A Description of several levels of database encryption and Encryption Algorithms

Shivnandan Singh, Pratibha Singh, Niranjan Kumar Agri

***Abstract*: Encryption is the best mechanism to provide security because today security is biggest measure for any system without security we cannot say that a given system is secure or not.**
Encryption in database is very important topic in these days because we want to perform query operation on database allow optimized encryption and decryption of data.

INTRODUCTION

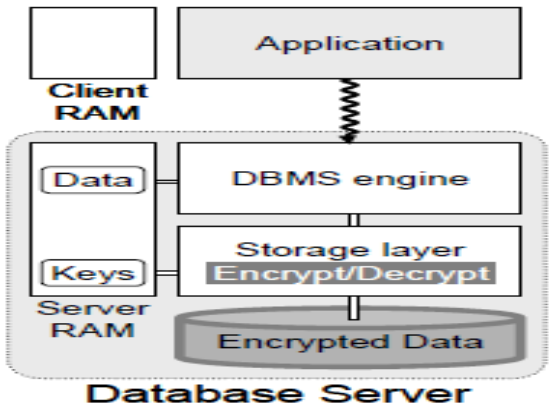
In this era of cyber world we are transferring a huge amount of data. It's our highest priority and biggest challenge is to protect our data. Data security has become a necessity for every individual who is connected to internet and use the internet for any purpose. Database security has been provided by operating system and physical security. Cryptographic support is another dimension of database security. Database encryption is providing the following facilities.

1) Encryption technique can prevent users from obtaining data in an unauthorized manner.
2) Encryption mechanism can verify authentic origin of a data item
3) Encryption mechanism also prevents from leaking information in a database when storage medium are lost.

Levels of Database Encryption

(a) Storage-level encryption

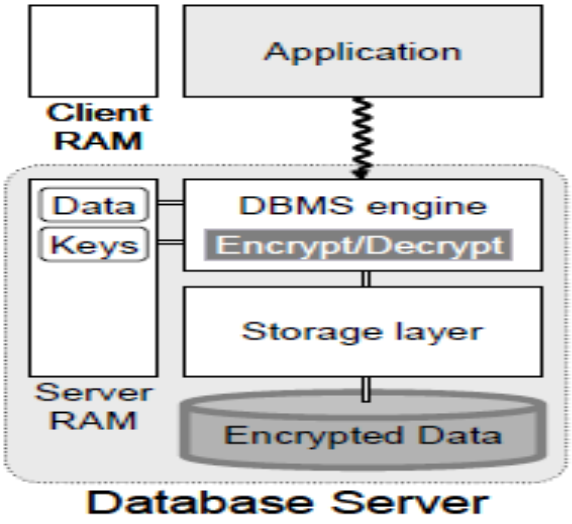
Storage-level encryption amounts to encrypt data in the storage subsystem and thus protects the data at rest (e.g., from storage media theft). It is well suited for encrypting files or entire directories in an operating system context. From a database perspective, storage-level encryption has the advantage to be transparent, thus avoiding any changes to existing applications



(b) Database-level encryption

Allows securing the data as it is inserted to or retrieved from the database. Encryption strategy can be the part of the database design and can be related to database sensitivity. Selective encryption is possible and can be done at various granularities such as tables, columns and rows. It can be related with some logical conditions. Depending on level of encryption of the encryption feature the encryption process may incur some changes to application.

For both strategies database is decrypted on the database server on time. Thus encryption keys must be transmitted on server side.



Manuscript received May 18, 2014.
Shivnandan Singh, PG Scholar, Galgotias University Greater Noida, UP, India.
Pratibha Singh, PG Scholar, Galgotias University Greater Noida, UP, India.
Niranjan Kumar Agri, PG Scholar, Galgotias University Greater Noida, UP, India.

(c) Application Level Encryption

Moves encryption/decryption process to the application that generates data. Encryption is thus performed with in the application that introduces the data in system, the data is send encrypted thus naturally stored and retrieved encrypted[1][2][3].To be finally decrypted within the application. This approach has the benefit to separate encryption keys from the encrypted data stored in the database since the keys never have to leave the application side. However, applications need to be modified to adopt this solution. In addition, depending on the encryption granularity, the application may have to retrieve a larger set of data than the one granted to the actual user, thus opening a security breach Indeed, the user (or any attacker gaining access to the machine where the application runs) may hack the application to access unauthorized data.

Finally, such a strategy induces performance overheads (index on encrypted data are useless) and forbids the use of some advanced database functionalities on the encrypted data, like stored procedures (i.e., code stored in the DBMS which can be shared and invoked by several applications) and triggers (i.e., code fired when some data in the database are modified).

PROPOSED ALGORITHM REA

A novel encryption algorithm REA is recommended, because of its simplicity, efficiency, and security. It can outperform competing algorithms. In this section provides a comprehensive yet concise algorithm. Also, gives a general analysis of the functioning of these structures.

The proposed algorithm REA is a symmetric stream cipher that can be effectively used for encryption and safeguarding of data. It takes a variable-length key, making it ideal for securing data. The REA algorithm encipherment and decipherment consists of the same operations, only the two operations are different: (1) added the keys to the text in the encipherment and removed the keys from the text in the decipherment. (2) Executed divide operation on the text by 4 in the encipherment and executed multiple operation on the text by 4 in the decipherment. Divide operation by 4 on the text to narrow the range domain of the ASCII code table at converting the text. The details and working of the proposed algorithm REA are given below.

Encryption Algorithm of the REA

The steps of the encryption algorithm REA are presented in the following steps.

Step1: Enter the text and the key.

Step2: Addition operation of the key to the text.

Step3: Convert the previous text to ascii code.

Step4: Convert the previous ascii code to binary data.

Step5: Reverse the previous binary data.

Step6: Gather each 8 bits from the previous binary data and obtain the ascii code from it.

Step7: Divide the previous ascii code by 4.

Step8: Obtain the ascii code of the previous result divide and put it as one character.

Step9: Obtain the remainder of the previous divide and put it as a second character.

Step10: Return encrypted text.

Conclusion

It is very important for us to encrypt the database on several levels because data is very important element for any organization and it is also very important to implement the encryption algorithms to implement the encryption. There are so many algorithm for database encryption but we have to implement most efficient encryption algorithm for database encryption.

References

[1] Hacigümüs H., Iyer B., Li C., Mehrotra S., Providing Database as a Service, International Conference on Data Engineering (ICDE), 2002, pp. 29-39.

[2] Rakesh Agrawal, Jerry Kiernan, Ramakrishna Srikant, Yirong Xu, Hippocratic databases, Proceedings of the 28th international conference On Very Large Data Bases, 2002, pp.143-154.

[3] Luc Bouganim and Philippe Pucheral, Chip-secured data access: confidential data on untrusted servers, Proceedings of the 28th international conference on Very Large Data Bases. 2002, pp. 131-142.