A Secure Data Encryption Method by Employing a Feedback Encryption Mechanism and Three-Dimensional Operation

Yi-Li Huang, Fang-Yie Leu, Cheng-Ru Dai

A Secure Data Encryption Method by Employing a Feedback Encryption Mechanism and Three-Dimensional Operation

Yi-Li Huang, Fang-Yie Leu, Cheng-Ru Dai,

Department of Computer Science, TungHai University, Taiwan
{yifung, leufy}@thu.edu.tw

Abstract. Currently, electronic documents are commonly exchanged between/among government offices in many countries. When a government office would like to transmit a high-security-level-electronic document to another office, the sending end officer needs to encrypt it so as to protect the document from being known to hackers. AES and DES have been commonly and widely invoked to protect documents in recent years. However, the two algorithms have so far faced the threats of Brute-Force cracks. To avoid the threats, in this study, we proposed a new data encryption approach, called the Secure Data Encryption Method (SeDEM for short), in which plaintext and system keys are encrypted by using a sequential-logic style encryption approach which further employs a three-dimensional operation and a feedback encryption mechanism to effectively protect encrypted data from brute-force and cryptanalysis attacks. The feedback encryption mechanism is a feedback process in which each of its calculation iteration generates three internally-used dynamic feedback keys for the next calculation iteration. The purpose is to effectively improve the security level and unpredictability of generated ciphertext. The three-dimensional operation is employed to further increase the computational complexity of the encryption technique so as to enhance the security level of the ciphertext, and difficulty of cracking the keys.

Keywords: DES, AES, symmetric encryption, Feedback Encryption, three-dimensional computing, dynamic feedback keys.

1 Introduction

Recently, many governments have adopted electronic documents to substitute traditional paper documents, aiming to achieve a paperless homeland and environment. So before a high-security-level document is transmitted through networks or the Internet, for security consideration, an encryption mechanism [1-3] is often required. Also, when a military office delivers a command to one of its subordinates, for example, to attack an enemy group sometime later, the command must be encrypted [4] before being sent out, particularly when the delivery goes through a wireless communication system.

On the other hand, owing to the popularity of wireless communication, wireless systems have been quickly developed, and mobile devices have been commonly used in our everyday life. However, owing to the wireless transmission nature, hackers can easily eavesdrop and crack those messages sent through wireless channels. That is why security problems have been more serious and attracted many more researchers' attention than before. Presently Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are two of the cryptographic techniques most widely used to protect transmitted messages. But their keys are relatively short, and current computer processing speeds have been significantly improved. DES encryption algorithm was successfully cracked in 1999 [4-7], implying that DES is no longer a high security encryption mechanism. Although AES has not been successfully cracked, no one dares to say that AES is always secure enough to protect transmitted data. In the following, we will use documents and messages interchangeably since documents are carried in messages.

Both AES and DES block ciphering requires complicated calculation on their own parent keys so as to generate a certain number of sub-keys to encrypt plaintext. But the combinatorial-logic style calculation is quite a problem since its outputs only rely on current inputs, without employing previous outputs as a part of the inputs to increase the security level of its ciphertext. Hence, their ciphertext may be cracked relatively easier by hackers by using cryptanalysis attacks, like chosen plaintext attack, and attacks by statistical methods and by Brute-force methods [5]. Namely, security levels of this style of encryption techniques are not as high as expected. So how to improve their security levels has been one of the focuses of security researchers.

The principles of modern encryption mechanisms are that even though the encryption process of a technique has been disclosed, as long as the hackers do not know all the encryption keys, the plaintext (i.e., the delivered documents) is still safe since without acquiring all keys, it is almost impossible for hackers to crack the ciphertext. On the other hand, if a ciphertext is generated by using a combinatorial-logic block encryption technique, the sub-keys produced by the parent key given when the system starts up are the same, i.e., no matter how complicate the encryption process is, the same plaintext block will generate the same ciphertext block. In this case, hackers can analyze the relationship between plaintext blocks and their corresponding ciphertext blocks by using Brute-force cracking methods. Hence, due to high speed of current computer systems, a symmetric encryption mechanism may be no longer secure.

So, in this study, we propose a new encryption approach, called the Secure Data Encryption Method (SeDEM for short), in which plaintext and system keys are encrypted by using a sequential-logic style encryption method which further employs the Feedback Encryption mechanism and Three-dimensional Operation (FETDO for short) to solve the abovementioned problems. Here feedback encryption is an encryption technique, in which a computational result of an encryption round R is fed back to the encryption mechanism for the next encryption round, i.e., Round R+1, as a part of (R+1)'s inputs, thereby increasing the unpredictability of ciphertext. The three-dimensional operation, referring to three different computations, includes an addition (+) [8,9], exclusive-or ($\oplus$), and exclusive-and ($\odot$), when encrypting a plaintext block. The purpose is to increase the encryption complexity so as to reduce the probability of cracking the encryption process by hackers.

