ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

A Scheme for Encrypting a Database

S.Pothumani,
Assistant Professor, Bharath University, Chennai-600073, India

ABSTRACT*:* Database contains related informations. Providing security to database is very difficult. To avoid the risk posed by this threat, database encryption has been recommended. This paper provide a good method to encrypt a database with good security and limited cost.

KEYWORDS*:* database; database security; cryptography; cryptographic keys; encryption; decryption; access control

I. INTRODUCTION

In today’s economy databases symbolize one of the most valuable assets. They form the basis for e-business, e-commerce, Enterprise Resource Planning (ERP) and other sensitive activities. Many organizations cannot work properly if their database is down; they are normally referred to as mission-critical system. Along with the wide application of database comes the need for its protection. Universally, huge amount of effort, time and resources are been spent in trying to make database systems meet security requirements. These security requirements normally include:

i. Prevention of unauthorized disclosure of information

ii. Prevention of unauthorized modification of information

iii. Prevent denial of service

iv. Prevent system penetration by unauthorized person

v. Prevent the abuse of special privileges

cryptographic technique can ensure excellent security for databases, by reducing the whole security process down to the protection of only few cryptographic keys. However, time cost involved in encrypting and decrypting data items can greatly degrade the performance of a database system. A compromise solution between performance and security can be achieved by only encrypting the sensitive data in a database.The objective of this paper is to propose a secure database encryptions scheme that provides maximum security, whilst limiting the added time cost for encryption and decryption. The encryption technique considered is Data Encryption Standard (DES), but the scheme is also applicable to other cryptographic techniques and standards.

Dorothy E. Denning [5, 7] has contributed immensely to efforts towards providing Database Security through Cryptographic means. In [5] he provides solutions to the security problems of field based protection, and presents a comparative study on implementing encryption at various database levels i.e table, attribute and field (element) levels. And in [7] he tries to solve database integrity problem using cryptographic checksum. Downs, D. and Popek, G. J, [12] proposed a system model using two trusted modules (security kernels), and use tags to ensure integrity. Davida, G.I, Wells, D.L and Kam, J.B, [11] proposed a blend record and field techniques based on remainder theorem. The above approaches are all different from ours in terms of structure, key management and implementation procedures.

The rest of the paper is organized as follows: Section 2 describes the model of the scheme. Section 3 its implementation. Section 4 the management of the cryptographic keys. Section 5 the encryption and decryption procedure. And section 6 concludes the paper

II. MODEL

Our proposed scheme adopts a two-level relational database system, wherein subjects (users) are assigned to either of two levels, L1 (low) and L2 (high). All Subjects have access right to their own personal private data (P). And in addition, subjects in L1 have access right only to unclassified (U) public data, whilst those in L2 have access right to both unclassified and classified (C) public data. The access rights of subjects in L2 to classified data is however limited to their “Need-to-know” sensitive data. The elements used in our scheme are as ***Unclassified data*****:** are non-sensitive data that forms the bulk of the database and are open to all users for access.

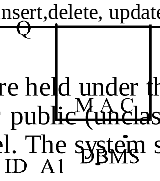
Classified data: are sensitive data that have restricted access. Example salary of employees is considered a sensitive data not to be disclosed to other subjects. But for some subjects such as account manager who has need-to-know salaries of all employees, access privilege to employees’ salary should be granted to them.

Private data*:* are user’s personal secret data such as credit card number which should be available only to them and for which others need to take direct permission from them before being accessed.

