Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Encryption-based multilevel model for DBMS

*Ahmed I. Sallam, El-Sayed El-Rabaie, Osama S. Faragallah**

Department of Computer Science & Engineering, Faculty of Electronic Engineering, Menouf 32952, Egypt

ARTICLE INFO

Article history:
Received 7 September 2011
Received in revised form
31 December 2011
Accepted 13 February 2012

Keywords:
Database security
Relational database
Multilevel security
SeaView model
Jajodia–Sandhu model
Smith–Winslett model
MLR model
Belief-consistent model
Multilevel database performance

ABSTRACT

In this paper, we propose an encryption-based multilevel model for database management systems. The proposed model is a combination of the Multilevel Relational (MLR) model and an encryption system. This encryption system encrypts each data in the tuple with different field-key according to a security class of the data element. Each field is decrypted individually by the field-key of which security class is higher than or equal to that of the encrypted field-key. The proposed model is characterized by three achievements: (1) utilizing an encryption system as an additional security layer over the multilevel security layer for the database, (2) reducing the multilevel database size, and (3) improving the response time of the data retrieval from the multilevel database. Also this paper summarizes our efforts in implementing a working multilevel secure database prototype. This prototype is used as a research tool for studying principles and mechanisms of the encryption-based multilevel model and multilevel secure database (MLS/DBMS) models (SeaView, Jajodia–Sandhu, Smith–Winslett, MLR, and Belief-Consistent Model). This prototype is implemented to be used to perform a series of experiments to measure the performance cost for applying encryption in multilevel database security.

1. Introduction

In multilevel database systems, data items and subjects have been assigned to classification levels, such as TS (Top Secret), S (Secret), C (classified), U (Unclassified). The classification levels are ordered as TS > S > C > U.

Access by subjects is restricted by mandatory access controls expressed as "no read up, no write down to follow the well-known Bell and LaPadula model. Subject can read the object that has the same classification level or lower and can write on the objects at the same level only" (Bertino and Sandhu, 2005; Imran and Hyder, 2009).

Many models for extending the standard relational model to deal with multilevel relations have been proposed. The SeaView (Pranjic et al., 2002) model was the first formal MLS secure relational database designed to provide mandatory security protection. The SeaView model extended the concept of a database relation to include the security labels. A relation that is extended with security classifications is called a multilevel relation. The Jajodia–Sandhu (Cuppens and Gabillon, 1999) model was derived from the SeaView model. It was shown by Jajodia and Sandhu that the SeaView model can result in the proliferation of tuples on updates and the Jajodia–Sandhu model addresses this shortcoming. The Smith–Winslett (Rjaibi and Bird, 2004) model was the first model to extensively address the semantics of an MLS database. The MLR (Lee et al., 2004; Sandhu and Chen, 1998) model is substantially based on the Jajodia–Sandhu model, and also integrates the belief-based semantics of the Smith–Winslett model. It was shown that all of the aforementioned models can present users with some information that is difficult to interpret. Consequently, the Belief-Consistent MLS (BCMLS) (Pranjic et al., 2003; Jukic et al., 1999; Jukic and Vrbsky, 1997) model addresses these concerns by including the

* *Corresponding author.*
E-mail address: osam_sal@yahoo.com (O.S. Faragallah).

0167-4048/$ – see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cose.2012.02.008

semantics for an unambiguous interpretation of all data presented to the users.

Several commercial database systems like DB2 (IBM) and ORACLE support encryption in their database management systems. In DB2 (IBM), encryption has been added by implementing SQL built-in functions that allow the application to encrypt and decrypt data. When data is inserted into the database it can be encrypted using an encryption password supplied by the user. When the data is retrieved, the same password must be supplied to decrypt the data. In ORACLE, transparent data encryption enables you to encrypt sensitive data, such as credit card numbers, stored in table columns. Encrypted data is transparently decrypted for a database user who has access to the data. Even if the encrypted data is retrieved, it cannot be understood until authorized decryption occurs, which is automatic for users authorized to access the table. When a table contains encrypted columns, a single key is used regardless of the number of encrypted columns. This key is called the column encryption key. The column encryption keys for all tables, containing encrypted columns, are encrypted with the database server master encryption key and stored in a dictionary table in the database.

Our principal objective in this paper is to propose an encryption-based multilevel database model by adding an encryption algorithm to the MLR multilevel model. The encryption system is used as additional security layer over the multilevel security layer for the database which provides high level of security and to solve problems associated with MLR model. Table 1 shows a comparison between the proposed encryption-based multilevel database model and the commercial database systems like DB2 (IBM) and ORACLE that support encryption in their database management systems.

The work presented in this paper offers several major contributions to the field.

1- Adding encryption system as additional security layer over the multilevel security layer for the database which provides high level of security and robustness against database attacks.
2- Reducing the multilevel database size by removing the attributes classification columns and encrypting the attributes by field-key according to its security level.
3- Simplifying the complexity of the multilevel database design by avoiding the creation of the additional columns for attributes classification.
4- Implementing a prototype to be used to perform a series of experiments to measure the performance cost for applying encryption in multilevel database security.

The rest of this paper is organized as follows. Section 2 illustrates the proposed encryption-based multilevel database model. Section 3 shows the implementation of DML (Data Manipulation Language) operations for the proposed model. Section 4 presents the performance study that was instrumented to compare the multilevel secure database (MLS/DBMS) models. Section 5 gives the analysis of the experimental results of the performance study. Section 6 concludes the paper and outlines the future work.

2. The proposed encryption-based multilevel database model

Many multilevel relational models have been proposed and these different models offer different advantages (Rask et al., 2005; Dave, 2008; Garuba, 2003). The MLR model is the most powerful model among the multilevel relational models. So we refine several of the best ideas from MLR model and add new ones to build our proposed Encryption-Based Multilevel Model.

