Database Encryption Using Enhanced Affine Block Cipher Algorithm

NOOR HABIBAH ARSHAD, SAHARBUDIN NAIM TAHIR SHAH, AZLINAH MOHAMED, ABDUL MANAF MAMAT
Faculty of Information Technology & Quantitative Sciences
Universiti Teknologi MARA
40450 Shah Alam, Selangor
MALAYSIA

Abstract: - Databases are vulnerable to attack from internal and external threats. Sensitive data stored in database appeared as target to attackers. Adding the database encryption, valuable information in database becomes more secure since the encrypted data ensure the confidentiality of the data. A new affine block cipher named Enhanced Affine Block Cipher technique is proposed for database encryption. This algorithm improves the weakness of the original affine cipher. The new encoding schema and modification Cipher Block Chaining (CBC) mode of operation for block cipher is designed for the new algorithm. The result has shown that the algorithm is working properly, where the decryption process produced similar output as the original plaintext and it ran through specified configuration and evaluated thoroughly with respect to database approach and algorithm technique to prove the design.

Keywords: - Database; Enhanced Affine Block Cipher; Encryption; Decryption; Cipher Block Chaining.

1 Introduction

Database becomes the storage of valuable information for individual or organization, contains data ranging from different degree of confidentiality, and widely accessed by various users. In today's enterprise environment, database systems are distributed and used in various applications such as e-Banking and e-Commerce. These applications are examples of real-time online resources that need to deliver value-added services through high confidentiality and availability of databases. As databases become networked in more complex multi-tiered applications, their vulnerability to external attack increases and critical business data stored in databases are obviously vulnerable for attackers. Therefore, to properly maintain the integrity and confidentiality of the data, database security becomes one of the most urgent challenges in database research. Database security is a wide research area and includes topic such as statistical database security, intrusion detection and most recently privacy preserving data mining [4]. One of the requirements for database security is database encryption.

Thus, this paper will focus specifically on some of the details on cryptographic algorithm technique used to implement the database encryption. Throughout this paper, the cryptography algorithm that will be used to provide security and confidentiality of data in the database are discussed and elaborated.

2 Database Encryption

In general, database sharable resource among many user or applications. A multiuser application in distributed system complicates the data security problem imposed upon a database. Hence, security is becoming one of the most urgent challenges in database research and industry. Past studies reviewed that database security is the most common architectures and methodologies for designing secure database [7, 5]. One of the important

aspects of database security is database encryption [1, 2, 9].

The original data that is readable and understood is called plaintext or cleartext. Method that used to code a plaintext that can conceal its meaning is called encryption. Once a message has been transformed with an encryption algorithm, the resulting message is called ciphertext. The encryption is used to ensure that information is hidden from unintended person, even from those who can see the encrypted data. In order to be able to read ciphertext, the other process is needed to decipher the ciphertext. The study of encryption and decryption is called cryptography [8]. According to Menezes *et al.* [6] and Russell and Gangemi [8] cryptography provides security in the areas of confidentiality, data integrity, authentication, and non-repudiation.

The goal of encryption is to make data unintelligible to unauthorized users and extremely difficult to decipher when it is attack. Symmetric key cryptography is the most commonly used technique to encrypt data in the storage or database. This ciphers use the same key when to encrypt and decrypt the data. There are two types of symmetric ciphers; block ciphers and stream ciphers. Stream ciphers are generally twice as fast as block ciphers but they require the use of unique keys. Block ciphers on the other hand, allow keys to be reused. There are some encryption features of block cipher technology were included in Database Management System (DBMS). The recommended minimum key length for all symmetric key ciphers is 128 bits.

A block cipher is a type of symmetric key cryptography that transforms a fixed length block of plaintext (unencrypted text) data into a block of ciphertext (encrypted text) data of the same length. This transformation takes place under the action of a user-provided secret key. Decryption is performed by applying the reverse transformation to the ciphertext block using the same secret key. An early and highly influential block cipher design was the Data Encryption Standard (DES), developed at IBM in 1974, and published as a standard in 1977. A successor to DES, the Advanced Encryption Standard (AES), was adopted in 2001 [11].