TABLE I. ELEMENTS OF THE MODEL

Set	Elements	Semantics
S	$\{s_1, s_2, s_3, \ldots\ldots\ldots, s_n\}$	Subject (Users)
O	$\{A_j, X_{ij}\}$ i = row ; j = column	Database object: {Attribute, data}
L(s)	$\{L_1, L_2\}$ $L_1 < L_2$	Clearance level of subjects {low, High}
L(o)	{[U, C], P} U < C < P	Classification of objects {[unclassified,classified], private}
K	$\{K_1, K_2, \ldots\ldots, K_r\}$	Set of special access privileges to sensitive object
V	σ,π,×, insert, del, update	Access operations: select,project,join,insert,delete, update

As database is a collection of related data, it is assumed that classified data are held under the same database attributes (column) different from those for unclassified data. Therefore classification for public (unclassified, classified) object is done at attribute level whilst that for private objects is done at data element level. The system structure of the model is as shown in figure 1.



Basically the model is divided into three layers: The first layer is the user interface layer which contains two blocks, one for level1 subjects and the other for level2 subjects. All subjects posses a unique key K_P in the form of a certificate that they use when accessing their encrypted private data in the database. The second layer is the database management layer which also contains two main blocks, one that implements the mandatory access control (MAC) to the database, and another that houses a tamper-free controller closely linked with a trusted subject (TS).

The functions of the controller (KC) are:

1. To generate and safely store two sets of encryption keys. K_P for each user’s private encrypted data and K_j for encrypted classified data.
2. To encrypt sensitive data before being storage in the database
3. To decrypt cipher text in the response to users queries that satisfied security requirements.4. To perform integrity check on user returned data

5. To facilitate authentication of user’s when necessary. The Trusted-Subject (TS) is in charge of:

1. Registering new subject , and their records
2. Deleting subjects and objects
3. Declassifying subjects and objects

4. Updating classified and private data.
5. Assign subjects access privileges to sensitive data.

The bottom layer contains the database. In order to facilitate the fast retrieval of data, the database system stores unclassified data in the clear whilst classified and private data are stored in encrypted form. The first field of every record uniquely identifies the record, and serves as its primary key (ID). Primary key should not be confused with cryptographic keys used to decrypt cipher text

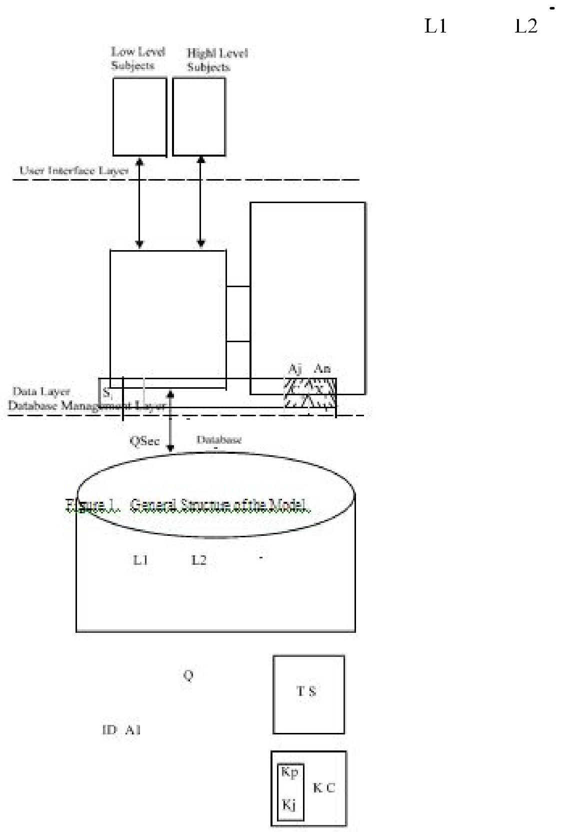


Figure 1. General Structure of the Model.

This section describes how the model operates to provide security to the database. As mentioned earlier classified and private data are stored in encrypted form in the database. If intruders manage to penetrate into the database, sensitive data will still be concealed from them, as they would lack the necessary decryption keys. This provides confidentiality to sensitive data stored in the database.