2.1. *MLR model*

Definition 2.1.1. A multilevel relation scheme is denoted by $R(A_1,C_1,A_2,C_2,\ldots,A_n,C_n,TC)$, where R is the multilevel relation, each A_i is a data attribute, each C_i is a classification attribute for A_i, and TC is the tuple-class attribute (Garuba, 2004).

Definition 2.1.2. A relation instance, denoted by $r(A_1,C_1,A_2,C_2,\ldots A_n,C_n,TC)$, is a set of distinct tuples of the form $(a_1,c_1,a_2,c_2,\ldots,a_n,c_n,tc)$.

Definition 2.1.3. A database is a collection of relations. A database state is a collection of all relation instances of a database at a particular time. Table 2 illustrates an example for data stored in multilevel database security in the MLR model format.

Table 1 – Comparison between the proposed encryption-based multilevel database model and the commercial database systems like DB2 (IBM) and ORACLE.

Criteria	Model		
	Encryption-based multilevel database	DB2 encrypted fields	ORACLE transparent data encryption
Encryption in multilevel security	Supported	Not supported	Not supported
Encryption type	Cell-based encryption (one password per cell)	Column-based encryption (one password per column)	Column-based encryption (one password per column)
Encryption key	Key is managed by database engine	Key provided by the user at runtime	Key provided by the user at runtime

Table 2 – The MLR model.

Employee	C-Employee	Department	C-Department	Salary	C-Salary	TC
Ahmed	U	Accounting	U	7000	U	U
Ahmed	S	Accounting	S	7000	S	S
Mohamed	TS	Sales	TS	10,000	TS	TS

We now give a formal description of the above intuitive ideas (Zuo et al., 2007; Pan, 2008). For all instances $r(A_1,C_1,A_2,C_2,\ldots,A_n,C_n,TC)$ and for all tuples $t \in r$, the data are interpreted as follows:

1- Apparent primary key A_1 and its classification attributeC_1.
$t[A_1,C_1]$ identifies an entity in r and also gives the class level of the entity.
$t[C_1] = c_1$ means the entity is created by a c_1-subject and can only be deleted by c_1-subjects. The entity is called a c_1-entity. In Table 2 the apparent primary key is [Employee, C-Employee].

2- Tuple-Class attribute TC.
$t[TC] = tc$ with $t[C_1] = c_1$ means that t is added by a tc-subject and all data in t are accepted by tc-subjects. Absence of t means the c_1-entity is not accepted by tc-subjects. In Table 2 the Tuple-Class is [TC].

2.2. *Encryption-based multilevel model definition*

In this research, we design a novel multilevel database security model, Encryption-Based Multilevel Model, to solve problems associated with MLR model.

In the proposed model, when the database administrator creates a level to be used in the multilevel database, the database engine will automatically create a symmetric key for this level. The symmetric key will be stored in the multilevel database to be used for encrypting and decrypting the data element that is classified by the level associated to this symmetric key. A multilevel relation scheme is denoted by $R(E_{C1}(A_1),E_{C2}(A_2),\ldots,E_{Cn}(A_n),TC)$, where each A_i is a data attribute and each C_i is a classification attribute for A_i. Table 3 illustrates an example for data stored the proposed Encryption-Based Multilevel Model.

In the proposed model, adding the encryption system to the MLR model led to solve the problems in the MLR model by removing the classification attributes from the multilevel database and then reducing the multilevel database size and making the database administration easier.

The encryption keys are stored as a hidden property for the classification levels of the multilevel database security. The database administrator cannot read the encryption keys. He can only read the classification levels of the multilevel database security. In our approach cashing has an impact that should be taken into our account. The impact of the cashing is due to storing the decrypted data during the transaction execution in the memory as a plain text which is a problem. Our approach solves the problem of cashing as follows:

1. Making the part of the memory that holds the decrypted data to be blocked so that it can only be accessed only from the database engine instance.
2. Supporting multilevel security to the data so the user can see only the data in his level and lower level. Supporting multilevel security in our approach overcomes the problem of caching because it generates a security layer that manages the data access in the memory.

3. Manipulation

There are five data manipulation statements in the proposed model because we modify MLR data model which contains five operations for manipulating data. Four of them are the traditional SQL statements INSERT, DELETE, SELECT, and UPDATE. The fifth statement is UPLEVEL which is new in the MLR data model.

3.1. *The INSERT statement*

The INSERT statement executed by a subject, with class level L, has the following general form:

INSERT INTO R $[(A_{j_1}[,A_{j_2}]\ldots)]$ VALUES $(a_{j_1}[,a_{j_2}]\ldots)$

Symbol explanation: R is the relation name, $[(A_{j1}[,A_{j2}]\ldots)]$ are the attributes names and $1 \le j_1,j_2 \ldots \le n$.

Each INSERT data manipulation can insert at most one tuple into the relation R. The inserted tuple t is constructed as follows:

For all attributes in database relation,

1. If there is an attribute A_i in the attribute list of the INTO clause, the data value a_i will be encrypted by field-key according to the class level of the subject who executes the insert statement.
2. If A_i is not in the attribute list of the INTO clause, set the data value null.
3. The tuple-class will be set to the class level of the subject who executes the insert statement.

3.2. *The DELETE statement*

The DELETE statement executed by a subject, with class level L, has the following general form:

DELETE FROM R [WHERE P]

Symbol explanation: R is the relation name, assuming relation R has data attributes $A_1,\ldots,A_n$; P is the predicate expression that may include the delete conditions. Only tuples $t \in r$ with $t[TC] = L$ are decrypted by key according to the classification level of the subject who executes the delete statement.

For those tuples $t \in r$ that satisfy the P predicate expression, r is changed as follows:

Table 3 – The encryption-based multilevel database model.