The affine block cipher [10] is one of the symmetric key cryptography that was known as classical cryptography and it is easier to break by ciphertext-only cryptanalysis. Some improvements have been done on affine cipher. Instead of using single letter, Koblitz [3] shows digraphs in his works but it is still not enough because of the second letter of each ciphertext digraph depends only on the second letter of the plaintext digraph. Thus, one could obtain a lot of information keys from a frequency analysis of the even numbered letters of the ciphertext. In this paper, enhanced affine block cipher algorithm with its encoding schema was designed to overcome affine cipher and it was implemented in securing data stored in database.

3 Database Encryption Approach

There are two main approaches for database encryption which is whether performing encryption and decryption inside the database or performing encryption and decryption outside the database. After reviewing the database encryption, the best ways to secure the information stored in database is database encryption and apply it at outside the database i.e. at application level encryption. This approach was selected because it provides good end-to-end data protection. By using this approach, encryption will be on the column and row basis. Hence, not all data stored in the database will be encrypted. Only sensitive information such as user identification, credit card number and password will be encrypted. By applying this approach, it will be more efficient in reducing the overhead of reading data. The cryptographic algorithm used for the database encryption is designed and implemented in java programming language and it acts as application server whereby the encryption and decryption processes are done at the application level.

This approach applied end-to-end encryption between client and applications

server. For encryption process, the data is encrypted at application server and then inserted into the appropriate fields or columns in the database. For decryption process, the encrypted information is retrieved from the database and then decrypts it at application server so that only authorized user can see the information. The keys used to encrypt and decrypt the data in this approach is stored in file storage at application server not in the database. Hence, this approach will add one security layer in securing the data stored in the database. The keys must be found before the attacker can see and know the contents of data. Figure 1 depict database encryption outside the database.

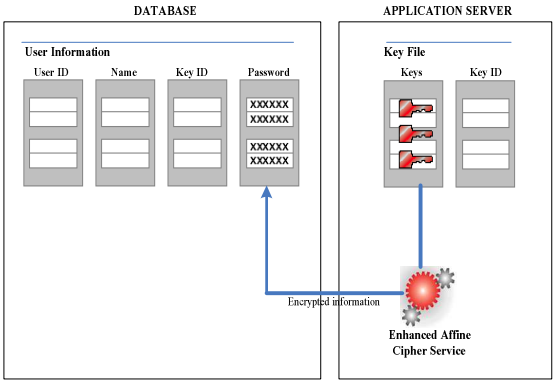

Figure 1 Applying Database Encryption outside the Database

4 Enhanced Affine Block Cipher

The analysis on affine block cipher was done and revealed that some new features can be added into its cipher such as the encoding schema and mode operation of block cipher. Therefore the new affine block cipher was designed and called enhanced affine block cipher to overcome the weaknesses of the original affine cipher. For implementation of these algorithms, the activity diagram was used to model the workflow behind the implemented system. The activity diagram is useful in understanding work flow analysis of synchronous behaviours across the process.

Figure 2 shows the process flow of encryption and decryption using Enhanced Affine Block Cipher. As seen in figure 2, the process started with either plaintext or ciphertext format as an input.

When plaintext is taken as an input, the Encoding activity is performed and followed by the Encryption activity and next the DecodingHex activity. The DecodingHex activity indicates that both Display Result activity and the Store Result in Database activity occur at the same time.