The rest of this paper is organized of follows. Section 2 briefly introduces the DES and AES. Section 3 describes the feedback encryption mechanism and the three-dimensional operation. Security analyses are presented and discussed in Section 4. Section 5 concludes this paper and addresses our future research.

2 Research Related

Block cipher refers to the process in which a fixed length plaintext is cryptographically manipulated by a series of operations so as to produce the corresponding secure ciphertext, often the length of which is the same as that of the plaintext.

2.1 Data Encryption Standard (DES)

DES is a typical block cipher technique, the block size of which is 64 bits. But in practice, the keys used by the DES algorithm to encrypt plaintext blocks are only 56 bits in length [4,5]. The remaining 8 bits are parity bits or unused, implying the ciphertext generated by this technique is not as secure as expected since a longer key's security level is generally higher than a shorter key's.

2.1.1 DES Structure

The DES encryption structure as shown in Fig. 1 consists of the initial permutation (IP for short), 16 processing stages (called 16 rounds) and the final permutation (IP^{-1} for short). IP and IP^{-1} are the mutual inverse arrays. Each of the 16 rounds contains a Feistel-function operation [4,5], denoted by F, and an $\oplus$ operation.

Before the first round, a plaintext block (64-bit) follows the given IP table to permute their bits. After that, the new 64-bit block is divided into two 32-bit subblocks. Let the right subblock be $IP_{1,1}$ which is directly input to the first Feistel function, named round-1 Feistel function which receives another input, called subkey1, to generate a result, $result_{1,1}$ (i.e., round1's 1^{st} result). Let the left subblock be $IP_{1,2}$ which is exclusive-ored with $result_{1,1}$ to generate $result_{1,2}$ (i.e., round1's 2^{nd} result). Let $IP_{2,1} = result_{1,2}$ and let $IP_{2,2} = IP_{1,1}$. The rounds continue. The general rule is that round i's Feistel function receives the two inputs subkey i and $IP_{i,1}$ to generate $result_{i,1}$ which is then exclusive-ored with $IP_{i,2}$ to generate $result_{i,2}$. After that, $IP_{(i+1),2}=IP_{i,1}$ and $IP_{(i+1),1} = result_{i,2}$, for all i= 1, 2,… …,16. Lastly, $IP_{17,1}$ is the right half and $IP_{17,2}$ is the left half of the 64-bit result of round 16. We input the right and left halves to IP^{-1} to produce the 64-bit ciphertext.

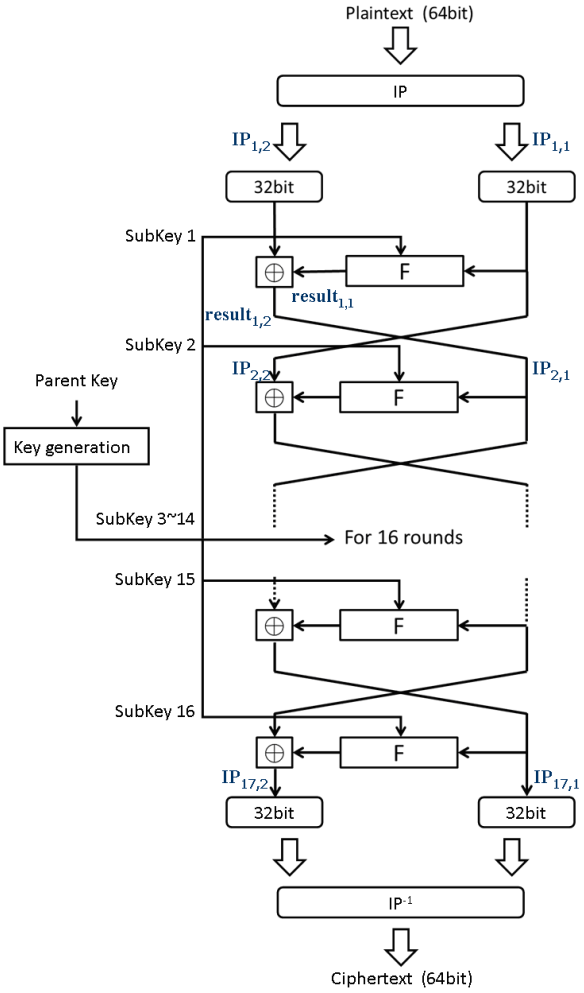

Fig. 1. The DES encryption structure [4,5]

2.1.2 Feistel function

The Feistel function's architecture, as shown in Fig. 2, consists of four main functions, including expansion, key mixing, substitution, and permutation, respectively, denoted by E, $\oplus$, S (named S-Box) and P.