To control user's access to information in the database, the MAC maintains an authorized view of the database for all subjects. With the labeling of public (unclassified and classified) objects at the attribute level, the maximum authorized view of a subject to a relation R is defined as follows:

Let L (A_i) denotes the classification of attribute A_i in the relation R. The set of attributes authorized to subject 's' with clearance L(s) is defined by:

$$A.auth(s) = \{A_i \in R \mid L(A_i) \leq L(s)\}$$

Thus the maximal authorize view of 's' on R is

$$s.MAX\_VIEW(R) = \pi_{A.auth(s)}(R).$$

This implies that any query Q_i from a subject has to be augmented to a secure form Q_{sec} that reflects the authorized view of the subject. As an example let us consider three classes of increasingly complex queries: select, select-project, and select-project-join.

Select Query

For a query Q_1 = SELECT all (records) from R (relation) WHERE (constrained by) F.

$$Q_1 = \sigma_F(R).$$

The secure form $Q_{\sec .1} = \sigma_F(\pi_{A.auth(s)}(R)).$

Select-Project Query

For a query Q_2 = SELECT all records from R WHERE F, and PROJECT on attribute set A.

$$Q_2 = \pi_A(\sigma_F(R)).$$

The secure form $Q_{\sec .2} = \pi_A(\sigma_F(\pi_{A.auth(s)}(R))).$

Select-Project-Join Query

For a query Q_3 that joins two sub queries Q_a and $Q_{b,}$ In general, for all users query to the database, MAC takes as input [2]

1. User's ID
2. User's clearance L(s)
3. User's query Q_i
4. D, a set of constraints that prevent users from making direct modifications to data and classes of subjects and objects.

If no security violation is detected, MAC provides an answer in responds to Q_i.

To maintain the integrity of the database, subjects are constrained from carrying out direct access operations that may change the state of the database. Such operations include Create, Update, Delete, and change of Classification. Request for such operations must be forwarded to the trusted subject who evaluates them with respect to security violations and if safe allows their execution. Further more the enciphered text also serves as checksum against which returned sensitive data are crosschecked. The restriction on implemented using the following two constraints

Constraint 1: For all $s \in S$,

If $s \neq TS$

S* = S L(s)* = L(s)

Constraint 2: For all $o \in O$,

If $s \neq TS$

O* = O

L (o)* = L (o) Content (o)* = Content (o)

The Star * in front of variables refer to their new state whilst unstared variables their old state. Constraint1 removes the right from all subjects to create other subjects or change their clearance level. And constraint2 removes the right from subject to directly create new object or change the classification or content of objects.management of keys is vital to the overall security of the database system. Key management includes every aspect of the handling of keys from their generation to their eventual destruction [3]. The proposed scheme makes use of three sets of encryption keys K_j, K_p and K for classified data, private data and controller's master key respectively. Their generation and distribution process are discussed below:

A. Key Generation

An ideal method of key generation would be one that chose the key at random. Unfortunately, absolute randomness to key generation is difficult to achieve. A possible source of generating random key values is through pseudo-random key generators with different seed startup values. This forms the bases for generating the controller master key K. The Generation of the other two keys is as follows:

- *Generation of classified data keys*: With classified data labeled at attribute level. Let A_j be an attribute identifier, and K the controller master key. An attribute key to classified data is defined by: $K_j = g\ (A_j, K)$ or $K_j = E_K(A_j)$, where g is the key generating function.
- *Generation of private data keys:* Users' private data elements should have unique key. Therefore their classification was done at data element level. It is assumed that the first field (ID) in every record uniquely identifies the record; i.e, it is the primary key of the database. For every private data X_{ij}, the private key is defined by $K_p = g\ (ID, A_j, T, K)$ or

  $$K_p = E_K(K_t) = E_K(K_i \oplus T) = E_K[(ID \oplus A_j) \oplus T].$$

  ID

  Where T is time stamp included to ensure that K_p is unique every time it is updated and $\oplus$ is the exclusive-or operator.

