ABSTRACT

The security of database is very important as data is the most valuable asset in the modern environment. Various security methodologies are available to secure databases. Most of the available methodologies focus on access control of the database, but there may be situations when someone intentionally or accidentally break or bypass the access control mechanism of the database and hence the confidentially of the database could be compromised. Variuos methods are available to preserve the confidentially of the database in those situations. The best solution in this case is encryption. Various Researchers proposed different methods by which the database encryption can be enforced and the content of database is encrypted and decrypted efficiently. All Encryption techniques available till date converts plain text into cipher text but the length of both will be same, this is a serious shortcoming of available cryptographic approach for example some attacker can guess the plain text by accessing the length of cipher.

In this paper we elaborate the use of database encryption to protect the database content. We propose a User supplied Key based authentication which generates the cipher of different length in comparison to plain text. By the use of this approach database can be protected against various kind of vulnerabilities.


INTRODUCTION

The database security is the process to secure the database from unauthorized access, malicious attacks by hackers and prevent the contents of database form accidental damages. The major threat to database content is –

- Loss of confidentiality
- Loss of integrity
- Loss of availability
- Loss of privacy and
- Loss of security

The modern database security model focuses on both database management and security of database. The security issues in Database may arise because of intentional or unintentional malicious activities done by hackers, legal users, Database Administrators, Application Designer and/or Network Administrators. These malicious activities can cause harm to internal database content either partially or fully like disclosure of confidential information, deletion of record, modification of record and execution of illegal transactions. Various methods of protection are available to protect the database from these security breaks. One major approach to deal with these situations is by the use of Cryptography.

The Database Encryption is a key concept because the database is now used in every field where process is computerized. The database is most vulnerable to security threats because this one of the best target for hackers.
and database attackers. The hackers prefer to break the database security because here they can get a lot of confidential information by breaking the security once. [6][7][2]

To overcome from these security problems in this paper we propose the use of User Supplied Key Based Encryption. By the use of this approach we can ensure the database security even if the access to database is compromised.

CRYPTOGRAPHY AND DATABASE SECURITY
The work Cryptography came from the Greek word “kryptos,”[10] which means hidden, and “graphia,” which mean writing, in this way cryptography is a way of writing text that is hidden so as no one can see the text in original way easily.

In other words Cryptography is a method used for the purpose of protecting the data either in a standalone computer or in client server environment. Cryptography is composed up of two steps, encryption and decryption.[12]

Encryption is the process of converting plain text into cipher text and Decryption is the reverse process. Both encryption and decryption are done by the use of some keys. These keys secured and are known only by the legal users.

![Fig 1. Cryptography in Standalone Database environment](image1)
The fig 1 Shows the situation where the cryptography is applied in the standalone Database environment. Here the plain text is converted into cipher text and stored into database and when the information is to retrieved from the database it is fetched and decrypted.[13][14]

![Fig 2. Cryptography in Network Database Environment](image2)
The fig 2 Shows the situation where the cryptography is applied in client server or network environment. Here the sender of the data first convert the plain text into cipher text by encryption and then send it to someone, now at the receiving end the data is received and decrypted by the use of decryption key.

**ENCRYPTION METHODOLOGY**

All Encryption techniques available base on the following two methodologies –

**Permutation:**

The permutation or transposition is a method of encryption by which the position of characters or set of characters are shifted according to a regular system in such a manner that the cipher text obtained contains a permutation of the plain text. The process of permutation is represented in the following example –

![Transposition technique of Encryption](image)

**Substitution:**

The substitution is a method of encryption by which unit of plain text are replaced by another unit of cipher text. The process of Substitution is represented with the following example –

![Substitution technique of Encryption](image)
LITERATURE REVIEW
N. A. Ghani [1] defines privacy protection becomes one of the important requirement in Web-based information system that deal with personal information. Owner should know to what extent information about them has been collected and how the information will be used and how they are able to control their own information. They explained Hippocratic database used the purpose as a central concept in the database development. But, owner is unable to control their Personal Information stored in HDB regardless they are able to. They discussed and introduced an initial architecture of owner-controlled Hippocratic database.

