Computing and Informatics, Vol. 32, 2013, 1192–1211

DATABASE SECURITY SYSTEM FOR APPLYING SOPHISTICATED ACCESS CONTROL VIA DATABASE FIREWALL SERVER

Eun-Ae Cho

Visual Display Division
Samsung Electronics Co., Ltd, Suwon, Republic of Korea
e-mail: ea.cho@samsung.com

Chang-Joo Moon

Department of Aerospace Information System Engineering
Konkuk University, Seoul, Republic of Korea
e-mail: cjmoon@konkuk.ac.kr

Dae-Ha Park

Department of Information Management and Security
The Cyber University of Korea, Seoul, Republic of Korea
e-mail: summer69@cyberkorea.ac.kr

Kang-Bin Yim*

Department of Information Security Engineering
Soonchunhyang University, Asan, Republic of Korea
e-mail: yim@sch.ac.kr

Abstract. Recently, information leakage incidents have occurred due to database security vulnerabilities. The administrators in the traditional database access control methods grant simple permissions to users for accessing database objects. Even

* corresponding author

though they tried to apply more strict permissions in recent database systems, it was difficult to properly adopt sophisticated access control policies to commercial databases due to performance degradations. This paper proposes a database security system including a database firewall server as an enhanced database access control system which can efficiently enforce sophisticated security policies to provide database with confidentiality using a data masking technique for diverse conditions such as the date, time, SQL string, and table columns to database systems.

Keywords: Database security, privacy, access control, database firewall, data masking

1 INTRODUCTION

Recently, diverse sensitive information, including personal private information, has been stored with an increasing frequency in databases [1, 4]. Organizations manage numerous databases, and their users request information through diverse access methods to these data.

In databases, diverse access paths to information can enhance usability of the information, but there is a strong possibility of malicious collection or illegal usage of the data because probability of information exposure can be increased proportionately. Thus, safe management is necessary for sensitive information in databases.

Several studies are currently being conducted to control accesses and manage data for database security [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. The methods of database security can be divided into two parts [2]: access control/audit methods and data encryption methods. Access control/audit methods look out for input/output route of the database, and data encryption methods deal with encrypted data in databases.

However, those studies have several problems. In the existing studies which use access control methods, detailed and various access requirements cannot be accommodated, and it is not easy to change these requirements when a user's security requirements change many times. While current security solutions have several strong points, they also have weak points such as insufficient confidentiality for data itself and decreased performance for database security. For example, Oracle 8i which holds a majority share of the DBMS (Database Management System) market includes its data encryption module as a default function. However, usage of the encryption module is very restrictive because of efficiency deterioration. Thus, development of information protection technologies for preventing illegal obtaining of information from inside or outside is urgently required [2]. Owing to the situations mentioned above, information leakage incidents of databases still occur. In order to solve this problem, the proposed database security system provides confidentiality and efficiency in management of integrated security policies, which are needed to protect data and to satisfy feasibility requirements.

The rest of this paper is organized as follows. Section 2 discusses the related works for existing database security methods and their problems, and Section 3 defines several symbols and meanings related to access control policy in detail. Section 4 describes the requirements of information user and proposes a new database security system including database firewall server. Section 5 shows implementation results, and Section 6 compares the existing models and the proposed model. Finally, Section 7 concludes the paper and presents future works.

2 RELATED WORKS

On the market, there are several database security solutions, such as Chakra^{TM} [5], Middleman^{TM} [6], DBSafer^{TM} [7], DB-i^{TM} [8] and dGriffin^{TM} [9], which use access control and audit methods, and Cube One^{TM} [10], D'AMO^{TM} [11], SafeDB^{TM} [12], and Secure.Data^{TM} [13], which use data encryption methods. The access control and audit methods have some merits in that they are able to guarantee the efficiency of DBMS and are convenient to operate, but data can be disclosed easily when it is robbed because it is not secured. While data encryption method [14] has a merit that it is safe, due to encryption, even if data are disclosed, it lowers its efficiency because it is installed first in DBMS directly and operated.

In addition, a policy-based research [15], which uses both access control models [16] with MAC (Mandatory Access Control), DAC (Discretionary Access Control), RBAC (Role Based Access Control) [3] etc., and data encryption methods [14, 17], is being carried out for controlling access and managing data for database security. However, this research has the following merits and demerits. First, the existing research, which uses access control model, manages the user's access, but cannot accept detailed and diverse access requests like access control for specific SQL statements or specific database columns. For example, it is possible to make and apply the policy that two random users can access the same attribute, but it is hard to give permission to each tuple of the attribute in the existing models [18]. Also, modifications are not easy when the user's security requirements change frequently. That is, it cannot be dealt effectively that users' access permissions of random data change many times. Moreover, the access control by each user cannot be provided for a piece of data. For instance, a situation can occur in which a user who has access permission to specific data can give access permission to part of that data to a user who does not have permission to it.

Among the above-mentioned existing studies for database security systems, a recent study suggests a security system for the various sizes of data group [18]. This system uses various access control policies, such as MAC, S-DAC (selective DAC), and RBAC, for protecting data. In 'Phase1', first, it examines the user IDs of those who want access and checks whether the access requirement is a part of the SQL statement or one of the roles. The system then starts to work according to each situation. According to the security level or role, access is decided by using the MAC and RBAC policies, and then in 'Phase2', the former phase's result can be filtered selectively again for the various sizes of data groups by using S-DAC policy. Finally, only the data items which pass all the security policies can be shown to the user.

3 PRELIMINARY NOTIONS

In our work, we define several symbols and meanings related to access control policy. First, user Ui is a subject of information accesses and means a user of database. The data is an object which means set of information and can be defined as data or a group of data. Security level called SL includes USL (user security level) and DSL (data security level), and can be defined according to the policy in each organization and/or system. Detailed expressions and definitions can be described as follows.