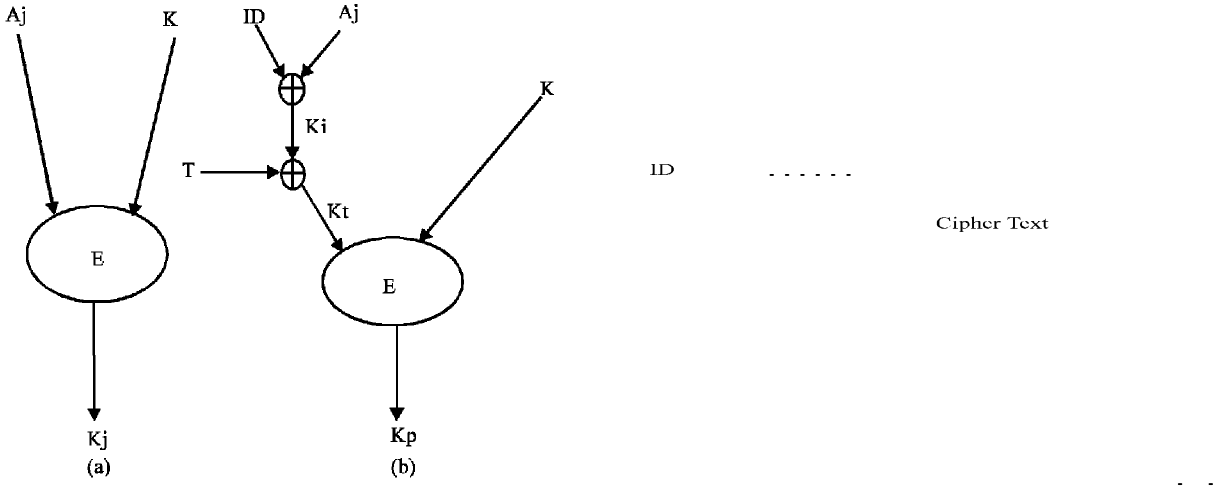


Figure2. a and b present and hierarchy structure for generating classified and private keys respectively.

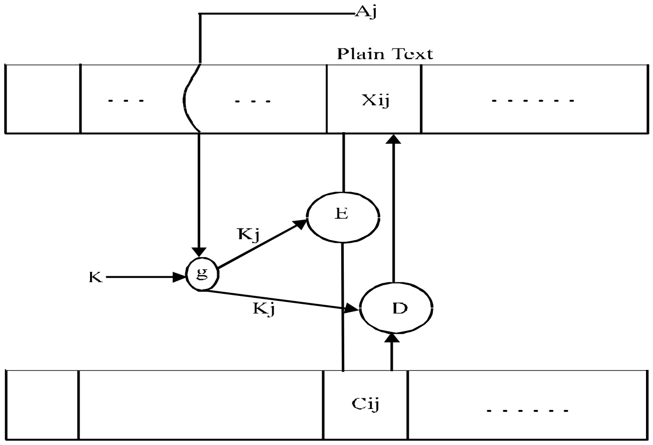
Using Data Encryption Standard (DES) technique, each identifier is assumed to be of at most 8 byte long. Padding is applied where necessary to fill the 8 bytes. Also the key generating function is chosen such that the probability of getting repeated keys is low and also computationally infeasible to determine one element key from other element keys [4].

B. Key Distribution

The generated classified and private data keys are stored in a tamper free controller. A copy of private data key K_p is sent to the owner of the private data as a form of certificate. And request for private data should be accompanied with its certificate for such request to be honored. Subjects can exchange their certificate with others whom they wish to allow view to their private data. However only the private data owners can request update to private data or it certificate. Thus the controller must authenticate the originator for update to private data. In general private data keys are refreshed more often than classified data keys as they have greater chances of being exposed.