Expansion transforms and extends a 32-bit pattern into 48 bits by using the expansion permutation [4,5]. Key mixing exclusive-ors E's output and a 48-bit sub-key to generate a 48-bit result which is divided into 8 6-bit patterns as the inputs of 8 S-boxes. Each S-box as a non-linear form transformation mechanism transforms a 6-

bit input to a 4-bit output, implying the output of the 8 S-boxes is 32 bits long. After that, permutation rearranges the 32-bit output based on a fixed permutation process. The final result is also 32 bits in length.

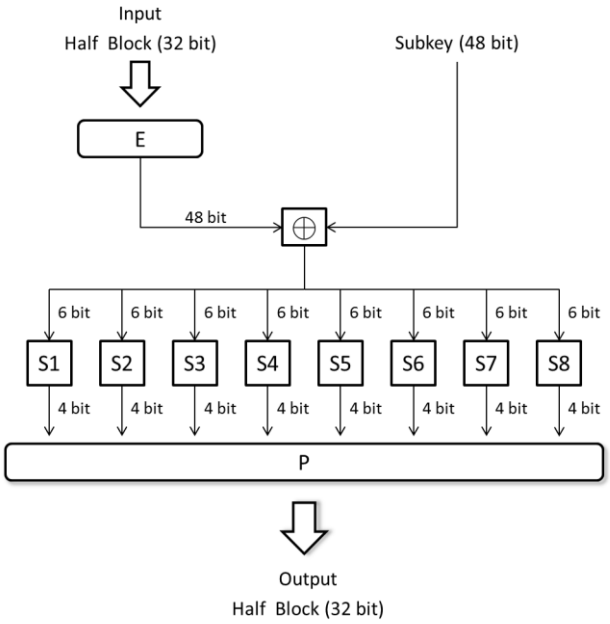

Fig. 2. The Feistel function of the DES [4,5]

2.2 Advanced Encryption Standard (AES)

The AES is also a kind of block cipher technique with block size 128 bits long. But its key length can be 128, 192 or 256 bits when necessary. The longer the length of the keys, the higher the security level of the system being considered. The AES uses a parent key to generate sub-keys.

Fig. 3 shows the AES encryption process which is performed on a 4 × 4 matrix, e.g., M, in which an element is 8 bits in length. The initial M contains a plaintext block, i.e., 128 bits (=4×4×8) in length. The AES encryption has 10 rounds. Each round, except the last one, comprises four stages:

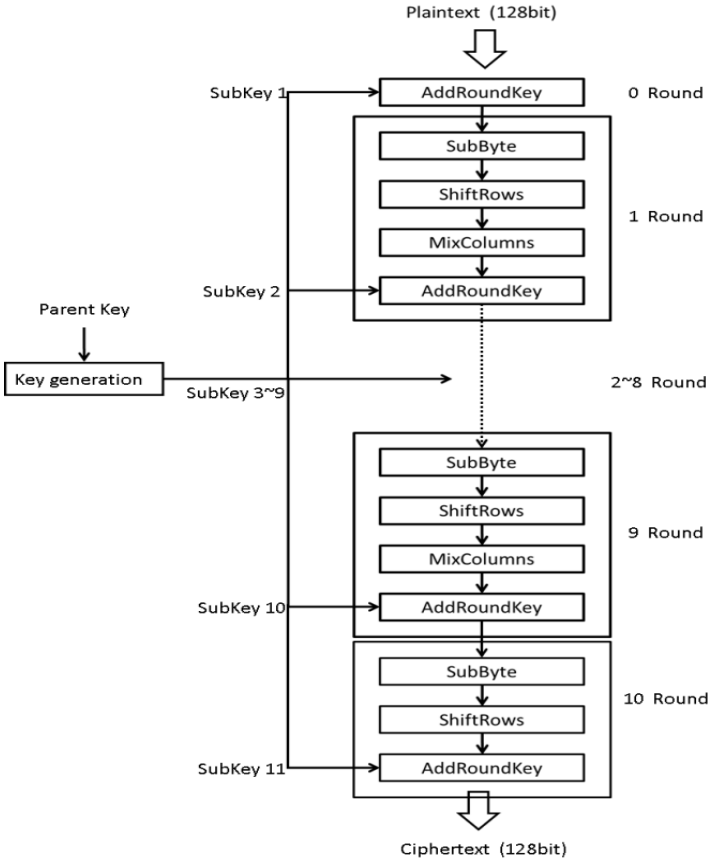

Fig. 3. The AES encryption process when key length is 128 bits [5]

Stage1: SubBytes. In this stage, an element of M, e.g., $a_{i,j}$, as shown in Fig. 4 is substituted by its corresponding element $a'_{i,j}$ which is retrieved from a pre-generated table, called Rijndael S-box [10-12], the elements of which are produced beforehand by invoking a non-linear function.