Symbols	Description
U	User set, $U = \{U_1, U_2, \ldots, U_i\}$
i	User identity
R	User role set, $R = \{R_1, R_2, \ldots, R_j\}$
j	Role identity
SL	Security Level set, $SL = \{SL_1, SL_2, \ldots, SL_s\}$
	* User security level $USL = \{USL_1, USL_2, \ldots, USL_s\}$
	* Data security level $DSL = \{DSL_1, DSL_2, \ldots, DSL_s\}$
s	Security level identity
C	Column set, $C = \{C_1, C_2, \ldots, C_t\}$
t	Column identity
N	Total number of users

Table 1. Definitions of access control policy symbols

For efficient policy management, prerequisites expression can be defined by using the above symbols in Table 1 as follows;

- $R(U_i) \subset R_j$
- $SL(R_i) \subset USL_s$, $SL(C_t) \subset DSL_s$
- $SL_1 \geq SL_2 \geq \ldots \geq SL_s$
- $1 \leq N(USL_s) \leq i$, $1 \leq N(R_j) \leq i$.

Each user is one of the members for a role, and each role has a USL and data column has a DSL. In addition, more than one kind of USL or role should exist. However, only DSL can include the 'Ø' (empty set). For each security level 's', lower number has priority over higher number.

Next, definitions related to the policies for particular condition can be described as shown in Table 2. The items in Table 2 can be used in defining the policies to control detailed access. There are several examples to generate policies for detailed access control as follows:

$$Condition \rightarrow subject, object, permission\ (or\ operation).$$

Symbols	Description
IP	IP address
Host	Host name
DbUser	Database user
DbTable	Database table
DbInst	Database instance
DbSess	Database session
DbSql	Database SQL
Time	Time
Date	Date
App	Application
LoginUser	Login user
Cmd	Command

Table 2. Definitions of detailed access control policy symbols

Security level policy table is divided into two parts for efficient retrieval and process of core data; USL_Table (user security level table) for managing user and DSL_Table (data security level table) for managing data. The policies for these tables can be illustrated as follows.

Policy 1. Basic Security Level Policy

$$[USL_k \geq DSL_i \cap R_i \in USL_k \cap C_t \in DSL_i] \rightarrow R_i, C_t, R.$$

When USL_k dominates DSL_i, the role 'R_i' which is a member or USL_k can 'R(Read)' a data 'C_t' which is a member of DSL_i. The permission can be listed not only as R(Read), but also as I(Insert), U(Update), D(delete). They can be represented as R_i, C_t, R, I, U, D.

Policy 2. Specific Policy related to particular user or data column for detailed access control:

- The expression of detailed access control:

$$\textit{Defined item} = \textit{value}.$$

- The expression of detailed data item for masking:

$$\begin{aligned} \textit{Columns} &= (\textit{table}_1 : \textit{column}_1, \textit{column}_2, \ldots, \textit{column}_m), \\ &\quad (\textit{table}_2 : \textit{column}_1, \textit{column}_2, \ldots, \textit{column}_m), \\ &\quad \ldots \end{aligned}$$

The detailed access control includes the diverse policies which are related to detailed items such as IP, Host, DbUser, DbTable, Time, Date, App, LoginUser, Cmd, DbInst, DbSess, DbSql as defined in Table 2. Masking columns can be divided into sections by table and they can have policies for each.

Example 1. We can suppose a scenario to apply the policies given above. A department has ten users for databases and five roles defined. In this case, user can be a member of each role as follows;

- $U = \{Alice, Julie, SCOTT, Calvin, Cho, Park, Bob, Lee, Moon, Yim\}$, $i = 10$
- $R = \{admin, manager, employee, customer, guest\}$, $j = 5$
- $\{Kim, Lee\} \subset admin$,
  $\{Alice, Moon, SCOTT\} \subset manager$,
  $\{Kim, Lee, Alice, Moon, SCOTT\} \subset employee$,
  $\{Yim, Julie\} \subset guest$,
  $\{\} \subset customer$.

The security levels which are related to the above-mentioned data and users are defined as the following expressions and also in Figure 1.

- Security Level $SL = \{SL_1, SL_2, SL_3, SL_4\}$, $s = 4$
- $C = \{SSN, Address, Cell\}$, $t = 3$

<USL Table>

Security Level	User Role
SL_1	Admin
SL_2	Manager
	Director
SL_3	Employee
SL_4	Public

<DSL Table>

Data Group Name	Security Level
SSN	SL_1
Address	SL_2
Cell	SL_1
Others(Name, Login_Id, Email)	SL_3

Figure 1. Examples of a USL and a DSL table

According to the mentioned security Policy 1, if a USL is higher than a DSL, that is, a USL dominates a DSL, the user can be allowed for the data value. For example, let us suppose that there is a table which has a schema in Table 3.

Id	Name	SSN	Login_Id	Address Cell	Email
1	Alice	...	...	...	...
2	Bob	...	...	...	...

Table 3. An example of table schema

When the user 'SCOTT' wants to get a result for Name, SSN and Email from this table through some query, the data value of 'SSN' would be shown with masked result to the user in this proposed system even if the result has these three columns because the role of 'SCOTT' is 'manager' and it could not satisfy Policy 1.

$$\begin{aligned} R(SCOTT) &= manager, \\ USL(manager) &= SL_2, \end{aligned}$$

$$\begin{aligned} DSL(SSN) &= SL_1 \\ \therefore USL(manager) &< DSL(SSN) \end{aligned}$$

On the other hand, MAC (Message Authentication Code) can be used between application server and proposed system when user has a process to access a database. If integrity check failed, access to the database can be blocked.

As we mentioned before, the defined policies are used separately from each database policy. Thus, pre and post process time can be considered; however, masking method can reduce time more than data encryption method can.

Example 2. We can consider another scenario for Policy 2. With respect to the table schema in Table 3, the requirements for detailed access control except predefined USL or DLS can be supposed as follows:

> "SCOTT can access the CUSTOMER table in database from 9:00 am to 6:00 pm through a PC which is using an IP address '192.168.XXX.XXX' and named 'HOSTNAME'. In addition, columns 'SSN', 'ADDRESS' and 'MOBILE_NO' should be masked in the query result for the user."

