Enforcing Database Security using Encryption and Secure Database Catalog

Surya Pratap Singh
Department of Computer Science
DDU Gorakhpur University
Gorakhpur (U.P.) – 273009

Akhilesh Kumar Mishra
Department of Computer science
Suyash Institute of technology
Gorakhpur(U. P.) -273016

Arvind Kumar Maurya
Department of Computer Science
DDU Gorakhpur University
Gorakhpur (U.P.) – 273009

Upendra Nath Tripathi
Department of Computer Science
DDU Gorakhpur University
Gorakhpur (U.P.) – 273009

Abstract: **In this modern environment the use of internet and use of computers are growing with a very fast rate and so as the use of database is also increasing at the same pace. The need of database security is also very important in this environment because at every online activities database is required and if for any reason the database security is compromised then this will become a major threat to sensitive and confidential information. Various database researchers proposed methods to protect the database form these vulnerabilities but none of the approach is fully able to protect the database form all kinds of security threats. One of the most popular methods to ensure the database security is through cryptography. In this paper we propose the use of Secure Database Catalog to store the encrypted data of the users in database to protect the information in database.**

Keywords: Database Security, Cryptography, threats, Secure Database Catalog

1. INTRODUCTION

Using of internet has become part of the daily life of any people. Online shopping has become very popular and e-business is the new face to do the business.[1]

In this time, when using all the old and new services, people are worried about the privacy and integrity of their business data. Recently, there was an attack on a popular web site which resulted in the possible stealing of the credit card numbers of several thousand customers.[5] This shows that if the credit card details or other personal information is somehow stolen can be very harmful for the customers of online sites. So the security in that environment is very important.[10][13]

A Relational Database Management System (RDBMS) plays a critical role in the new business. [14]Huge amount of business data are gathered, stored and processed, which demands more powerful support from the backend database (DB) server. If we look at a purchase activity of a web customer and trace the data, we can see that there are two security issues to be addressed –

1.1. Secure transmission of data: When a customer submits her/his personal and confidential information (e.g., credit card number) through her/his web browser, the information should remain confidential while transmitting form users web browser to the web server, the application server, and the backend DB server.[3][4]

1.2. Secure storage and access of data: When the personal and confidential data of customer arrive at the Database server, the data should be stored in such a way that the access to these data should be provided only to authorize people with appropriate authorization process [8]. The secure transmission of data is well studied and well supported in today's e-business market. All web browsers and web servers support SSL (Secure Socket Layer) and/or TLS (Transport Layer Security). A credit card number is well protected during transmission from a web browser to a web server via an SSL connection. However, once the data arrive at the database server, there is no sufficient support in storing and processing them in a secure way.

For example - an RDBMS might not even provide an encryption mechanism to securely store the credit card numbers. Although the general problem of secure data storage is well studied, the importance of secure data storage in an RDBMS has not been fully understood, and necessary steps to encrypt Database data haven’t been followed.[10]

We thought that it is very important to integrate cryptographic support into RDBMS. Cryptographic support is adds an important dimension in the database security and enhance the security policies. It is complementary to access control and cryptography and access control both can increase the security of confidential data stored in the database. When we thought about integrating cryptographic support into an RDBMS, there are three general approaches.[15]

1.3. Loose coupling: A third party crypto service can be consulted by a database server and there are only minor changes on the server side. For example, a set of stored procedures can be pre-installed in the server. Each stored procedure provides a special crypto service to the database users by calling the crypto primitives supplied by the third party package. One example is an encryption PL/SQL package that encrypts a table column with a user supplied encryption key.

1.4. Tight coupling: A complete set of basic crypto primitives are built into the database server as a set of new SQL statements, together with the necessary control and execution context to ensure that those new SQL statements can be executed securely. This approach is a much harder task than the previous one in terms of implementation, but it is preferable in the long run. The reason is simple: loose coupling is likely to open many security holes.

1.5. Mixture of Loose and Tight coupling: To accommodate the urgent need for security enhancement, only a small subset of crypto primitives are integrated into the database server, based on which other services can be built using other database utilities such as user defined functions and stored procedures.