Employee	Department	Salary	TC
□타越丽番搵垌5Hı	□타越丽番搵垌5Hı	□타越丽番搵垌5Hı	U
鑛캭□征肤跕□음ı	鑛캭□征肤跕□음ı	鑛캭□征肤跕□음ı	S
嗅□爭穧摧瑳口퀵ı	嗅□爭穧摧瑳口퀵ı	嗅□爭穧摧瑳口퀵ı	TS

1- Create a temporary tuple for the decrypted data to store the deleted tuple during the execution of the delete statement.
2- Tuple that satisfies the predicate expression will be deleted.
3- If there is tuple at higher level that has attribute depends on attribute in the deleted tuple, the value of this attribute will be set to null.

Deleting lower-level tuples may lead to setting data attributes to null at higher levels. This propagation because of the concept of data-borrow that was introduced in the MLR multilevel database model (Lee et al., 2004). In the data-borrow the higher level (borrower) can borrow the value that is currently owned by the lower level (owner). Therefore, in case some changes happen to the owner, corresponding changes should happen to the borrower.

3.3. *The SELECT statement*

The SELECT statement executed by a subject, with class level L, has the following general form:

SELECT $B_1[, B_2]$…FROM $R_1[, R_2]$…[WHERE P] [AT $c_1[, c_2]$…]

Symbol explanation: $R_1[,R_2]$… are the relations names; $B_1[,B_2]$… are the attributes names in $R_1[,R_2]$…, each B_i is a data attribute or classification attribute or tuple-class attribute; P is the predicate expression that may include the select conditions, AT $c_1[,c_2]$… is used to add the values of the classification levels that are lower or equal the class level of the user to the predicate P. Only those tuples $t \in r_1, r_2, \ldots$ that have $t[TC] = L$ will be decrypted by key according to the classification level of the subject who executes the select statement and will be taken into the calculation of P.

The "GROUP BY… HAVING" – Statements are considered in the aggregation functions like (count, sum) in the SELECT – statement which are not taken into our account in this paper. The "AT"-clause was used to add the values of the classification levels that are lower or equal the class level of the user to the predicate P. For those tuples $t \in r$ that satisfy the P predicate expression:

If decrypted tuple satisfies the predicate expression, this tuple will be included in the result of the SELECT statement.

3.4. *The UPDATE statement*

The UPDATE statement executed by a subject, with class level L, has the following general form:

UPDATE R SET $A_{j_1} = a_{j_1} [, A_{j_2} = a_{j_2}]$…[WHERE P]

Symbol explanation: R is the relation name, $A_{j1}, A_{j2}, \ldots$ are the data attributes names and P is the predicate expression that may include the update conditions. Only tuples $t \in r$ with $t[TC] = L$ will be decrypted by key according to the classification level of the subject who executes the update statement and will be taken into the calculation of P.

For decrypted tuples $t \in r$ that satisfies the predicate P, r is updated as follows:

1- Create a temporary tuple for the decrypted data to store the deleted tuple during the execution of the delete statement.
2- If there are no attributes of the primary key in the SET clause, the following steps will be followed.
   For all attributes in the SET clause:
   a- Encrypt the attribute value and update the tuple.
   b- If there is a tuple that has attribute depends on attribute in the updated tuple, the value of this attribute will be encrypted and updated.
3- If there are attribute of the primary key in the SET clause, the following steps will be followed.
   a- Encrypt the attribute value and update the tuple;
   b- If the primary key class is equal to the class of the subject who executes the update statement, all tuples that have the same primary key will be deleted.

3.5. *The UPLEVEL statement*

The UPLEVEL statement executed by a subject, with class level L, has the following general form:

UPLEVEL R GET A_{j_1} FROM $c_{j_1} [, A_{j_2}]$ FROM c_{j_2}…[WHERE P]

Symbol explanation: R is the relation name, $A_{j1}, A_{j2}, \ldots$ are the data attributes names, $2 \le j_1, j_2 \ldots \le n$, $c_{j1}, c_{j2}, \ldots$. are the values of classification levels for $A_{j1}, A_{j2}, \ldots$, respectively and P is the predicate expression that may include uplevel conditions. Only tuples $t \in r$ with $t[TC] \le L$ will be decrypted by key according to the classification level of the tuple Key[TC], and will be taken into the calculation of P.

For decrypted tuples that have at least one tuple $t' \in r$ that satisfies the predicate P, a L-tuple t is constructed as follows:

1- Create a temporary tuple for decrypted data to store the deleted tuple during the execution of the uplevel statement.

2- If A_i is in GET clause, get data value from the tuple with class equal to class in FROM clause and encrypt it.
3- If A_i is not in the GET clause, set data value to null.

After tuple t is constructed the following procedure will be applied:

1- If there is a tuple with its primary key equal to the primary key of the constructed tuple and its class equal to the class of the subject who executes the uplevel statement, this tuple will be replaced by the constructed tuple.
2- If there is no tuple with its primary key equal to the primary key of the constructed tuple and its class equal to the class of the subject who executes the uplevel statement, the constructed tuple will be added to the relation.

4. Performance study

This section describes our performance experiments to determine the relative performance of the multilevel database models (SeaView, Jajodia–Sandhu, Smith–Winslett, MLR, and Belief-Consistent Model) and the proposed encryption-based MLS model to illustrate the impact of varying the size and structure of the database on the performance of these models.

The machine that is used for our implementation consists of CPU speed 2.2 GHz, physical RAM size 3 GB and hard disk size 320 GB. The software that is used in our implementation is Microsoft SQL server 2008 R2 and the experiments measurements were captured at the machine using a monitoring tool provided by Microsoft SQL server. We make a performance evaluation for the encryption algorithms that are built in Microsoft SQL server 2008 R2 to choose the suitable encryption algorithm that will be used in our proposed encryption-based MLS model. From Fig. 1, we observe that the AES_128 encryption algorithm supports encryption in the multilevel database system with a good performance cost.