If these detailed access controls are wanted, the policies can be defined as follows:

- Control Expression:

$$\begin{aligned} \text{IP} &= \text{192.168.XXX.XXX and} \\ \text{Host} &= \mathit{HOSTNAME} \text{ and} \\ \text{DbUser} &= \mathit{SCOTT} \text{ and} \\ \text{DbTable} &= \mathit{CUSTOMER} \text{ and} \\ \text{Time} &= \text{09:00} \sim \text{18:00} \end{aligned}$$

- Masking Columns:

$$Columns = (CUSTOMER : SSN, ADDRESS, MOBILE\_NO)$$

When detailed policies are defined as above, if the user 'SCOTT' satisfies the security level policy and detailed access control policies such as IP, Host, DBUser, DBTable and Time, 'SCOTT' can see some results which include masked data such as SSN, ADDRESS and MOBILE_NO.

4 DATABASE SECURITY SYSTEM

This section proposes our access control system to show the secure method when accesses to databases are requested from a client.

4.1 Security Requirements

As we mentioned before, the existing database security system for various sizes of data groups could not cover the aforementioned problems and requirements. Thus, to solve these problems, the proposed system should satisfy the following requirements:

1. confidentiality of data itself
2. effective query and answer for access control
3. integrated management of heterogeneous database policy
4. detailed access control according to the diverse access requirements
5. data processing which is able to do the indexing [19].

Among these, requirement (1) can resolve data confidentiality problem of the existing database security system, and requirement (2) can resolve system overhead problem. Moreover, requirement (3) can cover time delay, efficiency, and integrated management problems, and (4) and (5) can satisfy the requirements of the information user. Additionally, this paper applies the proposed framework to a method of data masking in order to protect data in (1) and makes the data indexing possible in (5) simultaneously. Moreover, when the data masking method is used, data acquisition time accessed by user to database can be managed effectively; therefore, (2) can be satisfied. Here, this paper satisfies (4) through packet monitoring, and all of these processes are designed and implemented by monitoring all the access to the database to effectively integrate and manage the aforementioned 'databases using different policies' in requirement (3).

4.2 Proposed Database Firewall Server

This paper considers the above '4.1 Security Requirements' and proposes the access control model, which applies the policy by auditing and analyzing all requested packets between the client and database. This proposed system controls the access for various kinds of databases.

The proposed security system is composed of two parts, CAA (Client Authentication Agent) and DFS (Database Firewall Server), as shown in Figure 2.

As shown in Figure 2, the whole architecture is composed of CAA, DFS, ACPR (Access Control Policy Repository) and database parts. The DFS is physically placed in front of a database or database farm and controls access to prevent bypassing.

The CAA in the client removes the Ethernet frame header, TCP header, and IP header and extracts messages from the packet which passes a client, an application server, and a database [20]. Here, the message authentication code is generated and stored from an extracted message in order to verify the integrity, and then the packet is sent to the database. The DFS catches the packet going to the database by watching the network and then obtains and analyzes the TCP data part [21, 22].

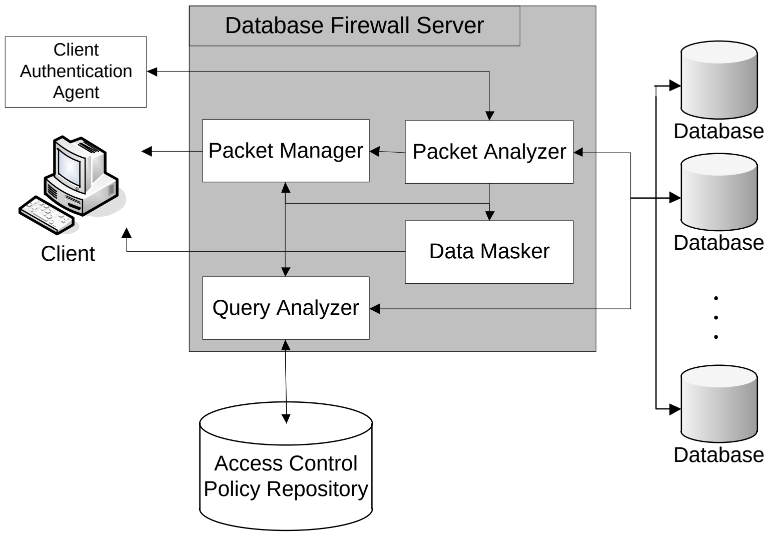


Figure 2. Whole structure of a database security system

Through the 'Packet Analyzer', the DFS extracts the packet's aforementioned headers, such as the Ethernet frame header, TCP header, IP address, and Mac address from the IP header of the packet, and then checks whether it passed user authentication.

The policy of access control system is used separately from database permissions. In addition, the system analyzes data request packets from client to database and the result packets from database to client and applies the access control policy before going to database or user.

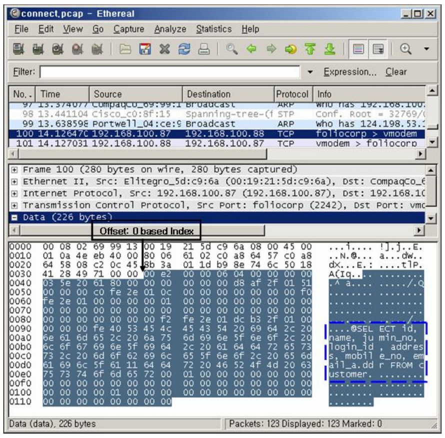
The server can communicate with client program through two kinds of method; access for external user via network protocol such as TCP/IP and access for internal user via BEQ (Bequeath protocol). However, in this paper, we focus on the TCP/IP for external user. The monitored packet contents are analyzed by using Ethereal [23] when queries are requested. The analyzed result is based on 100 sample data and, in case of reply packets, on 'SELECT' statements which have column information as SQL query results.

Figure 3 a) shows a monitoring window where user requests SQL query and receives result packets by using the 'Ethereal' tool while the process is being executed.