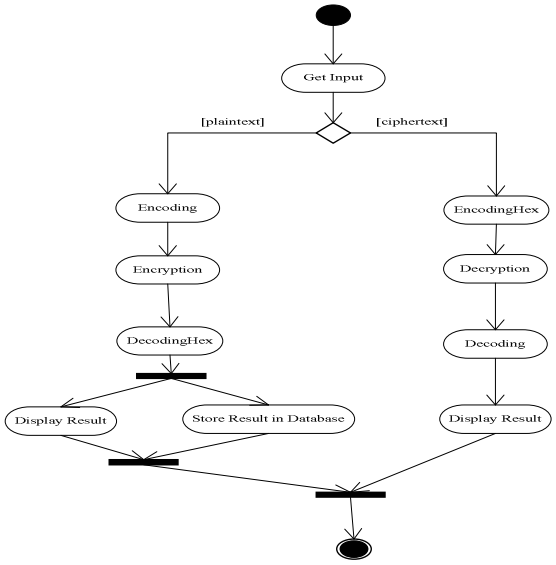

Figure 2 Activity Diagram for Database Encryption using Enhanced Affine Block Cipher

Meanwhile if ciphertext is the input, the EncodingHex activity is performed and would then indicate the Decryption activity and next the Decoding activity. The Decoding activity indicates the Display Result activity. Finally the parallel activities are combined to end the activity.

The inputs that have been used in the encryption process are plaintext, key, block length and initial vector. In the decryption process, the ciphertext, key, block length and initial vector are its input. The plaintext was divided into simple and long plaintext. The main purposes of the testing are to validate the functionality of the algorithm and also to ensure that the database encryption is working properly. From the result, it was found that the algorithm is working properly where the decryption process produced a similar output to the original plaintext.

4.1 Design of Enhanced Affine Block Technique

The design of enhanced affine block technique would be described in the next sections.

4.1.1 Encoding Schema

The encoding schema designed and developed was based on ASCII format. The plaintext and ciphertext is code and decode into certain number or value before encryption or decryption process. Hence, the encoding schema was used to enhanced affine block cipher.

The total of the ASCII characters set is 128. Therefore, the encoding schema is used based on these numbers where it contains encode and decode schema. In this encoding schema, during encryption process, the number will be converted into hexadecimal code whereas during decryption process, the number will be converted into characters.

Before plaintext and ciphertext is encrypted or decrypted, it was broken up into message units (block size). A message unit might be a single letter, a pair of letters (digraphs), a triple of letters or any number of letters. The encoding schema of message unit is done by an enciphering transformation function where it takes any plaintext message unit and transformed into a ciphertext message unit. In other words, it is a map from the set of P all possible plaintext message units to a set of C all possible ciphertext message units. The encoding schema of message unit is also done by deciphering transformation function where it takes any ciphertext message unit and transformed into an original plaintext message unit. In other words, it is also a map from the set of C all possible ciphertext message units to a set of P all possible plaintext message units.

4.1.2 Encode and Decode Schema of Plaintext Message Unit

First, let start with encode schema and the case of a message unit (block size of plaintext message) is single letter in ASCII character (128 characters) was labeled by integer 0, 1, 2, 3... , 128-1.

For block size = 1, the message unit of plaintext is $p = x_1$. The formula of encoding schema is as follows:

$$p = x_1$$
$$= \sum_{i=1}^{1} 128^{1-i} x_i \quad \text{so for every p of plaintext}$$
$$\text{p} \in \{0,1,2,3,.. .128^{1} - 1\} = Z_{128}$$

With the same techniques, it can be applied for block size equal to two.

For block size =2, the message unit of plaintext is $p = x_1x_2$

$$p = 128x_1 + x_2$$
$$= \sum_{i=1}^{2} 128^{2-i} x_i \quad \text{so for every p of plaintext}$$
$$\text{p} \in \{0,1,2,3,...128^{2} - 1\} = Z_{16384} = Z_{128^2}$$

Therefore, with the same techniques it could be used for block size = n, the message unit of plaintext is $p = x_1x_2...x_n$

$$p = 128^{n-1} x_1 + 128^{n-2} x_2 + 128^{n-3} x_3 + ... + x_n$$
$$= \sum_{i=1}^{n} 128^{n-i} x_i \quad \text{so for every p of plaintext}$$
$$\text{p} \in \{0,1,2,3,...128^{\text{n}} - 1\} = Z_{128^n}$$

In decode schema of plaintext, the value or number was obtained from encrypting process is converted into appropriate code. The process of converting a number (decimal numbers) into digits y_n, $y_{n-1}...y_1$ and y_0 such that

$$y = 128^{n-1} y_1 + 128^{n-2} y_2 + ... + y_n$$

It can be obtained by successively dividing y by 128 until quotient is 0. So the values are the remainders y_n, $y_{n-1}...y_1$, y_0. In case of encryption, the combination of these values is in hexadecimal number and is called ciphertext message.