2. REVIEW OF LITERATURE:

D. E. Denning at, al [1] describes the importance of cryptography to secure the data access in web based environment, they propose the use of secret key encryption to be applied to database tables to encrypt the information contained in the table.

Agrawal at, al [2] explained it is not possible to accurately estimate original values in individual data record in RDBMS they propose a novel reconstruction procedure to accurately estimate the original data values.

Hacigumus, H.[3] explore a novel paradigm for data management in which a third party service provider hosts "database as a service", providing its customers with seamless mechanisms to create, store, and access their databases at the host site. They identify data privacy as a particularly vital problem and propose alternative solutions based on data encryption

Kadhem, H [4] propose mixed cryptography database (MCDB), a novel framework to encrypt databases over un-trusted networks in a mixed form using many keys owned by different parties. The explain encryption process is based on a new data classification according to the data owner. The proposed framework is very useful in strengthening the protection of sensitive data even if the database server is attacked at multiple points from the inside or outside.

Chin-Chen Chang [5] presented two new database encryption systems. Both systems are based on the concept of the RSA (Rivest, Shamir, Adleman) master key. The first proposed system is a field-oriented encryption system with user master keys that correspond to the access rights of multiple fields. The second proposed system is a record-oriented encryption system with a user master key. Using the idea of master keys, we present a method that establishes a correspondence between the subsets of a given set and a set of integers.

3. PROBLEM IN THE EXISTING APPROACH

Although access control has been used as a security mechanism almost since the starting time of the database systems, for a long time security of a Database was considered an additional problem to be addressed when the need arise and after threats to the secrecy and integrity of data had occurred . Various database companies are adopting the loose coupling approach and adding optional security support to their products. The approach of adding security support as an optional feature is not such as to fully secures the database, instead it decreases the performance of the database and at the same time it can open new security holes in the Database.

Database security is a wide research area and includes topics such as statistical database security, intrusion detection, and most recently privacy preserving data mining. In this paper we describe database encryption and how it can be implemented to secure the database through the use of Secure Database Catalog.

3.1. Encryption: In this time RDBMS provides limited support for data encryption. Data are stored in tables in the form they are loaded, mostly in their plaintext form, which is not acceptable according to the requirement of high level of protection and privacy.

Example: Let us suppose there is a database table Customer created by the following SQL statement:

```
CREATE TABLE Customer
     (Cid  integer PRIMARY KEY
     Fname varchar(25),
     Lname  varchar(25),
     ccnum char(16));
```

Where the column ccnum records customers' credit card numbers.

The following statement creates a new record for customer Surya Pratap whose credit card number is 1212232345456767

```
INSERT INTO Customer VALUES
     (123, “Surya”, “Pratap”,”1212232345456767”);
```

The credit card number will be stored in its original plain-text form. The owner of table Customer or anyone with appropriate privilege can read this number with a simple SELECT statement. For example, let us consider user Avinash is the owner of table Customer. Note that Avinash might be a pseudo user whose account is created by the DBA to keep track the smooth functioning of the Database operations.

- If the table Customer is stored in operating system files, the whole table can be protected by using the file permission mechanism of the operating system. Only users with the correct permissions can access the file and thus the table. The main problem with this approach is that file protection restricts the access to the whole table, not just a particular column, and thus would make the database hard to use.
- Another problem is that database security is tied up with the underlying operating system which itself is rather vulnerable to both insider and outsider attacks.
- Another more serious problem with the loose coupling approach is that a database is used only as a passive data repository. When encrypted data need to be processed, they need to be fetched out of the database by another application server first. The database engine itself can do nothing useful about them. Example, indexing on an encrypted column is impossible, and no useful database operation can be performed against encrypted data.

4. PROPOSED METHODOLOGY

To overcome form the problem we defined earlier we propose following solutions, by the use of which the database security can be enhanced.

4.1. Secure Database Catalog: A traditional data dictionary stores the information that is required to ménage every database objects in the Database. A data dictionary may contain many table views and catalog. It is important that users (as well as DBA) do not change the contents of any catalog manually. But those catalogs will be maintained by the Database server and can be updated or changed by the execution of system commands. However, a DBA can still make changes in a catalog table if she/he wants to do so.