Al-Fedaghi, S. [2] defines sensitivity of personal information is one of the most important factors in determining the individual's perception of privacy. A "gradation" of sensitivity of personal information can be used in many applications, such as deciding the security level that controls access to data and developing a measure of trust when self-disclosing personal information. Al-Fedaghi, introduces a theoretical analysis of personal information sensitivity and defines its scope and puts forward possible methods of gradation.

K Hemanth [3], a secure and blind biometric authentication protocol, which addresses the concerns of user’s privacy, template protection, and trust issues. The protocol is based on asymmetric encryption of the biometric data; it captures the advantages of biometric authentication as well as the security of public key cryptography. The authentication protocol can run over public networks and provide non reputable identity verification. They proposed an approach that makes no restrictive assumptions on the biometric data and is hence applicable to multiple biometrics. They analyze the security of the protocol under various attack scenarios.

Stephane Jacob[4] explained Protecting the confidentiality in large databases without degrading their performance is a challenging problem, especially when encryption and decryption must be performed at the database-level or at the application-level. They focus on symmetric ciphers for database encryption since they are the only type of ciphers with acceptable performance for most applications. They point out that stream ciphers are the adequate type of encryption schemes.

PROBLEMS IN EXISTING APPROACH
The methodologies mentioned above are effective way of enforcing security but it possess the following major shortcomings –
All Encryption techniques available till date transforms plain text into cipher text but the length of plain text and cipher text is same. This the major fault because by assessing the cipher Text attacker can find the length of words and hence can guess some/all words by trying all possible words in the vocabulary.

PROPOSED METHODOLOGY
To overcome from the above mentioned problems in Database Security we propose the use of User Supplied Key base Encryption which is explained bellow-

**User Supplied Key based Encryption:**
in this approach we take a 128 bit key which is used to store the key value provided by the user. It is an integer variable which stores the key on the basis of which the database contents is encrypted the proposed method works in the following two steps-

**Encryption:**
in this step first the user provide the authorization credentials to login to the database, after verifying the credentials user is allowed access to the database now user provides his chosen key to encrypt the desired content of the database. The encryption process works by the following algorithm.

**Encryption Algorithm:**
- We take the following variables –
  - int uk: to store the user supplied key
  - char a, b : to store the plain and cipher text characters.
  - int n: to store the positional equivalent of the character in plain text character variable a.
  - int m: to store the positional equivalent of the character in cipher text variable b.
Now the proposed encryption works in following way –

**Step 1** – read a character of the field in variable a.

**Step 2** – now \( n = \text{positional equivalent of } a \).

The positional equivalent are taken as follows

- From a-z \( 1-26 \)
- From A-Z \( 27-52 \)
- From 0-9 and space \( 53-63 \)

**Step 3** – \( t = n + uk \)

**Step 4** – \( x = t \mod 63 \)
\( y = t / 63 \)

**Step 5** – \( b = \text{character equivalent of position in } x \).

**Step 6** – store the char in variable b along with y in the field.

**Step 7** – continue the step 1 to 6 till there is unprocessed character in the field.

**Example:**

let the value contend in some field of the database is surya pratap singh . Then the algorithm works in the following manner –

Let \( uk = 125 \)

**Iteration 1:**

a=’s’

\( n = \text{positional equivalent of character in } a \) (i.e. 19)

\( t = n + uk \)
\( = 19 + 140 \)
\( = 159 \)

\( x = t \mod 63 \)
\( = 159 \mod 63 \)
\( = 33 \)

\( y = t / 63 \)
\( = 159 / 63 \)
\( = 2 \)

Now \( b = \text{character equivalent of value in } x \)

\( b = ‘33’ \)

Now we store the value of \( b \) and \( y \) in the processed field i.e. 62

**Iteration 2:**

a=’u’

\( n = \text{positional equivalent of character in } a \) (i.e. 19)

