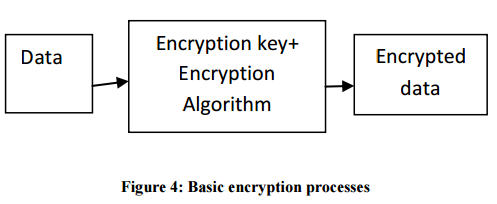
A database encryption scheme should meet several requirements. Among them are the requirements for data security, high performance, and detection of unauthorized modifications [2].

*[2]Min-Shiang H, Wei-Pang Y (1997) Multilevel Secure Database Encryption with Subkeys. Data and Knowledge Engineering 22, 117-131.*

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.414.1729&rep=rep1&type=pdf>

Title: Review of Attacks on Databases and Database Security Techniques

Data Encryption This is the basic technique used for securing any kind of information or data. So this technique can even be applied to databases.



Encryption is a process of translating plain text to encoded form called cipher text. This is usually carried out using secret encryption key and cryptographic cipher.

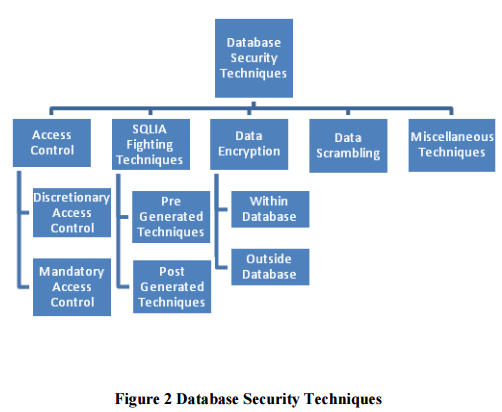


Figure 2 illustrates the basic process of encryption. Data are encrypted using encryption keys and encryption algorithms. Encrypted data are then stored in the database and decrypted when need to be used for processing purpose. There are two basic types of encryption commonly used. Symmetric Encryption is the type of encryption where a single secret key is used for both encryption and decryption. Asymmetric encryption is the type of encryption where a pair of secret keys is used. One of the keys is used for encryption and the other used for decryption. While performing database encryption, a decision about whether to perform the encryption inside or outside the database must be taken. Some of the issues involved in this technique are How to secure keys from attacker of the system? How to give administrative rights of manipulating data using keys? And How to provide limited access for keys? It is also important to provide proper authentication mechanisms because without them, it is easy to get access to keys using social engineering techniques [7].

(*[7 ] RSA Security Inc., “Securing Data at Rest: Developing a Database Encryption Strategy, A White Paper for developers, e-business managers and IT‖, Website, September 7 2012, http://www.rsa.com/products/bsafe/whitepapers/DDES\_WP\_0702.pdf*)

The important aspects which need to be considered while encrypting database is how to manage the encryption keys. Some of the aspects related to this issue are Number of encryption keys required, storage of keys, protection for the access of keys, and frequency of change of keys Recommended approach for storing the keys is, separate the keys and data residing in the database. Generally the keys are stored in hardware like access restricted files or hardware storage modules. The process of encryption can be performed either within the database or outside the database. If encryption is performed within the database, then there is less impact on application environment. But there are performance and security tradeoffs which need to be considered while implementing this policy. Understanding the encryption algorithm supported by DBMS also plays key role while devising strategy to implement this technique. The drawback of this approach is encryption keys also are stored in the same database. Another way to implement encryption in database is performing it on separate encryption servers. Encryption and decryption computations are performed encryption server. So here overhead of encryption is removed from DBMS and moved on to separate encryption servers to maintain the performance of DBMS. Encryption keys and data can also be separated. This approach is usually followed while encrypting database [7].The algorithms which are generally used for database encryption and often supported by DBMS are DES, Triple DES, RC2, RC4, DESX and AES.

The database encryption scheme can be implemented using different approaches. There are two main things to consider while considering database encryption. First thing is granularity of the data to be encrypted or decrypted. Granularity can be field level, row level or page level. Row or page level granularity may lead to encrypting large amount of data which can be overhead on the system. So generally column level encryption of only sensitive data is performed. The second thing is choice of encryption algorithm which is suitable for encrypting given data in database [8].

(*[8 ] Lianzhong Liu, JingfenGai, ―A new lightweight database encryption scheme transparent to applications‖, Piscataway, NJ USA, IEEE international conference 2008, page 135-140*.)

One encryption system approach describes two phases called initialization phase and run phase. In the initialization phase, all the metadata like the columns to be encrypted, the type and length of the columns, encryption algorithm and encrypted columns on which index is required. Such metadata is stored in the Security Dictionary. It will be loaded into memory first time it is used.

In the run phase of this scheme, the application does the normal activities performed on the database without thinking about encryption. Encryption/decryption engine performs data encryption and decryption based on metadata stored in Security Dictionary [8].

There are various configurations available for encrypting and decrypting databases. Some of them are listed below [2].

*([2 ] Shmueli, Erez, Vaisenberg, Ronen, Elovici, Yuval and Glezer, Chanan(2009)Database Encryption- An Overview of Contemporary Challenges and Design Considerations SIGMOD Record vol38, No 3.*)

File System Encryption: Here the physical disk where database resides is encrypted. Entire database is encrypted using single encryption key so discretionary access control cannot be implemented.

DBMS Level Encryption: There are many schemes for this kind of encryption. One scheme is based on Chinese Remainder theorem in which every row is encrypted using different sub keys for different cells. So encryption at row level and decryption at cell or field level is possible by this scheme.

There are some schemes based on Newton’s interpolation polynomials which are used for database encryption.

There is a SPDE scheme which encrypts each cell I the database with its cell coordinates like table name, column name and row id etc. So in this scheme static leakage attacks and splicing attacks are prevented.

Application level Encryption: In this technique, a middleware is suggested which translates queries fired by user into new bunch of queries which will execute on encrypted database.

This technique was implemented in Data Protector System.

Client-side encryption: This technique is generally used in case of ―Database as a service‖ scenario where the entire database is outsourced by the organization to reduce the maintenance costs. So here data privacy is the major concern. Encryption is the basic solution in this scenario.

Indexing encrypted data: There are many indexing mechanisms proposed. B tree index structure is prepared over plain text values in the table and then encryption of the table is performed at the row level. Encryption of the B tree is done at the node level.

Another scheme involves constructing index on plain text values and then encryption of each page of the index is done separately. One more modification is suggested which involves encrypting different index pages with different keys depending on page number.

There is another scheme suggested which computes XOR of plain text values with sequence of pseudo random bits which are generated by the client according to plain text value and a secure encryption keys.

A database encryption system must adhere to some characteristics such as it should be secure enough so that it requires high work factor to break, encryption and decryption should be performed fast without compromising DBMS performance, encrypted data should be small compared to unencrypted data, it should be possible to perform encryption and decryption of records without taking into consideration their physical or logical position in database, encryption scheme must support logical sub schema concepts of databases, encrypted record should be one value which is function of all fields, the encryption scheme should be as flexible as possible with respect to combinations of read and write operations, encryption system should not force DBMS to keep duplicate copies of data so that sub schema should be supported [9].

*([9 ] George I. Davida, David L. Wells, ―A database encryption system with sub keys‖, ACM transactions on database systems, vol.6, No.2, June 1981, page 312-328*)