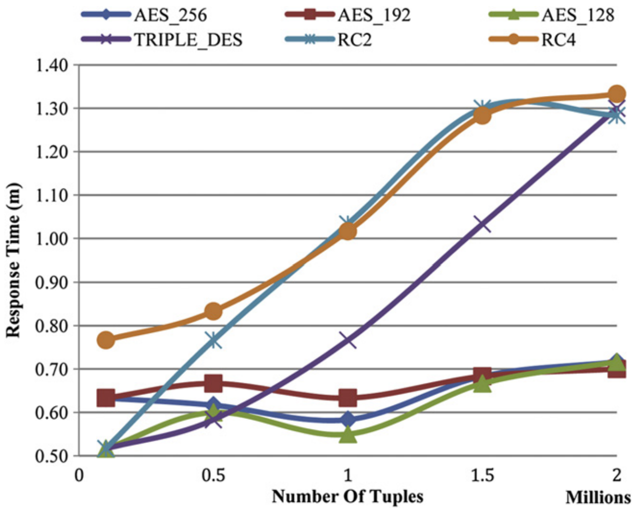


Fig. 1 – The impact of varying the number of tuples on the performance of encryption in multilevel database.

4.1. *Experimental database structure*

The Timesheet database is created and populated to facilitate our performance study. The Timesheet system that is used in the implementation is described as following:

The Employee relation provides information about Employees.

Employees(EMPID, Code, Name, Department, Type, Contract, Shift, Religion, Job, Position, Address, City).

The departure relation is used to store the departure notice of each employee when he leaves the site of the work.

Departure(EmpID, DepartureDate, ReturnDate, DepartureType)

The TimeSheet relation is used to store the timesheet of each employee every day.

TimeSheet(EMPID,Date, TimeSheet, OverTime, Remarks)

The Annual Rights relation is used to store the rights of each employee every year.

AnualRights(EMPID, Year, Description, InL, ADays, GDays)

Fig. 2 shows ER diagram for the timesheet system that is used in the implementation of the prototype to facilitate our performance study.

The experiments investigates the impact of varying the number of tuples, the number of attributes and the number of security levels on the performance of the multilevel database models and the proposed encryption-based MLS model. For each query, the monitoring tool records the time of the system to respond to the query. For each experiment, we plot the response times in a graph as a function of the variable that is being investigated.

4.2. *Select query*

The following experiments investigate the impact of varying the number of tuples, the number of attributes and the number of security levels on the performance of the multilevel database security models and the proposed encryption-based MLS model when executing the selection query.

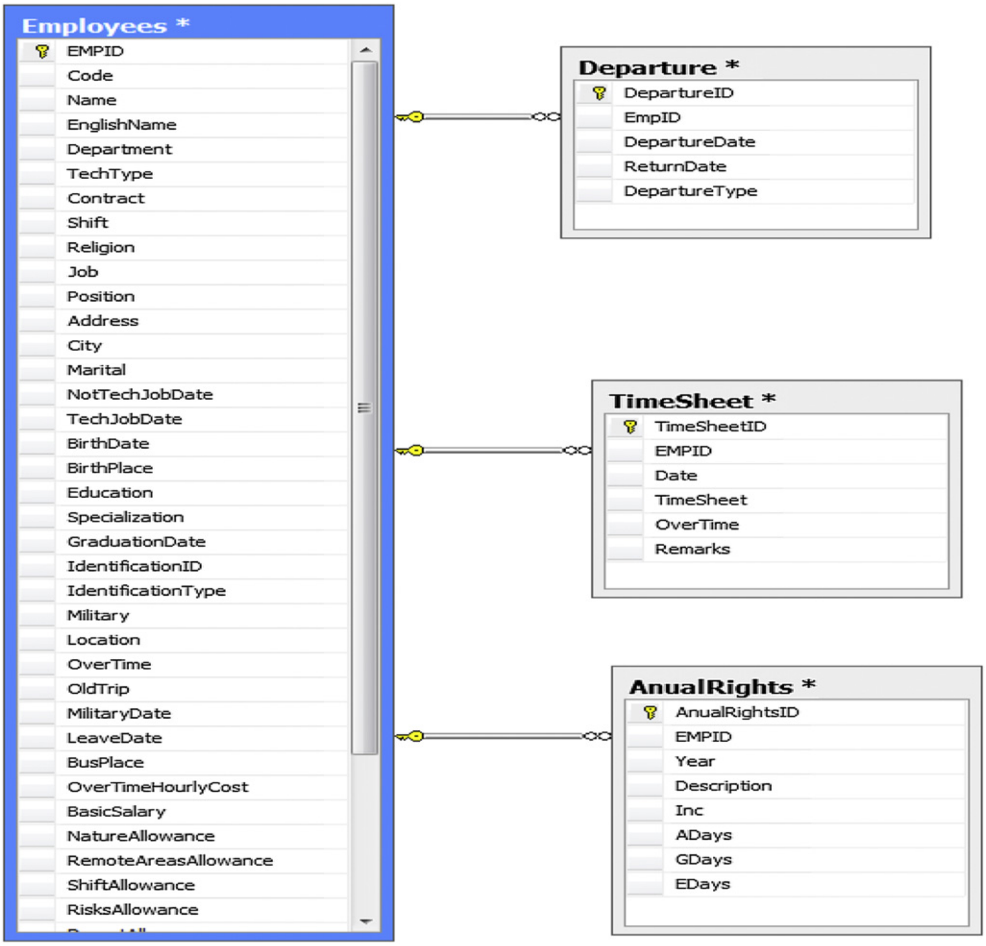
The where clause in the SELECT query will be taken into consideration when we evaluate the performance of the multilevel database security models and the proposed encryption-based MLS model when executing the selection query.