V. Encryption and Decryption

The Classified and private data elements X_{ij} are encrypted/decrypted using Data Encryption Standard (DES) technique. Where X_{ij} is less than the 8-byte block size of DES, X_{ij} is replicated as many times as necessary to fill the block. If X_{ij} however exceeds the block size, then the encryption is performed using cipher block chaining with initialization block.



classified sensitive data are stored in encrypted form, their decryption process is very fast, as only one key is needed to decrypt a whole column of encrypted classified data. Also, although accessed encrypted private data has to be decrypted separately using their unique keys, requests for private data are very seldom, being carried out only once on a while. This makes the time cost for their encryption and decryption to have less significance on the overall performance of the scheme.

Finally, because all the keys are held in a tamper free controller, with only certificate issued to owners of private data, refreshment of cryptographic keys will be required less frequently.

The only downside side of our scheme is that queries such as sums, averages, counts and other statistical functions that aggregate over a range of data in the database cannot be performed directly. However users themselves at the user interface layer can perform such task.

Checksum

$$S_{ij} = C_{ij} = E^{j}(X_{ij})$$

for classified data, and

VI. SUMMARY AND CONCLUSION

This paper investigates the role cryptograph can play in database security. In database systems, sensitive data stored in the clear are vulnerable to attack. No matter the amount of security measures taken, there would always be some security leaks which attackers can use to penetrate the database. However, if sensitive data are encrypted before storage in the database, risks from security leaks can be eliminated. And the whole database security issue will reduced down to the protection of few cryptographic keys.

Our proposed database encryption scheme is considered efficient because it provides maximum security to the database whilst the added time cost for encryption and decryption is very minimal. All aspects of security concerned from confidentiality, access control, integrity, authentication to non-repudiation were addressed.

1. Confidentiality –The encryption of sensitive data provides confidentiality.

2. Access Control – Maintaining an authorize view of subjects, controls access to the database

3. Integrity – Constraining subjects from performing operations that change the state of the database, maintains the integrity of database items.

4. Authentication and Non-repudiation - the use of certificate by subjects to access private data provides authentication and non-repudiation.

Furthermore, the use of authorized view and a tamper free controller solves indirect threats from inference channel, insecure information flow and ciphertext searching.

To reduce the time spent on encryption and decryption, the scheme divides the database into sensitive and non-sensitive data. Non-sensitive data which forms the bulk of the database are stored in the clear facilitating their fast retrieval.

REFERENCES

[1] D.W Davis, and W.L Price, " Security for Computer Networks." A wiley-interscience publication. 1984

[2] William Stallings, "Cryptography and Network Security Principles and Practice" 2nd ed. Prentice-Hill Inc. 1999.

[3] D. E Denning, "Field Encryption and Authentication." Proc. Of CRYPTO 8.9, Plenum Press. 1983.

[4] H.M Carl, M.M Stephen, "Cryptography: A new dimension in computer data security." A wiley-interscience publication. 1982.

[5] D. E Denning, "Cryptographic checksums for multilevel data security. "Proc. of Symp.on Security and Privacy, IEEE Computer Society, 1984.pp. 52-61.

[6] D.E.Bell, and LaPadula, "Secure Computer Systems: Unified Exposition and iWtics Interpretation," Report ESD-TR-75-306, MITRE Corporation, Bedford, Mass. 1976.

[7] R.S Sandhu, "Lattice-Based Access Control Models." Computer, 26:11,1993. pp 9-19

[8] R Sandhu., V Bhamidipati., and Q Munawer. "The ARBAC97 Model for Role-Based Administration of Roles." ACM Trans. on Info. and System Security, 1999.vol 2:1, pp 105-135.

[9] G.I Davida, D.L Wells, and J.B Kam, "A Database Encryption System with sub keys." ACM Trans. On Database Systems 1981 vol 6:2.

[10] D Downs,. and G. J Popek, "A Kernel Design for a Secure Data Base Management System." *Proc.*3rd *Conf.* Very Large *Data Bases*, IEEE and ACM, New York, 1977 pp. 507-514.