\( t = n + uk \)
\( = 21 + 140 \)
\( = 161 \)

\( x = t \mod 63 \)
\( = 161 \mod 63 \)
\( = 35 \)

\( y = t / 63 \)
\( = 161 / 63 \)
\( = 2 \)

Now \( b = \text{character equivalent of value in } x \)

\( b = ‘8’ \)

Now we store the value of \( b \) and \( y \) in the processed field i.e. 82

Now in the similar way the algorithm encrypt the remaining characters. Now “surya pratap singh” becomes “628252B2o2D232o272o232D262w212u2v2”. In this manner the database content is encrypted.

**Decryption:**

In this step the valid user can decrypt the database content. The decryption algorithm works by the following algorithm.
Decryption Algorithm:

- We take following variable for this purpose
  - int uk: to store user supplied key
  - Char a: to store the character values in the field from 1,3,5,7.... positions
  - int y: to store the converted character value in the field from 2,4,6,8,... positions.
  - int m: to store positional equivalent of char in a
  - int n: to store the final positional equivalent
  - int t: to store the processing values.

- Now the decryption algorithm works in following manner
  - Step 1 – read first character of the field in variable a and second variable in y.
  - Step 2 – now m = positional equivalent of a
  - Step 3 – t = m + y * 63
  - Step 4 – n = t - uk
  - Step 5 – b = character equivalent of position in n.
  - Step 6 – store the char in variable b in the field.
  - Step 7 – continue the step 1 to 6 till there is unprocessed character in the field.

Example - we explain it by the use of previous example let the cipher text be “628252B2o2D23252o272o232D262w212u2v2”. The decryption algorithm works in following manner

**Iteration 1:**

- a = ‘6’
- y = 2
- m = positional equivalent of value in a (i.e. 15)
- t = m + y * 63
  - = 33 + 2 * 63
  - = 33 + 126
  - = 159
- n = t – uk
  - = 159 – 140
  - = 19
- Now b = character equivalent of value in n
  - i.e., b = ‘s’
- Now we store the value of b in the processed field

**Iteration 2:**

- a = ‘8’
- y = 2
- m = positional equivalent of value in a (i.e. 17)
- t = m + y * 63
  - = 35 + 2 * 63
  - = 35 + 126
  - = 161
- n = t – uk
  - = 161 – 140
  - = 21
- Now b = character equivalent of value in n
  - i.e., b = ‘u’
- Now we store the value of b in the processed field

In the same we the algorithm decrypt the remaining characters. Now “628252B2o2D23252o272o232D262w212u2v2” becomes “surya Pratap singh”.

**SIMULATION AND RESULT**

We implemented our proposed encryption method by the use of JAVA Swing as front end and My SQL as back end. The fig 5 shows the login screen to assess the database.
Here user has to pass username and password, when the username and password is matched the user proceeds to next form where the user can choose encrypt or decrypt as an option it is represented in fig 6.

If the user selects Encryption option he/she proceed to the Encryption form represented in fig 7. Here the user have to first select the table name to be accessed Now user have to options show the content of the table and Encrypt and show. To encrypt the records the user first enters the key to encrypt and click on Encrypt and show button. The contents of the table is encrypted accordingly and stored in the selected table.
For the decryption purpose we use the same user supplied key to decrypt the database content. Here the user have to provide the decryption key and then click on the decrypt and show button then the original plain text data is represented on the users screen. The decryption process is shown in the fig 8.
CONCLUSION
The database security is gaining the prime importance in the modern environment because it is the best way to secure the database contents. The database encryption can work in un-trusted environment where we cannot ensure secure access to the database. If Encryption techniques are enforced properly the user who somehow intrude in the database cannot decrypt, i.e., cannot understand the database contents. In this paper we explained security issues of the database and we propose the use of User Supplied Key based Encryption by which we can protect the database content effectively.

In this paper we implemented the encryption and decryption of database content by the use of user supplied key where the length of cipher text and plain text is different.

REFERENCES