The SELECT statement that is used in the following experiments is described as follows:

Select * from Employee where department = Sales

4.2.1. *Impact of varying the number of tuples*

This experiment was designed to determine if the cost of processing varying numbers of tuples has an impact on the performance of the multilevel database models. We vary the number of tuples to 100,000, 500,000, 1,000,000, 1,500,000 and



Employees *
EMPID
Code
Name
EnglishName
Department
TechType
Contract
Shift
Religion
Job
Position
Address
City
Marital
NotTechJobDate
TechJobDate
BirthDate
BirthPlace
Education
Specialization
GraduationDate
IdentificationID
IdentificationType
Military
Location
OverTime
OldTrip
MilitaryDate
LeaveDate
BusPlace
OverTimeHourlyCost
BasicSalary
NatureAllowance
RemoteAreasAllowance
ShiftAllowance
RisksAllowance

Departure *
DepartureID
EmpID
DepartureDate
ReturnDate
DepartureType

TimeSheet *
TimeSheetID
EMPID
Date
TimeSheet
OverTime
Remarks

AnualRights *
AnualRightsID
EMPID
Year
Description
Inc
ADays
GDays
EDays

Fig. 2 – ER diagram for the timesheet system that used in the implementation.

2,000,000; fix the number of attributes at 3; fix the number of security levels at 4. From Fig. 3 the response times grow for all models as the number of tuples increases. Also supporting encryption in the proposed model reduces the database size because of removing the extra attributes which are used for the class levels.

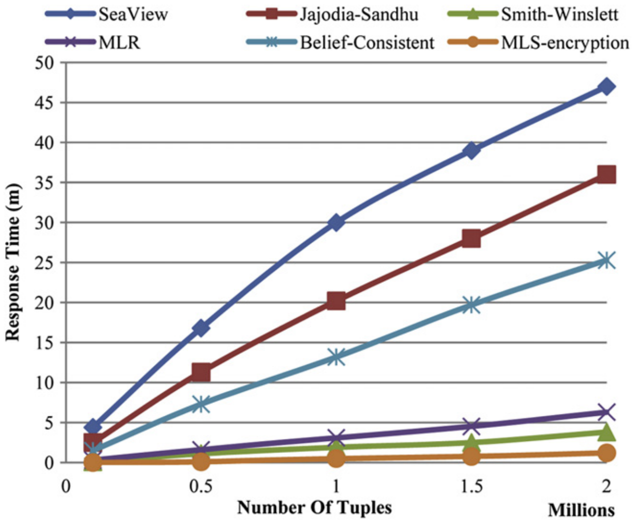


Fig. 3 – Impact of varying the number of tuples in selection query.

4.2.2. *Impact of varying the number of attributes*

This experiment was designed to determine if the cost of processing varying the number of attributes has an impact on the performance of the multilevel database models. We vary the number of attributes to 2, 3, 4, 5 and 6; fix the number of tuples at one million; fix the number of security levels at 4. From Fig. 4 the response times grow for all models as the number of attributes was increased. Also supporting encryption in the proposed model reduces the database size because of removing the extra attributes which are used for the class levels.

4.2.3. *Impact of varying the number of security levels*

This experiment was designed to determine if the cost of processing varying the number of security levels has an impact on the performance of the multilevel database models. We vary the number of security levels to 2, 3, 4, 5 and 6; fix the number of tuples at one million; fix the number of attributes at 4. From Fig. 5 the response times grow for all models as the number of security levels was increased. Also supporting encryption in the proposed model reduces the database size because of removing the extra attributes which are used for the class levels.

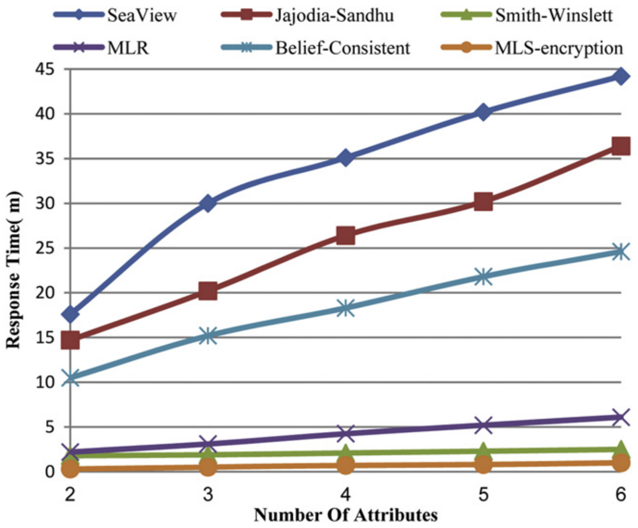


Fig. 4 – Impact of varying the number of attributes in selection query.

4.3. *Join query*

The following experiments investigate the impact of varying the number of tuples, the number of attributes and the number of security levels on the performance of MLS Models and the impact of supporting encryption in MLS database when executing join query. The where clause in the JOIN query will be taken into consideration when we evaluate the performance of the multilevel database security models and the proposed encryption-based MLS model when executing the JOIN query. The JOIN operation involves two tables the Employee table and the Departure table. The JOIN statement that is used in the following experiments is described as follows:

Select ∗ from Employee join Departure on Employee.Name = Departure.Name where Employee.department = Sales

4.3.1. *Impact of varying the number of tuples*

The number of tuples is varied to 100,000, 500,000, 1,000,000, 1,500,000 and 2,000,000; fix the number of attributes at 3; fix the number of security levels at 4. Fig. 6 illustrates the impact of varying the number of tuples in join query. From Fig. 6 the response times grow for all models as the number of tuples increases. Also supporting encryption in the proposed model reduces the database size because of removing the extra attributes which are used for the class levels.