To prevent the Database from unauthorized access and other security related problems we introduce the concept of Secure Database Catalog.

A Secure Database Catalog is just like a traditional system catalog but we provide two more security properties:

(a) The updation of the Catalog is protected by manual updating by users (including DBA)

(b) The access of the Catalog is protected by a very strict authentication process like biometric or multilevel password protection.

The Secure Database Catalog now stores all the information about the database objects such as contents of each and every table view etc, as well as when the records are updated in the database by any transaction it stores the updated record with the updation credentials, it also maintains the copy of previous state of the database which is consulted to recover in case of any failure. The contents of this Secured Database Catalog can only be assessed by strict access control policies and can't been updated normally. In this manner with the use of Secure database Catalog the data in the database being protected from various kind of vulnerabilities.

4.2. Security Mechanism based on Password Encryption: All Relational DBMS's available can authenticate a user through a password authentication mechanism. In this mechanism when a user types in the correct user name with appropriate password, then the access to the Database server is provided and the user can start using any of the database resource with the privilege mechanism.

The server can always keep a copy of the user password in memory during the session when a user is connected. This password will be used to do any necessary encryption for this particular user.

Example, suppose Surya's password is "Tiger".

When Surya logs in, the database server will get a memory copy of this password. When Surya creates a new customer record by issuing the following query

INSERT INTO Customer VALUES
(111, "Ravi", "Kant singh", "1234567898764321");

The database server will first encrypt the credit card number with key "Tiger" and then store the cipher text in the table Customer.

Later, when Surya wants to get Ravi kant Singh's credit card number, she runs the command

SELECT ccnum FROM Customer
WHERE userid = 111;

Since Surya must have logged in to run the SELECT statement, the server must have already obtained a copy of Surya's password "Tiger". The server will first decrypt the content of column Customer(ccnum) with "Tiger" and then return the original card number Dec("Tiger", Customer(ccnum)).

We propose another approach which uses a variation of the user password. When a user logs in by providing his/her password is used as a key to generate a working key that is used in all encryption operations. The main advantage of this approach is that now the user's password is not used directly in the possibly frequent encryption operations. Now for each column, a combination of the table name, column name and the password can serve as the key for working key generation and thus different columns are protected with different keys. additionally, we can use a unique row identifier into the working key generation process so that identical value appeared in the same column but different rows will be encrypted to different cipher texts.

Example, a working key for Customer(ccnum) can be generated by

Key = E("Tiger";"Customer"; userid; "ccnum").

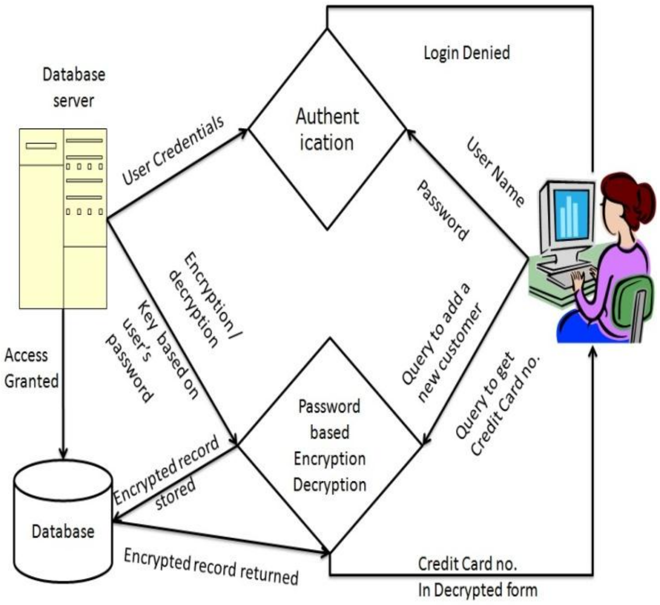
Login Denied
Database server
Authent ication
User Credentials
User Name
Password
Encryption / decryption
Key based on user's password
Query to add a new customer
Query to get Credit Card no.
Access Granted
Password based Encryption Decryption
Encrypted record stored
Database
Encrypted record returned
Credit Card no. In Decrypted form

Fig 1 Architecture of Password Based Encryption

The password based approach is most secure way because once the encrypted version of a credit card number is stored in the table Customer, only a user with the correct password (i.e., Surya) can decrypt and thus access the original card number.