As shown in Figure 3 a), the user requests the following SQL query.

```
SELECT id, name, jumin_no, login_id, address, mobile_no, email_addr
FROM Customer
```

Figure 3 b) shows the result packet for requesting of Figure 3 a) to client. The packet which has a result from server includes all the data because the policies are not applied.



```
No. . Time      Source            Destination        Protocol Info
 97 13.374077 CompaqCo_69:99:1 Broadcast          ARP      who has 192.168.100.
 98 13.441104 Cisco_c0:8f:15   Spanning-tree-(f   STP      Conf. Root = 32769/(
 99 13.638598 Portwell_04:ce:9 Broadcast          ARP      who has 124.198.53.1
100 14.12647C 192.168.100.87   192.168.100.88     TCP      foliocorp > vmodem
101 14.127031 192.168.100.88   192.168.100.87     TCP      vmodem > foliocorp
```

```
⊞ Frame 100 (280 bytes on wire, 280 bytes captured)
⊞ Ethernet II, Src: Elitegro_5d:c9:6a (00:19:21:5d:c9:6a), Dst: CompaqCo_6
⊞ Internet Protocol, Src: 192.168.100.87 (192.168.100.87), Dst: 192.168.10
⊞ Transmission Control Protocol, Src Port: foliocorp (2242), Dst Port: vmc
⊟ Data (226 bytes)
```

```
0000  00 08 02 69 99 13 00 19  21 5d c9 6a 08 00 45 00   ...i.... !].j..E.
0010  01 0a 4e eb 40 00 80 06  61 02 c0 a8 64 57 c0 a8   ..N.@... a...dW..
0020  64 58 08 c2 0c 45 8b 3a  01 1d b9 8e 74 6c 50 18   dX...E.: ....tlP.
0030  41 28 49 71 00 00 00 e2  00 00 06 04 00 00 00 00   A(Iq.... ........
0040  03 5e 20 61 80 00 00 00  00 00 00 d8 af 2f 01 51   .^ a.... ...../.Q
0050  00 00 00 c0 fe 2e 01 0c  00 00 00 00 00 00 00 f0   ........ ........
0060  fe 2e 01 00 00 00 00 01  00 00 00 00 00 00 00 00   ........ ........
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0080  00 00 00 00 00 00 00 f2  fe 2e 01 dc b3 2f 01 00   ........ ...../..
0090  00 00 00 fe 40 53 45 4c  45 43 54 20 69 64 2c 20   ....@SEL ECT id,
00a0  6e 61 6d 65 2c 20 6a 75  6d 69 6e 5f 6e 6f 2c 20   name, ju min_no,
00b0  6c 6f 67 69 6e 5f 69 64  2c 20 61 64 64 72 65 73   login_id , addres
00c0  73 2c 20 6d 6f 62 69 6c  65 5f 6e 6f 2c 20 65 6d   s, mobil e_no, em
00d0  61 69 6c 5f 61 11 64 64  72 20 46 52 4f 4d 20 63   ail_a.dd r FROM c
00e0  75 73 74 6f 6d 65 72 00  01 00 00 00 00 00 00 00   ustomer. ........
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0100  00 00 00 00 01 00 00 00  00 00 00 00 00 00 00 00   ........ ........
0110  00 00 00 00 00 00 00 00                            ........
```

```
Data (data), 226 bytes        Packets: 123 Displayed: 123 Marked: 0
```

a)

connect.pcap – Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

```
No. . Time      Source            Destination        Protocol Info
101 14.127031 192.168.100.88   192.168.100.87     TCP      vmodem > foliocorp [F
102 14.127681 192.168.100.87   192.168.100.88     TCP      foliocorp > vmodem [F
103 14.128298 192.168.100.88   192.168.100.87     TCP      vmodem > foliocorp [F
104 14.213811 192.168.100.36   192.168.100.255    NBNS     Name query NB PRTSVR<
```

```
⊞ Transmission Control Protocol, Src Port: vmodem (3141), Dst Port: folioco
⊟ Data (777 bytes)
    Data: 0309000006040000000006000285070000000F0000003205...
```

```
0040  06 00 02 85 07 00 00 00  0f 00 00 00 32 05 64 0f   ........ ....2.d.
0050  2c 08 01 00 00 00 00 00  00 00 00 00 00 00 7f 07   ,....... ........
0060  02 c1 03 0d 50 41 52 4b  2c 53 65 6f 6e 67 68 77   ....PARK ,Seonghw
0070  61 0e 36 35 30 35 30 32  2d 31 32 33 31 32 39 30   a.650502 -1231290
0080  08 53 48 50 41 52 4b 36  35 19 42 75 70 79 65 6f   .SHPARK6 5.Bupyeo
0090  6e 67 2d 67 75 20 49 6e  63 68 65 6f 6e 20 4b 6f   ng-gu In cheon Ko
00a0  72 65 61 0d 30 31 37 2d  33 39 31 32 2d 39 32 30   rea.017- 3912-920
00b0  31 0f 73 68 70 61 72 6b  40 64 61 75 6d 2e 6e 65   1.shpark @daum.ne
00c0  74 15 07 00 7f 07 02 c1  04 0a 4b 49 4d 2c 54 61   t....... ..KIM,Ta
00d0  65 68 65 65 0e 38 31 30  33 30 31 2d 32 34 37 33   ehee.810 301-2473
00e0  34 31 32 07 54 48 4b 49  4d 38 31 22 59 61 6e 67   412.THKI M81"Yang
00f0  6a 61 65 2d 64 6f 6e 67  20 53 65 6f 63 68 6f 2d   jae-dong  Seocho-
0100  67 75 20 53 65 6f 75 6c  20 4b 6f 72 65 61 0d 30   gu Seoul  Korea.0
0110  31 39 2d 38 34 32 31 2d  33 34 32 32 0d 74 68 6b   19-8421- 3422.thk
```

```
Data (data), 777 bytes        Packets: 123 Displayed: 123 Marked: 0
```

b)

Figure 3. SQL query packet capture windows: a) SQL query request packet from user, b) SQL query result packet from server

Table 4 describes the core part of the packet analysis related to the above figures. As shown in Figure 3 b) and Table 4, query result packets between server and client can be analyzed. The proposed system uses a data masking method for each column to protect data from a user who has lower security level.