4.3.2. *Impact of varying the number of attributes*

The number of attributes in each table is varied to 2, 3, 4, 5 and 6; fix the number of tuples at one million; fix the number of security levels at 4. Fig. 7 illustrates the impact of varying the number of attributes in join query. From Fig. 7 the response times grow for all models as the number of attributes was increased. Also supporting encryption in the proposed model reduces the database size because of removing the extra attributes which are used for the class levels.

4.3.3. *Impact of varying the number of security levels*

The number of security levels is varied to 2, 3, 4, 5 and 6; fix the number of tuples at one million; fix the number of attributes at 4. Fig. 8 illustrates the impact of varying the number of number of tuples in join query. From Fig. 8 the response times grow for all models as the number of the security levels was increased. Also supporting encryption in the proposed model reduces the database size because of removing the extra attributes which are used for the class levels.

4.4. *Update query*

The number of the updated tuples is varied to 100,000, 500,000, 750,000 and 1,000,000; fix the number of attributes at 3; fix the number of security levels at 4. Fig. 9 illustrates the impact of varying the number of tuples in update query. From Fig. 9 the response times grow for all models as the number of tuples increases. Also supporting encryption in the proposed model

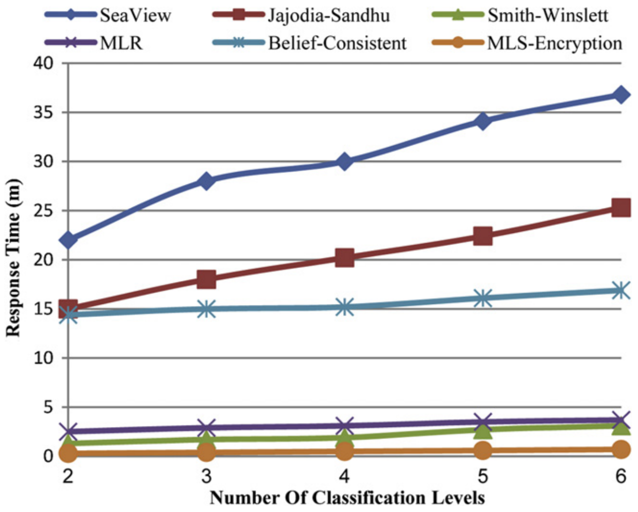


Fig. 5 – Impact of varying the number of security levels in selection query.

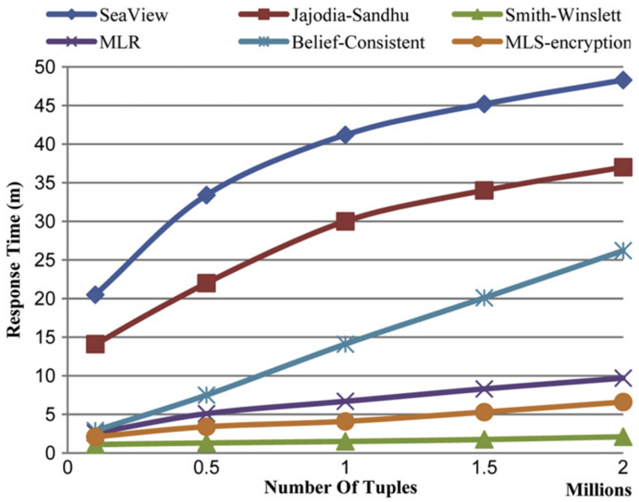


Fig. 6 – Impact of varying the number of tuples in join query.

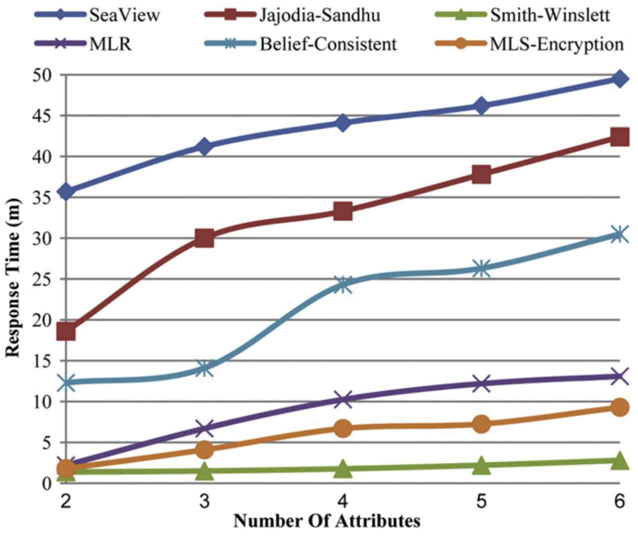


Fig. 7 – Impact of varying the number of attributes in join query.

decreases the performance of multilevel database because during the execution of the update statement, the encryption and the decryption mechanisms will be included together in the update procedure.

The where clause in the UPDATE query will be taken into consideration when we evaluate the performance of the multilevel database security models and the proposed encryption-based MLS model when executing the UPDATE query. The UPDATE statement that is used in the following experiments is described as follows:

Update Employee set salary = salary
+ 100 where department
= Sales

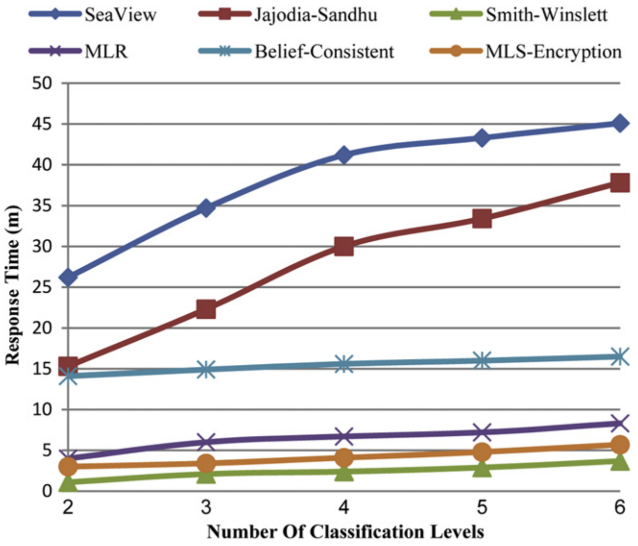


Fig. 8 – Impact of varying the number of security levels in join query.