4.1.3 Encode and Decode Schema of Ciphertext Message Unit

First, let start with encode schema and the case of a message unit (block size of ciphertext message) is single letter in ASCII character

(128 characters) was labeled by integer 0, 1, 2, 3... , 128-1.

For block size = 1, the message unit of ciphertext is $c = y_1$
The formula of encoding schema is as follows:

$$c = y_1$$
$$= \sum_{i=1}^{1} 128^{1-i} y_i \quad \text{so for every c of ciphertext}$$
$$c \in \{0,1,2,3,.. .128^1 - 1\} = Z_{128}$$

With the same techniques, it was also apply for block size equal to two.
For block size =2, the message unit of ciphertext is $c = y_1 y_2$

$$c = 128 y_1 + y_2$$
$$= \sum_{i=1}^{2} 128^{2-i} y_i \quad \text{so for every c of ciphertext}$$
$$c \in \{0,1,2,3,...128^2 - 1\} = Z_{16384} = Z_{128^2}$$

Therefore, with the same techniques, it was also concluded that as follows;
For block size = n, the message unit of ciphertext $c=y_1 y_2 ... y_n$

$$c = 128^{n-1} y_1 + 128^{n-2} y_2 + 128^{n-3} y_3 + ... + y_n$$
$$= \sum_{i=1}^{n} 128^{n-i} y_i \quad \text{so for every c of ciphertext}$$
$$c \in \{0,1,2,3,.. .128^n - 1\} = Z_{128^n}$$

In decode schema of ciphertext, the value or number was obtained from decrypting process is converted into appropriate code. The process of converting a number (decimal numbers) into digits x_n, $x_{n-1}...x_1$ and x_0 such that

$$x = 128^{n-1} x_1 + 128^{n-2} x_2 + ... + x_n$$

It can be obtained by successively dividing x by 128 until quotient is 0. So the values are the remainders x_n, $x_{n-1}...x_1$, x_0. In case of decryption, the combination of these values is in decimal number and is called plaintext message.

4.2 Design Enhanced Affine Block Cipher

The Affine cipher works by transforming the letters of the alphabet to their corresponding numerical value (which is from 0 to 25), then utilize the encryption formula as follows;

$$e_{a,b}(x) = (ax + b) \bmod 26$$

This encryption function must be bijective, and a must have a multiplicative inverse mod 26 (gcd(a,26) is equal 1). For decryption function

$$d_{a,b}(y) = a^{-1}(y - 1) \bmod 26$$

The invertible integers mod 26 are set of {1,3,5,7,9,11,15,17,19,21,23,25}.

New affine cipher namely enhanced affine block cipher was designed based on the encoding schema as mentioned earlier. For the first step, recall the affine cipher as follows:

Let $P = C = Z_{128^n}$ and n is block size.

$$K = \{(a,b) \in Z_{128^n} \times Z_{128^n} : \gcd(a,128^n) = 1\}$$

for $K = (a,b) \in \kappa$,

the encryption functon is defined as

$$e_K(x) = ax + b \bmod 128^n$$

and the decryption function is defined as

$$d_K(y) = a^{-1}(y - 1) \bmod 128^n$$

where $(x,y \in 128^n)$

The second step for enhancement of affine cipher is done by adding modes of operation into the block cipher algorithm. . This technique is similar to cipher block chaining (CBC) mode. Figure 3 and figure 4 show that the processes of encryption and decryption of enhanced affine block cipher with its modes of operation.