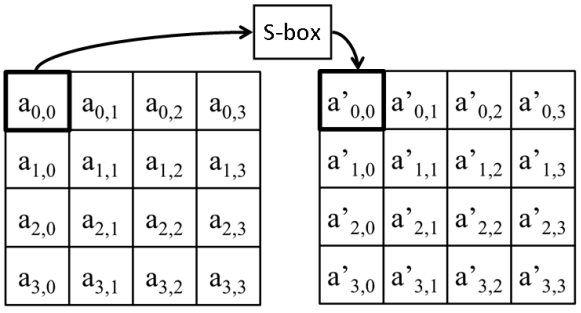

Fig. 4. The SubBytes stage [5,6]

Stage2: ShiftRows. In this stage, all elements of row r_i in M as illustrated in Fig. 5 are left rotated i times, $0 \leqq i \leqq 3$, even though the name of this stage is ShiftRows.

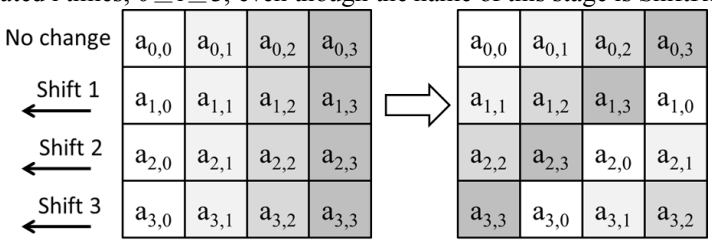

Fig. 5. The ShiftRows stage [5,6]

Stage3: MixColumns. The MixColumns stage as shown in Fig. 6 linearly converses a column $(a_{0,i}, a_{1,i}, a_{2,i}, a_{3,i})^T$, which is four bytes in length, to $(a'_{0,i}, a'_{1,i}, a'_{2,i}, a'_{3,i})^T$ by invoking the method of the Rijndael mix columns [10-12], implying an element of the matrixes in Fig. 6 is one byte in length. The conversion process is shown in Fig. 7.

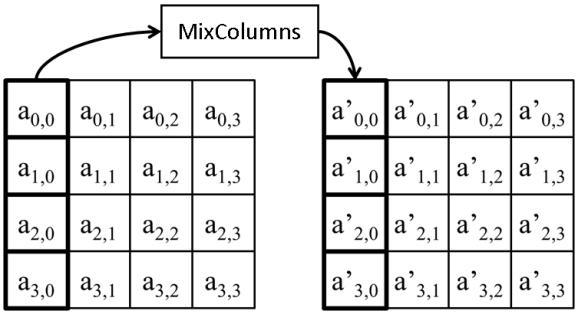

Fig. 6. The MixColumns stage [5,6]

$$\begin{pmatrix} a'_{0,x} \\ a'_{1,x} \\ a'_{2,x} \\ a'_{3,x} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} a_{0,x} \\ a_{1,x} \\ a_{2,x} \\ a_{3,x} \end{pmatrix} \quad 0 \leqq x < 4;$$

Fig. 7. Column conversion of MixColumns stage [5]

In fact, it invokes an "xtime" function [5,10] whose inputs and outputs are all 1 byte in length, and which left shifts each input for one bit with the least significant bit being filled by a 0. If the input's most significant bit before shift is 1, the shift result will exclusive-or with $\{1b\}_{hex}$.

That means the square matrix on the right hand side of the matrix calculation shown in Fig. 7 is the MixColumus function illustrated in Fig. 6. Hence,

$$a'_{0,0} = (\{02\} \bullet a_{0,0}) \oplus (\{03\} \bullet a_{1,0}) \oplus a_{2,0} \oplus a_{3,0}$$
$$a'_{1,0} = a_{0,0} \oplus (\{02\} \bullet a_{1,0}) \oplus (\{03\} \bullet a_{2,0}) \oplus a_{3,0}$$
$$a'_{2,0} = a_{0,0} \oplus a_{1,0} \oplus (\{02\} \bullet a_{2,0}) \oplus (\{03\} \bullet a_{3,0})$$
$$a'_{3,0} = (\{03\} \bullet a_{0,0}) \oplus a_{1,0} \oplus a_{2,0} \oplus (\{02\} \bullet a_{3,0})$$

in which

$$\{02\} \bullet a_{i,j} = a_{i,j} \bullet \{02\} = \text{xtime}(a_{i,j})$$
$$\{03\} \bullet a_{i,j} = a_{i,j} \bullet (\{01\} \oplus \{02\}) = a_{i,j} \bullet \text{xtime}(a_{i,j})$$