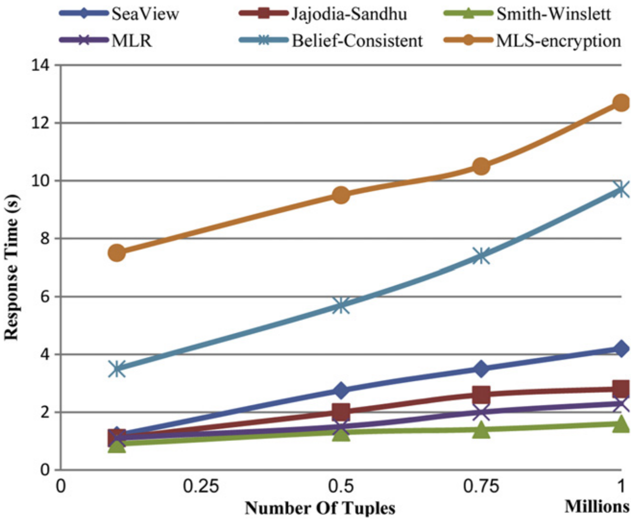


Fig. 9 – Impact of varying the number of tuples in update query.

5. Analysis of experimental results

Through a number of experiments, this paper compared the performance of all the multilevel database security models and the proposed encryption-based MLS model. We investigated the performance by varying the numbers of tuples, attributes and security levels using the SELECT and the JOIN queries.

The performance of the Smith–Winslett model is the best because it does not support classification at the level of each single attribute, the access classes can be assigned only to key attributes and to tuples as a whole. The MLR model has a performance less than the performance of the Smith–Winslett model because it supports classification at the level of each single attribute. The Belief-Consistent model has a performance less than the performance of the MLR model because it supports combination of classification levels for each single attribute to enable the user to assert his beliefs of lower-level user's information. The Jajodia–Sandhu model has bad performance because the impact of union operation between single level relations in the recovery algorithm. The SeaView model has a very bad performance because the impact of the join operation between vertical single level relations and the impact of the union operation between horizontal single level relations in the recovery algorithm.

From the experimental results in the previous section the proposed encryption-based multilevel database model, which is a combination of the MLR model and encryption algorithm, has a performance better than the performance of MLR model in retrieving data from the multilevel database. This improvement of the performance of the proposed model is due to the reduction of the multilevel database size, removing the attributes classification columns and encrypting the attributes by field-key according to its security level. Adding the encryption algorithm reduces the overhead in MLR model of checking the class level hierarchy for each data element when retrieving data. The proposed model uses the encryption keys that belong to the user to encrypt each data element when retrieving data.

The performance of the proposed encryption-based multilevel database model is less than the performance of the MLR model in updating data because the overhead of supporting the encryption algorithm in the update query is executed. In the proposed model the data is first decrypted, ensuring the condition of the update statement is met, executing the update statement and encrypting the data again.

6. Conclusion and future work

The major contribution of our work is the proposition of an encryption-based multilevel database model. The proposed model used an encryption algorithm as an additional layer of security over the MLR model in multilevel database security. A working multilevel secure database prototype was implemented in Microsoft SQL server R2 and to measure the performance experiments that were evaluated using the prototype. Also in this paper the impact of supporting encryption algorithm in the multilevel database security was measured and the cost performance was evaluated by varying the numbers of tuples, attributes and security levels using the SELECT, JOIN and UPDATE queries.

Supporting encryption in the multilevel database security improved the performance of retrieving data in the SELECT query and the JOIN query. This improvement in the performance is due to the reduction of the database size because the extra classification attributes are replaced by supporting the encryption algorithm in the multilevel database security. Also the multilevel database design had become easier because there was no change in the structure of the base table.

Although the proposed encryption-based multilevel database model improved the performance of retrieving data, it had a bad performance in updating the data. This bad performance in updating the data is due to the encryption and the decryption for the data during the execution of the UPDATE query.

In the future, the impact of supporting advanced DB-mechanisms like partitioning and indexes in the proposed encryption-based multilevel database model will be taken into consideration. Also the protection of the decrypted data when supporting advanced DB-mechanisms will be investigated in our future research.

REFERENCES

Bertino Elisa, Sandhu Ravi. Database security-concepts approaches, and challenges. IEEE Transaction on Dependable and Secure Computing 2005;2(1):2–19.

Cuppens Frederic, Gabillon Alban. Logical foundations of multilevel databases. Data & Knowledge Engineering 1999; 29(3):259–91.

Dave Pinal. Introduction to SQL server encryption and symmetric key encryption tutorial. Available, http://dotnetslackers.com/articles/sql/IntroductionToSQLServerEncryptionAndSymmetricKeyEncryptionTutorial.aspx; 2008.

Garuba Moses. Performance study of a cots distributed DBMS adapted for multilevel security. Ph.D. thesis, Department of Mathematics Royal Holloway, University of London, Egham, Surrey Tw20 0ex, England; 2003. Available: http://digirep.rhul.ac.uk/items/f076f347-2036-6bd0-98c8-e1d2dc9cf4ab/1/.

Garuba Moses, Appiah Edward, Burge Legand. Performance study of a MLS/DBMS implemented as a kernelized architecture. In: Proceedings of the international conference on information technology: coding and computing (ITCC'04); 2004. p. 566–70.