Offset	Length (byte)	Description
offset + 0	2	1) Whole length of TCP Payload. 2) In case of long data, data is fragmented in IP layer.
offset + 4	2	When the response is sent from database server to client, if '*offset* + 4' has value '0x06' and '*offset* + 5' has value '0x04', then server sends the result of client query.
offset + 41	1	[Case of SELECT query] Number of column is assigned for query result. [Case of DML or DDL] Value '0x00' is assigned.
offset + 46	1	Starting point for column information. Value '0x01' is assigned. Each column information begins with value '0x01'.
offset + 85	1	Length of column name. $L \rightarrow len(colname)$
offset + 86	L	Column name.
offset + 85 + L + 9	1	Starting point for the next column information. Value '0x01' is assigned.

Table 4. The core part of the query result packet from server

All the accesses from client to database are managed by the 'Packet Manager'. In addition, it verifies user authentication and data integrity by using one-way hash function, message authentication code, session key encryption, and so on. Furthermore, it confirms access control policy for the packet through the ACPR (Access Control Policy Repository) and applies the policy which is verified by the 'Query Analyzer'. It can deny access to the data for unauthorized users. The data which should be hidden in the database by defined policies can be protected through masking process in the 'Data Masker'. After data masking, results are transmitted to the client. At this time, it can be reasonably assumed that the network between the DFS and databases (or database servers) is safely protected by using a secure channel such as SSL.

For detailed access control, the policy can be set according to items such as IP address, host, database user, database table, time, date, application, login user, command, database instance, database session, database SQL, and so on. Data columns which should be masked can be classified and set for each table, and policy for them can also be set for each table.

5 IMPLEMENTATION OF THE PROPOSED SYSTEM

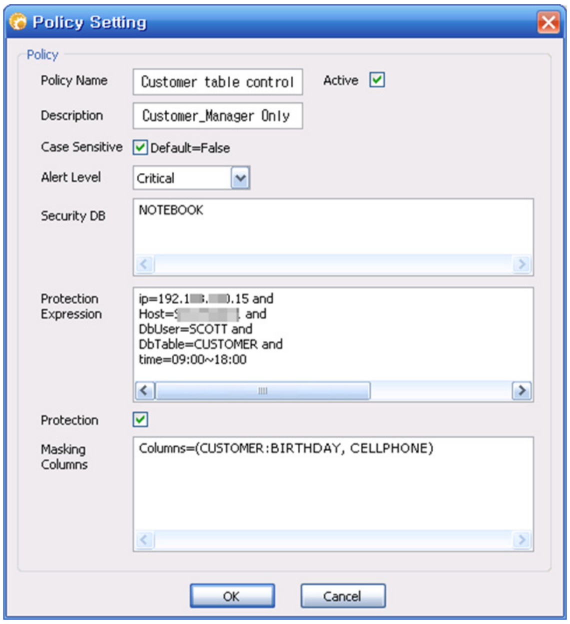
In this implementation for the database composition environment, 'Oracle', which is a commercial DBMS, is used, and we can access the Oracle database [24] by using SQL*Plus [25]. Windows 2000 and Oracle Client Release 9.x version are used for the client and DFS. In addition, UNIX and Oracle9i Enterprise Edition Release 9.x are used for the ACPR and object database of control, and Windows 2000 and Visual C++ 6.0 are used for the development composition environment.

In order to obtain packets and mask data in the network, the 'Windows Packet Filter Kit' developed by Vadim V. Smirnov is used. Windows Packet Filter Kit is a kind of library which can obtain packets through the NDIS (Network Driver Interface Specification) connecting Windows's TCP/IP layer and network adapter [26, 27]. In addition, 'General SQL Parser' from `sqlparser.com` is used for parsing the obtained SQL query [28].

To communicate with client programs, the Oracle server applies accesses according to the two separate cases, which are accesses of an external user using TCP/IP and accesses of an internal user using BEQ. This paper focuses on TCP/IP only as the accesses of an external user in databases.

This proposed system uses a masking method by column unit to protect data from unauthorized users according to the result of a query in a packet between a client and server. This method first analyzes packets of a query received from the database server and decides whether the data containing them needs to be hidden. If it needs to be masked, it changes the first character into a '*' mark and the other characters into a 'NULL' string. This is different from the 'data encryption' method.

In the case of data encryption, encrypted data can be longer than original data according to the encryption algorithm. This means that content and length of the structured protocol is changed in the Oracle database server. Then, the DFS should regenerate packets entirely to correspond to the Oracle protocol. Furthermore, payload length information in the TCP header should be changed. Thus, the proposed system uses the masking method which is more effective in terms of cost, and it is also more appropriate for the purpose of hiding data column values from unauthorized users simply. Figure 4 shows the detailed set of the access control policy and masking column for access.



Policy Name	Customer table control
Active	✓
Description	Customer_Manager Only
Case Sensitive	✓ Default=False
Alert Level	Critical
Security DB	NOTEBOOK
Protection Expression	ip=192.1[redacted].[redacted]0.15 and Host=S[redacted] and DbUser=SCOTT and DbTable=CUSTOMER and time=09:00~18:00
Protection	✓
Masking Columns	Columns=(CUSTOMER:BIRTHDAY, CELLPHONE)

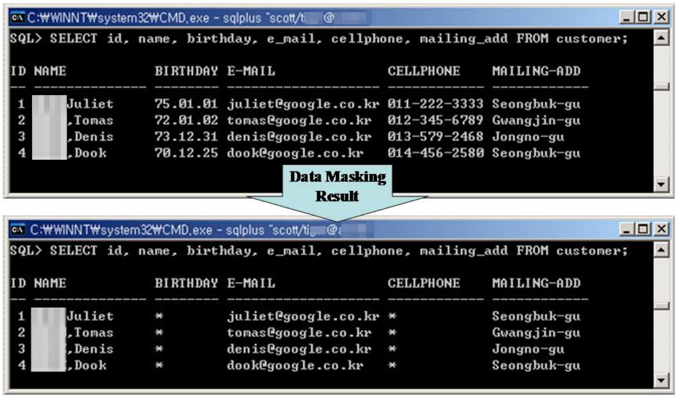
Figure 4. An example of access control policy set

As shown in Figure 4, the administrator can set a wanted control part in 'Protection Expression', and the control policy can be set for all items mentioned in Section 4.2. In the above policy set window, 'BIRTHDAY' and 'CELLPHONE' among the result data are set for masking. Figure 5 describes the data masking result for the above policy set.