Stage4: AddRoundKey. In this stage, each $a_{i,j}$ in M is exclusive-ored with $k_{i,j}$ where $k_{i,j}$ is an element of a given round sub-key table used to convert $a_{i,j}$ to $a'_{i,j}$, $0 \leqq i, j \leqq 3$. Fig. 8 gives an example. The parent key is used by Rijndael's key schedule [10-12] to generate round sub-keys for each round.

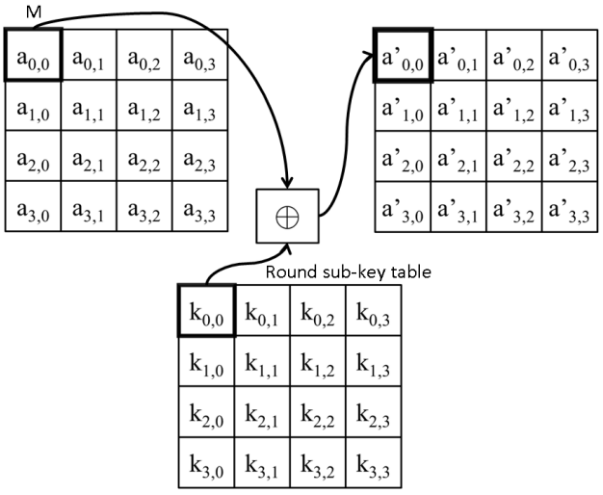

Fig. 8. The AddRoundKey stage [5,6]

2.3 Output Feedback and Cipher Feedback

Output Feedback (OFB for short) [13] and Cipher Feedback(CFB for short) [13] as two commonly used block cipher modes of operation provide feedback mechanisms to resist the plaintext-ciphertext pair statistics attack. They can also invoke other block cipher techniques, e.g., DES and AES, to further improve their security level.

The technical aspects of OFB and CFB are very similar. Both of them need an Initialization Vector together with a key K to trigger a block cipher encryption mechanism. The output of the mechanism, denoted by R, is then XORed with a plaintext block p_i to produce the corresponding ciphertext block c_i, no matter whether OFB or CFB is invoked.

With the OFB, R as shown in Fig. 9 is directly fed back as a key of the next block cipher encryption mechanism. With the CFB, the feedback parameter as shown in Fig.

10 is c_i, rather than R, i.e., the inputs of the CFB include Initialization Vector IV, plaintext p, key K, and ciphertext C where $C=c_1,c_2,c_3\ldots\ldots c_n$.

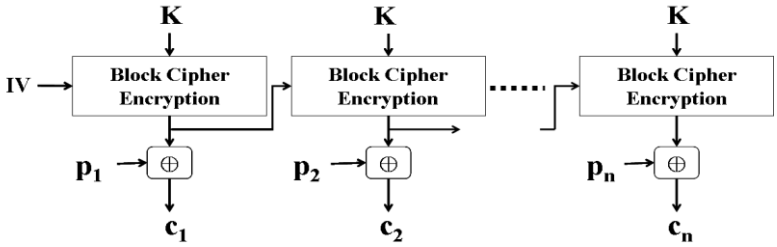

Fig. 9. The OFB mode [13]

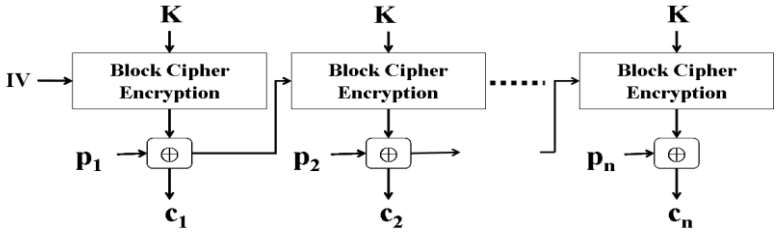

Fig. 10. The CFB mode [13]

3 Feedback Encryption and Three Dimensional Operations

The parameters and functions employed in this study are defined below.
Plaintext：p_i ,$1 \leqq i \leqq n$, (n is the total number of the blocks of the plaintext)
System key：K_i ,$1 \leqq i \leqq 7$
Dynamic key：a_i, b_i, d_i, $1 \leqq i \leqq n$
Dynamic Feedback key：a_{i-1} , b_{i-1} , d_{i-1} , $1 \leqq i \leqq n$
Initial feedback key：$a_0 = K_8$, $b_0=K_9$, $d_0=K_{10}$
Ciphertext block：c_i, $1 \leqq i \leqq n$

Fig. 11 illustrates the FETDO architecture in which before the first round, the values of the system feedback keys (a_{i-1}, b_{i-1}, and d_{i-1}) are all null. That means a_0, b_0 and d_0 require initial values.