Imran Sohail, Hyder Irfan. Security issues in databases. In: Proceedings of the second international conference on future information technology and management engineering; 2009. p. 541–5.

Jukic Nenad A, Vrbsky Susan V. Asserting beliefs in MLS relational models. Proceedings of the SIGMOD Record 1997;26(3):30–5.

Jukic Nenad, Vrbsky Susan V, Parrish Allen, Dixon Brandon, Jukic Boris. A belief-consistent multilevel secure relational data model. Information Systems 1999;24(5):377–402.

Lee Sang-Won, Kim Yong-Han, Kim Hyoung-Joo. The semantics of an extended referential integrity for a multilevel secure relational data model. Data & Knowledge Engineering 2004; 48(1):129–52.

Pan Leon. Using criterion-based access control for multilevel database security. In: Proceedings of international symposium on electronic commerce and security; 2008. p. 518–22.

Pranjic Mario, Fertalj KreSimir, Jukic Nenad. Importance of semantics in MLS database models. In: Proceedings of the 24th international conference on information technology interfaces; 2002. p. 51–6.

Pranjic Mario, Jukic Nenad, Fertalj Krcsimir. Implementing belief-consistent multilevel secure relational data model: issues and solutions. In: Proceedings of the 25th international conference on information technology interfaces IT1; 2003. p. 149–54.

Rask Art, Rubin Don, Neumann Bill. Implementing row- and cell-level security in classified databases using SQL server 2005. Available, http://technet.microsoft.com/en-us/library/cc966395.aspx; 2005.

Rjaibi Walid, Bird Paul. A multi-purpose implementation of mandatory access control in relational database management systems. In: Proceedings of the 30th VLDB conference, Toronto, Canada; 2004. p. 1010–20.

Sandhu Ravi, Chen Fang. The multilevel relational (MLR) data model. ACM Transactions on Information and System Security 1998;1(1):93–132.

Zuo Xiao-Dong, Liu Feng-Mei, Ma Chao-Bin. A new approach to multilevel security based on trusted computing platform. In: Proceedings of the sixth international conference on machine learning and cybernetics, Hong Kong; 2007. p. 2158–63.

Prof. S. El-Rabaie (Senior Member, IEEE'1992-MIEE-Chartered Electrical Engineer) was born in Sires Elian (Menoufia), Egypt in 1953. He received the B.Sc. degree with Honors in Radio Communications from Tanta University, Egypt, 1976, the M.Sc. degree in Communication Systems from Menoufia University, Egypt, 1981, and the Ph.D. degree in Microwave Device Engineering from the Queen's University of Belfast, 1986. He was a Postdoctoral Fellow at Queen's (Dept. of Electronic Eng.) up to Feb. 89. In his doctoral research he constructed a CAD package used in nonlinear circuit simulations based on the harmonic balance techniques. Since then he has been involved in the development of GaAs FET doublers, triplers and oscillators from X to K band. He was invited in 1992 as a Research Fellow in the North Arizona University (College of Engineering and Technology) and in 1994 as a visiting Prof. in Ecole Polytechnique de Montreal (Quebec), Canada. Prof. El-Rabaie has authored and co-authored more than 90 papers and technical reports, fifteen books under the titles (Computer aided simulation and optimization of nonlinear active microwave circuits, The whole dictionary for the computer and the Internet terminologies, Basics and technologies of data communications in computer networks, Technologies and Internet programming, The distance learning and its technologies on the third millennium, computer principles and their applications in education, software engineering (1), Management of computer networks (1,2), Advanced Internet programming, data-base principles, building of compilers, software engineering (2), Ethics of profession). In 1993, he was awarded the Egyptian Academic Scientific Research Award (Salah Amer Award of Electronics) and in 1995, he received the award of the best researcher on (CAD) from Menoufia University. He has shared in translating the first part of the Arabic Encyclopedia. Now he is the Vice Dean of Postgraduate Studies and Research, Faculty of Electronic Engineering, Menoufia University. Address: Faculty of Electronic Engineering, 32952 Menouf, Egypt. E-mail: srabie1@yahoo.com, srabie1@hotmail.com. Mobile: 0184985170 – 0198699975.

List of published books in computer science and educational technology:

1) The whole dictionary for the computer and the Internet terminologies,
2) Basics and technologies of data communications in computer networks,
3) Technologies and Internet programming,
4) The distance learning and its technologies on the third millennium,
5) Computer principles and their applications in education,
6) Software engineering (1),
7) Management of computer networks (1),
8) Advanced Internet programming,
9) Management of computer networks (2),
10) Data-base principles,
11) Building of compilers,
12) Management of computer networks (2),
13) Ethics of profession,
14) Software engineering (2).
15) Computer aided design of nonlinear microwave circuits.

Osama S. Farag Allah was born in Menoufia, Egypt on August 29, 1974. He received B.S. in Computer Science & Engineering (1997) from Menoufia University, Faculty of Electronic Engineering, Egypt in 1997, M.Sc. in Computer Science & Engineering (2002) from Menoufia University, Faculty of Electronic Engineering, Egypt in 2002, and Ph.D. in Computer Science & Engineering (2007) from Menoufia University, Faculty of Electronic Engineering, Egypt in 2007. He was appointed as a demonstrator at the Department of Computer Science and Engineering, Faculty of Electronic Engineering, Menoufia University, from 1997 to 2002. He became an Assistant Lecturer in 2002 and promoted to a Lecturer in 2007. His research interests cover Computer networks, Network security, Cryptography, Internet security, Multimedia security, Image encryption, Watermarking, Steganography, Data hiding, Chaos theory.

Ahmed I. Sallam was born in AL Gharbia, Egypt on April 10, 1982. He received B.S. in Computer Science & Engineering (2005) from Al Azhar University, Faculty of Engineering. He became a senior Software Engineer in Qarun Petroleum Company in 2008.