Figure 5 is a test result for SQL query performance in a client by using SQL*Plus [25] after applying the proposed security model. For the query result, 'BIRTHDAY' and 'CELLPHONE' column data changed and are marked with '*' instead of the original personal data.

6 COMPARISON AND EVALUATION

This section compares the existing database security system [18] and the proposed access control system. The existing database security system for various sizes of


C:₩WINNT₩system32₩CMD.exe - sqlplus "scott/t[redacted]@[redacted]
Data Masking Result
C:₩WINNT₩system32₩CMD.exe - sqlplus "scott/ti[redacted]@[redacted]

```
SQL> SELECT id, name, birthday, e_mail, cellphone, mailing_add FROM customer;

ID NAME           BIRTHDAY E-MAIL               CELLPHONE    MAILING-ADD
-- -------------- -------- -------------------- ------------ ------------
 1 [redacted]Juliet      75.01.01 juliet@google.co.kr  011-222-3333 Seongbuk-gu
 2 [redacted],Tomas      72.01.02 tomas@google.co.kr   012-345-6789 Gwangjin-gu
 3 [redacted],Denis      73.12.31 denis@google.co.kr   013-579-2468 Jongno-gu
 4 [redacted],Dook       70.12.25 dook@google.co.kr    014-456-2580 Seongbuk-gu
```

```
SQL> SELECT id, name, birthday, e_mail, cellphone, mailing_add FROM customer;

ID NAME           BIRTHDAY E-MAIL               CELLPHONE    MAILING-ADD
-- -------------- -------- -------------------- ------------ ------------
 1 [redacted]Juliet      *        juliet@google.co.kr  *            Seongbuk-gu
 2 [redacted],Tomas      *        tomas@google.co.kr   *            Gwangjin-gu
 3 [redacted],Denis      *        denis@google.co.kr   *            Jongno-gu
 4 [redacted],Dook       *        dook@google.co.kr    *            Seongbuk-gu
```

Figure 5. Execution result of SQL $*$ plus column masking

data groups has several problems, as mentioned, and the proposed method supports security for data and provides integrated management of databases and detailed access control at the same time. A more detailed analysis is described in Table 5.

Items	Existing System [18]	Oracle Database (8i-11g)	Proposed System
Performance reduction due to the data en/decryption time	Exist	Exist	Non-exist
Time delay for policy application	Exist	Non-exist	Non-exist
Confidentiality for data itself	Impossible	Possible	Possible
Integrated management of heterogeneous database policy	Impossible	Possible	Possible
Control for specific column of query result	Possible	Possible	Possible
Control for specific time or SQL query	Impossible	Possible	Possible
Data Indexing	Possible	Possible	Possible

Table 5. A comparison between the existing and proposed systems

In Table 5, the first and second items are related to the efficiency of data processing. The existing system in [18] and Oracle databases have the data encryption module internally. So, the performance is not very good; but in the proposed system, the module for data masking is working independently. So, there is no reduction of performance. Confidentiality for data is a very important issue to guarantee the database security because database security is of no significance if it is not ensured. However, in the existing methods, there are no protections to each data. Therefore, if packets are stolen while transmitting data, all the packet contents can be exposed. In the proposed method, data can be hidden through data masking. Thus, contents of important data can be protected even though the packet is stolen.

In addition, existing methods take time to data disposal because it should satisfy three policies through two phases [18] while the proposed model has almost no time delay because it applies simple masking technique instead of data encryption and decryption.

In terms of integrated management of a heterogeneous database policy, in the existing method, permission conflict problem can occur between different databases in extending permission grant structure. Moreover, integrated policy cannot be applied because it is given by each database. In the proposed method, independent data executions are possible because data is operated in front of databases (or a database farm) directly. Through this physical feature, the extension of permission structure and integrated management can be possible.

In addition, as shown in the table above, the existing method can control specific column data by using three kinds of policies (MAC, RBAC, S-DAC), but it cannot deal with other specific items, such as time or SQL query detail. On the other hand, the proposed system can apply control not only for specific columns, but also for the time, SQL, and the user by using packet monitoring. The packet monitoring method can also be a solution for various access requirements, which is one of the problems mentioned in the introduction.

Finally, confidentiality of data can be guaranteed when data encryption method is used, but the time can be delayed for encrypting data. It also makes hard to set data indexing. To solve this, in terms of data indexing, the former method controls the access to each column by applying the several kinds of methods, regardless of data confidentiality assurances. Therefore, data is not encrypted, and data indexing is possible while the proposed method executes data masking before the data are transmitted through network. Thus, data confidentiality and data indexing can be ensured simultaneously even though data is not encrypted.

To examine time delay when data is masked, we have the following experiment in this paper. The experiment is performed to retrieve data in a table which is composed of 8 columns; we run it 10 times from 1 000 to 10 000 rows. We have four data groups to have masking experiment such as no masked data, one-column masked data, two-column masked data and three-column masked data. To inspect more diverse cases, we use various data types such as integer type data for column A, encrypted data for column B and text type data for column C, and each length averages of data types which would be masked are 14 bytes, 32 bytes and 19.875 bytes. Thus, averages of the masked data length of these data group are 0 bytes for 'no masked data group', 14 bytes for 'column A', 46 bytes for 'column A + column B', and 65.875 bytes for 'column A + column B + column C' for the experiment.

The results of this experiment with the above data groups are shown in Figure 6.