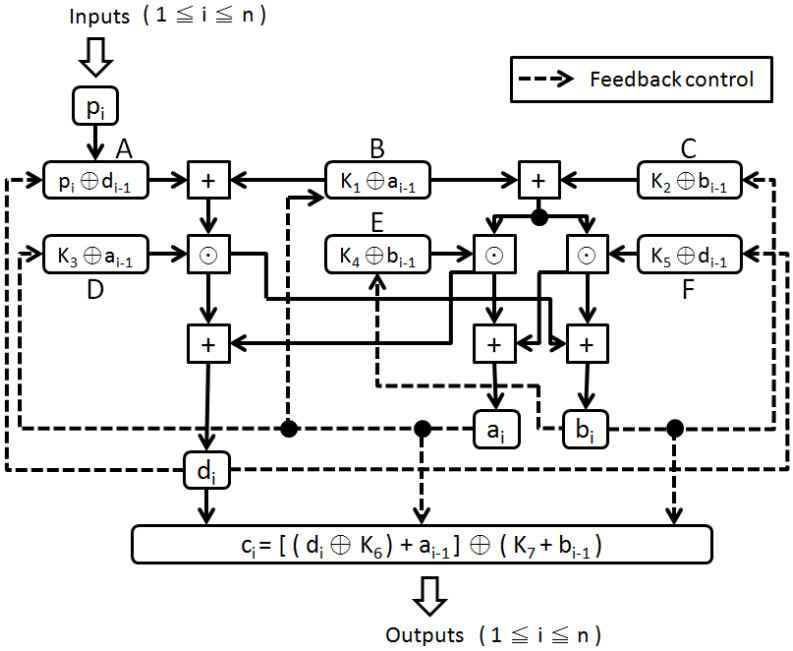

Fig. 11. The architecture of the FETDO Encryption Method

3.1 Encryption

The encryption process of the SeDEM is as follows.

Let ($p_i \oplus d_{i-1}$), ($K_1 \oplus a_{i-1}$), ($K_2 \oplus b_{i-1}$), ($K_3 \oplus a_{i-1}$), ($K_4 \oplus b_{i-1}$) and ($K_5 \oplus d_{i-1}$) are respectively denoted by A, B, C, D, E and F to simplify the expressions of the following equations.

$$d_i = [\ (A+B) \odot D\ ] + [\ (B+C) \odot E\ ] \quad (1)$$

$$a_i = [\ (B+C) \odot E\ ] + [\ (B+C) \odot F\ ] \quad (2)$$

$$b_i = [\ (A+B) \odot D\ ] + [\ (B+C) \odot F\ ] \quad (3)$$

$$c_i = [\ (d_i \oplus K_6) + a_{i-1}\ ] \oplus (K_7 + b_{i-1}) \quad (4)$$

The feedback encryption mechanism has two stages, preparation stage and encryption stage. In the preparation stage, the equations being employed include Eqs.(1)~(3) in which d_i, a_i and b_i are used to encrypt plaintext blocks into ciphertext blocks.

Before the start of the i_{th} encryption iteration, named round i, a_{i-1}, b_{i-1} and d_{i-1} are known or have been calculated in round i-1. Superficially, the complexities of the expressions deriving d_i, a_i, b_i and c_i are high. In fact, the costs of the required operations are lower than those of the DES and AES.

Basically, Eqs.(1), (2) and (3) are produced almost at the same time after A~F are calculated. The total number of operations for deriving A~F is six $\oplus$s (see Fig. 11)). Calculating, a_i, b_i, and d_i needs extra eight operations (i.e., five +s and three $\odot$s, even though the numbers of +s and $\odot$s in Eqs. (1)~(3) are nine and six, respectively, since several $\odot$s and +s in the three equations are the same operations).

After that, the number of operations used to derive c_i is four (two +s and two $\oplus$s). Hence, the total number of operations in generating a_i, b_i, and d_i are eighteen (= 6+8+4, in which there are eight $\oplus$s, seven +s, and three $\odot$s).

3.2 Encryption

To decrypt c_i to p_i, the receiving site needs to first calculate A~F, Eq.(2), and Eq.(3). From Fig. 11, we can see that the number of operations required to generate a_i and b_i is thirteen (i.e., six $\oplus$s, three $\odot$s and four +s, excluding the + operation right above d_i).

