i+1})$, then the action k_i is performed.

[0062] In some examples, the IDP **102** increments the value on the scorecard **111** in response to accesses, or attempted accesses, to sensitive resources. In some examples, sensitive resources include access to prespecified applications and prespecified network addresses. In other examples, the IDP **102** increments the value of the scorecard by a greater amount in response to a failed or disallowed attempt to access the sensitive resource.

[0063] C. DEP

[0064] **FIG. 7** depicts an example of interaction between a DEP **601** connected to a database **603** and a DEP **602** connected to a file system **604**. In one example view **605**, two columns are encrypted, but the table itself is not. In this example, authorization is handled entirely by the database's DEP **601**. In a second example view **606**, one column is encrypted, and the table is encrypted as well. In this example, the database's DEP **601** provides the decryption key for the column, while the file system's DEP **602** provides the decryption key for the table. In this example, a database administrator would be precluded from accessing secure data due to the table-level encryption. In the third example view **607**, no columns are individually encrypted; but the table is encrypted. In this example, the file system's DEP **602** provides the decryption key.

[0065] In some examples, when a PEP **101** determines that a given query is to be blocked, it performs one or more of the following tasks: disconnecting the user; denying access to cryptographic keys; writing a record in a log file; and sending an error return code, coupled with no data, back to the requesting application.

[0066] In some examples, the PEP **101** includes a machine and program authorization (MPA) component. This component prevent or restrict users with valid login names and passwords from connecting to the database unless they access the database from a machine that has been preauthorized. Machines are authorized if they have an authorized IP address, and additionally, if they are able to specify both the port on which the database server is listening and the name of the database.

[0067] In some examples, the PEP **101**, IDP **102**, DEP **103**, and AGP **104** are part of the Protegrity™ Secure Data server, which is available from Protegrity Corporation of Stamford, Conn.

[0068] It is to be understood that while the invention has been described in conjunction with the detailed description thereof, the foregoing description is intended to illustrate and not limit the scope of the invention, which is defined by the scope of the appended claims.

Having described the invention, and a preferred embodiment thereof, what is claimed as new, and secured by Letters Patent is:

1. A method for controlling data access in a data-at-rest system, the method comprising:

executing a link intrusion prevention analysis between multiple layers of the data-at-rest system;

introducing a privacy policy at enforcement points that span multiple system layers; and

dynamically altering the privacy policy.

2. The method of claim 1, wherein the data-at-rest system is a database system.

3. The method of claim 1, further comprising modifying the protection of data at one of the multiple system layers.

4. The method of claim 3, wherein the step of modifying is performed based on a result of the link intrusion prevention analysis.

5. The method of claim 1, wherein the privacy policy comprises access control information.

6. The method of claim 1, wherein the privacy policy comprises intrusion detection information.

7. The method of claim 1, wherein the privacy policy comprises cryptographic information.

8. A computer-readable medium containing instructions for causing a computer to perform the method of claim 1.

9. A method for controlling access to a database system, the method comprising:

assigning a first access criterion and a second access criterion to a user role;

receiving a query from a user, the user having an access history;

determining that the user matches the user role;

comparing, in a first system layer, the access history to the first access criterion; and

comparing, in a second system layer that differs from the first system layer, the access history to the second access criterion.

10. The method of claim 9, wherein the first access criterion comprises a privacy policy.

11. The method of claim 9, further comprising learning a value for the first access criterion.

12. The method of claim 9, further comprising selecting a response to the query,

wherein the response is selected from the group consisting of blocking the query, alerting a system administrator and allowing the query, and allowing the query.

13. The method of claim 12, wherein selecting a response to the query comprises selecting a response to the query based on a result of the step of comparing in a first system layer.

14. A computer-readable medium containing instructions for causing a computer to perform the method of claim 9.

15. A method for accessing data, the method comprising:

in a first system layer, receiving a first request from a user, the user having an access history, the access history including a counter;

in the first system layer, comparing the counter to a first threshold; and

transmitting a second request to a second system layer, the second request being based on the first request.

16. The method of claim 14, further comprising comparing the counter to a second threshold.

17. The method of claim 14, wherein the counter comprises a scorecard.

18. The method of claim 14, further comprising:

determining that the counter exceeds a third threshold; and

alerting a system administrator.

19. The method of claim 14, further comprising:

in the first system layer, transmitting a notification to the second system layer to deny the second request.

20. A system comprising:

a computer-readable medium as recited in claim 14; and

a computer in data communication with the computer-readable medium.

* * * * *