5. CONCLUSION

The need of the database security is very important in the modern environment. Various researchers proposed different methodology to prevent the database from different types of security problems but none of the approach is fully able to secure the database, in this paper we analyses the security features of the current RDBMS's and finds out some of the important security problems. We introduce the key concept of Secure Database Catalog and proposed a new approach of password based encryption approach which is able to protect the confidential data of RDBMS.

REFERENCES

[1] D. E. Denning. Cryptography and Data Security. Addison-Wesley Publishing Company, Inc., 1982.

[2] R. Agrawal and R. Spirant. Privacy Preserving data mining. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, Dallas, Texas, 2000.

[3] Hacigumus, H." Providing database as a Service" Proceedings. 18th International Conference on Data Engineering, 2002.

[4] Kadhem, H. Amagasa, T. ; Kitagawa, H. "A Novel Framework for Database Security Based on Mixed Cryptography" Fourth International Conference on Internet and Web Applications and Services, 2009. ICIW '09.

[5] Chin-Chen Chang, Chao-Wen Chan "A database record encryption scheme using the RSA public key cryptosystem and its master keys" 2003 International Conference on Computer Networks and Mobile Computing, 2003.

[6] A. Brodsky., C Farkas, and S Jujodia., "Secure Databases: Constrains, Inferences, Channels, and Monitoring Disclosures." IEEE Transactions on Knowledge and Data Engineering 2000. vol. 12:6

[7] 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.

[8] D. Buehrer and C. Chang , "A cryptographic mechanism for sharing databases" , Proceedings of International Conference on Information & Systems , pp.1039 -1045

[9] L. Bouganim and P. Pucheral. Chip-secured data access: confidential data on untrusted servers. In VLDB '02: Proceedings of the 28th international conference on Very Large Data Bases, pages 131-142. VLDB Endowment, 2002.

[10] E. Damiani, S. D. C. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Balancing confidentiality and efficiency in untrusted relational DBMSs. In CCS '03: Proceedings of the 10th ACM conference on Computer and communications security, pages 93-102, New York, NY, USA, 2003. ACM.

[11] S. B. Guthery and T. M. Jurgensen , Smart Card Developer's Kit , 1998 :Macmillan Technical Publishing

[12] T. Dierks and E. Rescorla. The TLS protocol version 1.2, 2006.

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

[14] W. K. Wong, D. W. Cheung, E. Hung, B. Kao, and N. Mamoulis. Security in outsourcing of association rule mining. In VLDB '07: Proceedings of the 33rd international conference on Very large data bases, pages 111-122. VLDB Endowment, 2007.

[15] T. Dierks and C. Allen , The TLS Protocol - Version 1.0, Internet-Draft , 1997

Authors Profile

Surya Pratap Singh is MCA and UGC-NET qualified and pursuing Ph.D. In the department of Computer Science DDU Gorakhpur University, Gorakhpur (U.P. India) under the supervision of Dr. U.N. Tripathi. The area of research interest is Database Security, Networking. Mr. Surya Pratap Singh has published 15 papers in different national and international conferences/ Journals.

Akhilesh Kumar Mishra is M.TECH scholar from Suyash Institute of Technology affiliated to UPTU; He has done BTECH from Sachdev Institute of Technology. The area of research interest is Distributed Database Security. Mr. Akhilesh Kumar Mishra has published 2 papers in different national and international conferences/Journals.

Arvind Kumar Maurya is MCA and UGC-NET qualified and pursuing Ph.D. In the department of Computer Science DDU Gorakhpur University, Gorakhpur (U.P. India) under the supervision of Dr. U.N. Tripathi. The area of research interest is Distributed Database Security, Networking. Arvind Kumar Maurya has published 2 papers in different national and international conferences/Journals

Dr. Upendra Nath Tripathi is Assistant professor in Department of computer science DDU Gorakhpur University, Gorakhpur (U.P. India). He has 13 years of teaching and research experience. He has published 46 papers in various National and International Journals/conferences. His area of research interest is database systems, networking.