Let

$$G = (B+C)\odot E \tag{5}$$

and

$$H = c_i \oplus (K_7 + b_{i-1}) \tag{6}$$

Then

$$d_i = \begin{cases} (H - a_{i-1}) \oplus K_6 & ; \text{if } H \geqq a_{i-1} \\ (H + \bar{a}_{i-1} + 1) \oplus K_6 & ; \text{if } H < a_{i-1} \end{cases} \tag{7}$$

$$p_i = \begin{cases} [(d_i - G)\odot D - B] \oplus d_{i-1}, if\ d_i \geqq G\ and\ (d_i - G)\odot D \geqq B \\ [(d_i - G)\odot D + (\bar{B} + 1)] \oplus d_{i-1}, if\ d_i \geqq G\ and\ (d_i - G)\odot D < B \\ [(d_i + \bar{G} + 1)\odot D - B] \oplus d_{i-1}, if\ d_i < G\ and\ (d_i + \bar{G} + 1)\odot D \geqq B \\ [(d_i + \bar{G} + 1)\odot D + (\bar{B} + 1)] \oplus d_{i-1}, if\ d_i < G\ and\ (d_i + \bar{G} + 1)\odot D < B \end{cases} \tag{8}$$

Here, G can be obtained before acquiring a_i (see Fig. 11). So no extra operations are required. To derive H, two operations, i.e., one $\oplus$ and one +, are needed. When deriving d_i (see Eq.(7)), in worst case i.e., when $H<a_{i-1}$, three operations, i.e., two +s, one $\oplus$, and one judgment are required. On calculating p_i, also in worst case, i.e., when $d_i<G$ and $(d_i+\bar{G}+1)\odot D<B$, two times of judgment and six operations (i.e., four +s, one $\odot$ and one $\oplus$) are needed since $(d_i+\bar{G}+1)\odot D$ in $p_i=((d_i+\bar{G}+1)\odot D+(\bar{B}+1))\oplus d_{i-1}$ can reuse the one calculated in the judgment $(d_i+\bar{G}+1)\odot D<B$, no extra cost is required. But calculating $\bar{G}$, $\bar{B}$ and $\bar{a}_{i-1}$ consumes three $-$s since they are respectively the one's complement of G, B, and a_{i-1}. Namely, in worst case, deriving d_i consumes four operations (rather than three), including two $+$s, one $\oplus$, one $-$, and one judgment, and deriving p_i needs eight operations (rather than six), including four $+$s, one $\odot$, one $\oplus$, two $-$s and two times of judgment.

In summary, to decrypt c_i to p_i, in worst case, we need three judgments and twenty seven (=13+2(for H)+4(for d_i)+8(for p_i)) operations.

4 Safety Analysis and Comparison

A well-designed encryption mechanism must be one with high security level to effectively protect a system from being attacked by hackers, and with high performance and low cost to efficiently perform encryption and decryption [14]. In the following, we will analyze FETDO mechanism and compare it with other cryptographic methods.

4.1 Brute-force and Cryptanalysis

It is very inefficient if someone wishes to solve the FETDO's ten system keys (including seven system keys K_1~K_7, and three dynamic keys a_i, b_i, and d_i) by using a brute force method because the ten keys are not directly generated by the given parent key, and they have $2^{(n \times 10)}$ combinations where n is the key length. The probability of correctly guess their current values on one trial is $1/(2^{(n \times 10)})$ which is approximate to zero, even if n=64.

With the FETDO, current ciphertext is not only a function of current plaintext, but also affected by previous inputs. Its first ciphertext block (c_1) has five unknown variables which are calculated by using two-dimensional operations, i.e., $\oplus$ and +. The computational complexity is high. Therefore, if hackers would like to analyze d_1 from c_1, they have to first solve a_0 and b_0 (see Eq.(4)). Without a_{i-1} and b_{i-1}, they cannot solve the ciphertext blocks c_i, i=1,2,… …n, implying the ciphertext blocks c_1,c_2, c_3, ... …c_n are securely protected.

For security consideration, the ten keys as parameters are built in the developed program so as to significantly reduce the burden of hardware. Hackers cannot crack the keys by using differential cryptanalysis and analyze the key generation process. Further, the three-dimensional operation is a non-linear computation so it is difficult for hackers to solve the operation by using differential and linear cryptanalyses.

4.2 Flexible design on Plaintext Blocks and Keys

CPU processing speeds of recent computers are faster day by day. The 128-bit blocks and 128-bit keys of the AES must be expanded someday. Once they are expanded, the encryption system of the AES has to be redesigned to meet the expansion, e.g., S-box required in the SubBytes stage and the Round sub-key table used in the AddRoundKey stage need to be expanded. But the FETDO still works because the sizes of a key and a block are equal, and can be dynamically adjusted when necessary.