In Figure 6, the number which is shown on the left side in each group is the result of masking reply time for 'No masked data' and the right side number is for 'Column A + B + C'. As shown here, on the whole the reply time results of no masked data are shorter than those of others. However, the reply time is not always increased in proportion to column size as shown this figure. Even the result of no

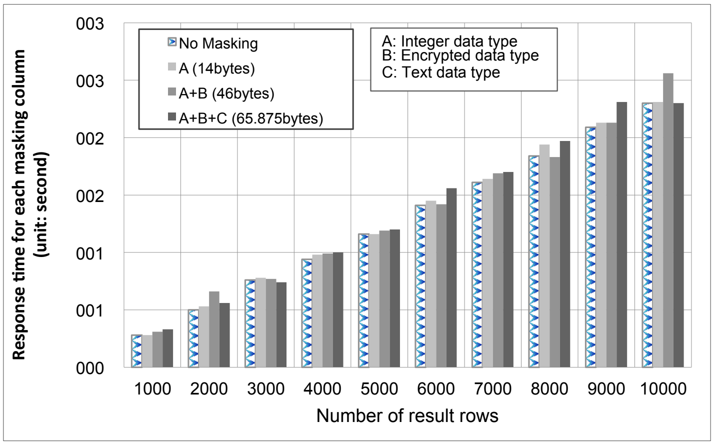


Figure 6. Reply time graph for each masked column group

masked data takes a longer time than that of others in this graph. It means that the reply time is not affected much by masking.

In addition, the detailed access control is possible such as access time, particular SQL control, and permission grant of the result rows and etc., because this system is placed in front of the target databases or a database farm, and extension or integration management of the permission structure can be performed effectively since policy grant and management are separated from the database.

Table 6 shows the access control state according to the permission in the database for the existing database security system and the proposed system. We generate a random database login account and SQL queries and check several security requirements. In this table, 'O' means that the access can be controlled for each SQL query operation, and 'X' means that it cannot be controlled.

As shown in Table 6, the existing system cannot control the security requirements regardless of whether user is authorized or unauthorized. However, the proposed system can control it precisely. Particularly in the requirement 'Access by time slot', access is controlled accurately for the data, except the time assigned by each user. The data masking can also be performed if common query access that is controlled by the column is impossible in the proposed system.

Table 7 compares the advantages and disadvantages of the related commercial solutions.

As shown in Table 7, first, in case of the access control and/or auditing method used in Chakra$^{\text{TM}}$, Middleman$^{\text{TM}}$, DBSafer$^{\text{TM}}$, DB-i$^{\text{TM}}$and dGriffin$^{\text{TM}}$, there are several merits such as no performance degradation and user management function while secure data management is not possible. Second, in case of data encryption method using in Cube One$^{\text{TM}}$, D'AMO$^{\text{TM}}$, SafeDB$^{\text{TM}}$, and Secure.Data, data is relatively secure while there is some burden to manage the data. Finally, in case

Security Requirements	Authorization	Existing System [18]		Proposed System	
		Common column	Target column	Common column	Target column
Common access	AU	X	X	X	X
	UAU	O	X	O	O
Access by time slot	AU	X	X	X	X
	UAU	X	X	O	O
Access by DB account theft	AU	X	X	O	O
	UAU	X	X	O	O
Access by IP address theft	AU	X	X	O	O
	UAU	X	X	O	O
Control column access for data masking	AU	X	X	X	O
	UAU	X	X	X	O

AU: Authorized User, UAU: Unauthorized User

Table 6. Comparison of database access controls

Related commercial solutions	Security Methods	Advantage	Disadvantage
Chakra™[5], Middleman™[6], DBSafer™[7], DB-i™[8], dGriffin™[9]	Access Control and/or Auditing	Performance of DBMS is guaranteed. Operation is convenient. User access management is possible. ([7, 9])	Data security is impossible. Detailed access is not allowed for specific column or low. Dynamic access request cannot be processed. ([7, 9])
Cube One™ [10], D'AMO™ [11], SafeDB™ [12], Secure.Data™ [13]	Data Encryption	Data security of itself is guaranteed. ([5, 8])	Performance of DBMS can be reduced. Indexing is ineffective. ([5, 8])
Oracle Database Firewall [30]	Access Control, Data Encryption, Data Masking	Diverse methods exist. User can select encryption method and key. ([29, 30])	Encryption or masking function performs in the system inside. Performance can be reduced. ([29, 30])

Table 7. Comparison of dis/advantage according to the related commercial solutions

of Oracle supporting diverse data protection method, there are some good features which are diverse methods to use data and key. The solution is composed of software type, and the data encryption or masking is internally executed in the system. Thus, it is not used widely because it may decrease the effectiveness [29].

7 CONCLUSION

Recent research on security systems for various sizes of data groups focused on several requirements related to data size. However, it could not assure data confidentiality in databases. In addition, in defining data groups, overhead could occur, and adding the policy could also cause a decrease of performance efficiency and duplication of the policy. Moreover, integrated management would not be possible for various databases.

Thus, in this paper, the database firewall server was proposed to solve these problems by providing confidentiality, performance efficiency, and integrated management for enforcing security policies. The proposed security system uses the authentication method to prevent the alteration of the accessed user information to the database and exchanges the encrypted SQL authentication code between CAA and DFS to ensure the integrity of the requested SQL from the client. In addition, it reduces the inconvenience of SQL generation owing to the strong access control policy by each column and implements the data masking technique for data access according to the permission.

Thus, the contributions of this paper are as follows:

1. Confidentiality for data and effective data processing.
2. Effective Query and Answer performance for access control.
3. Integrated management for heterogeneous database policies.
4. Detailed access control according to diverse access requirements.
5. Indexable data processing.

First, this proposed system can protect the data and make an index by using data masking method at the same time. Also, user can manage the time to access databases and obtain the data with this data masking method. Here, we used a packet analyzing method by monitoring. So, detailed access control is possible at the packet level. In addition, all these processes are based on the reference monitor model. So, we designed and implemented that this proposed system can monitor entire access to the databases in order to manage the heterogeneous databases, which may have different policies, efficiently and integratively. Therefore, the proposed system can control the user access in detail and the data masking module and masking data decision module is implemented separately. So, it can reduce the time delay to apply policies including the changed policies dynamically. Moreover, our system can protect the data from the attack caused by undefined access because it can manage all requests from user applications to databases through the reference monitor.