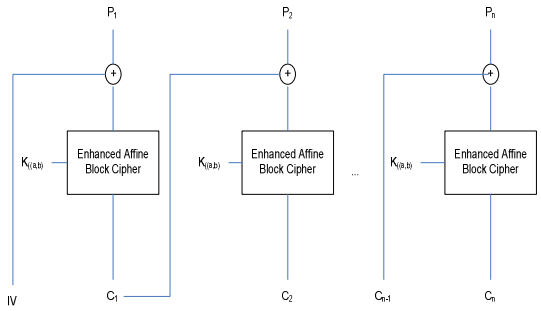

Figure 3 Modes of Operation during Encryption Process

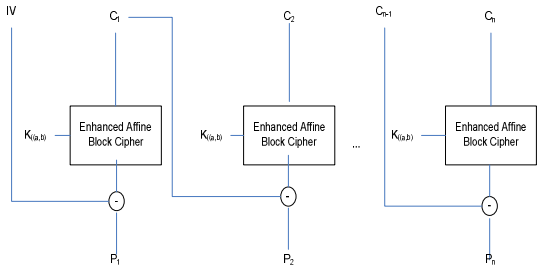

Figure 4 Modes of Operation during Decryption Process

It was discovered that, by applying CBC mode; during the operation XOR, certain values are more than the value of the modulo 128^n. Due to the basic properties that is congruence between the value and the modulo; it cannot give exact value during the decryption process. Based on analysis and initial testing, instead of using XOR as mode of operation, this algorithm was used as an additional operation for encryption and subtraction operation for decryption. The mathematical formula for this mode of operation is as follows:

$$C_i = e_K(P_i + C_{i-1}),\ C_0 = IV \text{ for encryption}$$

and

$$P_i = d_K(C_i) - C_{i-1},\ C_0 = IV \text{ for decryption}$$

In encryption process, each plaintext block is added with previous ciphertext block, and then encrypted. An initialization vector (IV) is used as a seed for the process. In decryption process, each decrypted ciphertext block is subtracted with the previous ciphertext.

This proposed enhanced affine block cipher could be used for any application systems which needs the sensitive data to be protected.

5 Conclusion

This paper focused on the design of database encryption at application level using enhanced affine block cipher. This improvement has been made because of the weakness found in the original affine cipher. In this paper, the improvement is made by using a new encoding schema and mode of the operation for the encryption and decryption process. The enhanced affine block cipher is developed and implemented where the selected sensitive data is encrypted outside the database (application level) and then it is inserted into database.

The enhanced affine block cipher can be used to explore other existing symmetric cryptography algorithms or combine it to other techniques. The mode of operations used in enhanced affine block cipher also can be extended into others approaches. The database encryption can also be applied in hybrid cryptography techniques. This technique can be applied by combining the symmetric key cryptography and asymmetric key cryptography.

References:

[1]Chen, G., Chen, K., Dong, J., "A Database Encryption Scheme for Enhanced Security and Easy Sharing,"*Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design*, 2006

[2]He, J. and Wang, M., "Cryptography and Relational Database Management Systems," IEEE, 2001.

[3]Koblitz,N., *A Course in Number Theory and Cryptography*. New York: Springer-Verlag, 1988.

[4]Mattsson, U.T. "A Practical Implementation of Transparent Encryption and Separation of Duties in Enterprise Databases: Protection against External and Internal Attacks on Databases," *IEEE International Conference*, 2005.

[5]Maurer, U.,"The Role of Cryptography in Database Security," ACM SIGMOD, 2004.

[6]Menezes, van Oorschot. P.C., and Vanstone, S.A., *Handbook of Applied Cryptography*. CRC Press, 1999.

[7]Piattini, M.G., Ferntindez-Medina, E., "Secure databases: State of The Art," *IEEE*, 2000.

[8]Russell, and Gangemi,G.T, *Computer Security Basics*. O'Reilly, 1991.

[9]Sesay, S., Yang, Z., Chen, J., and Du Xu, " A Secure Database Encryption Schema," *IEEE*, 2004.

[10]Stinson, D.R., *Cryptography;Theory and Practice.*CRC Press, 1995.

[11]Tropical Software, 2007. http://www.tropsoft.com/strongenc/des3.htm