For future research, we need to build a standardized access control policy which can be applied to all databases regardless of the type. Furthermore, we need to analyze the protocol in more detail for strong access control. In addition, we need to study effective management for access control policies when the databases for audit or management are increased.

Acknowledgment

This work was supported by the Soonchunhyang University Research Fund.

REFERENCES

[1] Borking, J. J.—Van Eck, B. M. A.—Siepel, P.: Intelligent Software Agents: Turning a Privacy Threat into a Privacy Protector. Information and Privacy Commissioner of Ontario 1999.

[2] Lee, H. G.—Lee, S. M.—Nam, T. Y.: Database Encryption Technology and Current Product Trend. Electronics and Telecommunications Trend Analysis, Vol. 22, 2007, No. 1, pp. 105–113.

[3] Nagarajan, A.—Jensen, C. D.: A Generic Role Based Access Control Model for Wind Power Systems. Advances in Trust Management, JoWUA, Vol. 1, 2010, No. 4, pp. 35–49.

[4] Lu, Y.—Tsudik, G.: Privacy-Preserving Cloud Database Querying. Frontiers in Trust Management, JISIS, Vol. 1, 2011, No. 4, pp. 5–25.

[5] Chakra^{TM}, `http://www.warevalley.com/` (2012).

[6] Middleman^{TM}, `http://www.banet.co.kr/` (2012).

[7] DB Safer^{TM}, `http://www.pnpsecure.com/` (2012).

[8] DB-i^{TM}, `http://www.somansa.com/` (2012).

[9] dGriffin^{TM}, `http://www.sinsiway.com/` (2012).

[10] CubeOne^{TM}, `http://www.eglobalsys.co.kr/` (2012).

[11] DAmo^{TM}, `http://www.pentasecurity.com/` (2012).

[12] SafeDB^{TM}, `http://www.initech.com/` (2012).

[13] Secure.Data^{TM}, `http://www.protegrity.com/` (2012).

[14] Davida, G. I.—Wells, D. L.—Kam, J. B.: A Database Encryption System with Subkeys. ACM Transactions on Database systems, Vol. 6, 1981, No. 2, pp. 312–328.

[15] Piattini, M.—Fernandez-Medina, E.: Secure Databases: State of the Art. Security Technology, Proceedings of the IEEE 34th Annual 2000 International Carnahan Conference 2000, pp. 228–237.

[16] Ferraiolo, D. F.: Role-Based Access Control. Artech House, Computer Security 2003.

[17] Elovici, Y.—Waisenberg, R.—Shmueli, E.—Gudes, E.: A Structure Preserving Database Encryption Scheme. Proceedings of the Secure Data Management in a Connected World 2004 (SDM 2004), LNCS No. 3178, pp. 28–40.

[18] Jeong, M. A.—Kim, J. J.—Won, Y. G.: A Flexible Database Security System Using Multiple Access Control Policies. LNCS No. 2736, 2003, pp. 876–885.

[19] Bertino, E.—Sacks-Davis, R.—Ooi, B. C.—Tan, K. L.—Zobel, J.—Shidlovsky, B.—Cantania, B.: Indexing Techniques for Advanced Database Systems. Kluwer Academic Publishers 1997.

[20] Keesling, D.—Womack, J.: Oracle9i Database Administration Fundamentals II, Production 2.0, D37492, Oracle Corporation 2002.

[21] Kim, S. .—Nam, G. W.—Kim, S. C.: Filtering Unauthorized SQL Query by Uniting DB Application Firewall with Web Application Firewall. Proceedings of the Korea Institutes of Information Security and Cryptology Conference 2003, pp. 686–690.

[22] Jang, G. O.—Koo, H. O.–Oh, C. S.: Detection of Internal Illegal Query Using Packet Analysis. The Korean Society of Computer and Information, Vol. 10, 2005, No. 3, pp. 259–265.

[23] Ethereal, Inc., `http://www.ethereal.com/` (2012).

[24] Steiner, D.—Moore, V.: Oracle9i Net Services Administrator's Guide. Release 2 (9.2), Part No. A96580-02, Oracle Corporation 2002.

[25] Gennick, J.: Oracle SQL*Plus: The Definitive Guide. O'Reilly 2005.

[26] Molnár, K.—Kyselák, M.: Application Programming Interface Offering Classification Services to end-User Applications. Proceedings of the International Conference on Systems and Networks Communication 2006 (ICSNC '06), p. 49.

[27] Windows Packet Filter Kit. `http://www.ntkernel.com/` (2012).

[28] SQL Parser. `http://www.sqlparser.com/` (2012).

[29] Korea Database Agent (KDB). A Guideline for Database Security, Korea Database Agent 2011.

[30] Oracle Corporation 2011. Oracle Database 11g Release 2 (Oracle Database 2 Day + Security Guide).

Eun-Ae Cho received her Ph. D. degree in the Department of Computer Science and Engineering, Korea University in 2009. She was a researcher at Purdue University, West Lafayette, IN, USA from 2009 to 2011. She is a senior researcher with Samsung Electronics. Her research areas include information security, home network security, user privacy, and policy management.

Chang-Joo Moon received his Ph. D. degree in the Department of Computer Science and Engineering, Korea University, Seoul, Republic of Korea in 2004. He is an Associate Professor in the Department of Aerospace Information Engineering, Konkuk University, Seoul. His research areas include real-time embedded systems, computer security, and system integration.

Dae-Ha Park received his Ph. D. degree in the Department of Computer Science and Engineering, Korea University, Seoul, Republic of Korea in 2004. He is an Associate Professor in the Department of Information Management and Security, The Cyber University of Korea, Seoul. His research areas include database security, computer security, and information security management systems.

Kangbin Yim received his Ph. D. degree in the Department of Electronics Engineering, Ajou University, Suwon, Korea in 2001. He is an Associate Professor in the Department of Information Security Engineering, Soonchunhyang University, Asan, Korea. He was a visiting professor at Purdue University in 2011. His research interests include vulnerability assessment, malware analysis, and security on cyber physical systems and embedded systems.