Table 1. Security Analysis.

	DES	AES	FOTDO
Operation structure	Combination logic	Combination logic	Sequential logic
Operator	1	1	3
Round	16	10, 12, 14	1
Key	1	1	10
Block and Key Flexible design	Low	Low	High
Computing Complexity	*Middle*	High	Low
Security level	Low	Middle	High

4.3 Comparison

Table 2. Cost of Encryption / Decryption processes.

	Encryption	Decryption
DES(64-bit block)	32⊕s + 1 IP + 1 IP^{-1} + 128 S-BOX + 16Expansion + 16 Permutation.	The same number of operations as that of the encryption process.
AES(128-bit block, 128-bit key)	176⊕s (AddRoundKey) + 160 Substitutions (SubBytes), + 30 ShiftRows (ShiftRows), + (576(at least 432) ⊕s + 144 time of judgment + 144 ShiftRows) (MixColumns).	The same number of operation as those of the encryption process for AddRoundKey, + SubBytes, and + ShiftRows MixColumns: 116(at least 684)⊕s +432 time of judgment + 432 ShiftRows.
FCTDO	18 = (8 ⊕s + 3 ⊙s + 7 ＋s,)	MAX 27 = (9 ⊕s + 4 ⊙s + 11 ＋s + 3 －s)

Table 1 summarizes the features of the DES, AES and FETDO. In order to improve data delivery security, the AES and DES encryption methods improve their security levels by adding an option of encryption feedback, e.g., the CFB[13,15] mode and OFB[13,15] mode. But due to the following reasons they are still not secure enough.

The feedback values of the ciphertext blocks are relatively easier to be cracked and known. Although the OFB does not expose the feedback value, when the first ciphertext block is cracked, it will face the same problem of the CFB.

The FETDO does not have this problem since it has three internal feedback keys, i.e., a_i, b_i and d_i, and does not expose the feedback values as a part of its output. To achieve this, it uses two keys, i.e., K_6 and K_7, and two-dimensional operators, i.e., $\oplus$ and $+$, to protect the output. Hackers cannot acquire the feedback keys, consequently highly improving its security level for transmitted data.

Table 2 summarizes the cost of the encryption and decryption processes of the three schemes, AES, DES and FETDO. We can see the FETDO outperforms the other two.

5 Conclusion and Future work

In this paper, we discussed and analyzed two encryption algorithms, including DES and AES, which employ combinatorial-logic style monotonic encryption operations, i.e., only keys are used to encrypt plaintext. Hence, their ciphertext is relatively easier to be cracked compared to that of the FETDO. The FETDO solves this problem by using a feedback encryption mechanism to increase the unpredictability of the ciphertext, and a three-dimensional operation and multiple keys to increase the cracking complexity so as to improve its system security level which is higher than those of the AES and DES, and decrease its encryption/decryption cost which is lower than those of the AES and DES.

In fact, the security level of employing multiple keys is similar to that of using lots of one-time-keys [16]. Generally, the encryption mechanism of a security system requires a large number of calculation for the keys exchanged before its data communication begins. The purpose is to increase its security level. However, all the keys as parameters used by our mechanism are built in the developed program, thus consuming a small hardware space to store the key. Today, security technologies advance quickly. Increasing the number of encryption keys (space factor) does not seriously impact the encryption and decryption costs (timing factor). On the contrary, this can exploit low cost and high security, and is suitable for being used by current applications.

When documents need to be securely protected during their delivery, like transmitting military secrets [1] or UIDs and passwords for e-commerce transactions [17], we can distribute the three dynamic keys, a_i, b_i, and d_i, to three key men. Before encrypting documents, the values of the three keys have to be input to the cryptographic system. The user's responsibility is only preparing the documents. But on the receiving end, the three key men have to participate in the decryption. As a result, even if there is a spyware invasion, the documents are still effectively and confidentially protected.

However, the FETDO does not provide fault tolerance and parallel encryption/decryption. If it can provide a non-linear dynamic substitution operation, i.e., a dynamic S-box [18], which provides random S-boxes, i.e., different S-boxes for different rounds, the system will be more secure than it was. Also, we would like to

derive its reliability model so that users can predict the reliability of the system before using it. These constitute our future research.

References

1. Y.L. Huang and F.Y. Leu, "Constructing a Secure Point-to-Point Wireless Environment by Integrating Diffie-Hellman PKDS RSA and Stream Ciphering for Users Known to Each Other," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 2, no. 3, September 2011, pp. 96-107.
2. S.M. Lee and D. S. Kim and J.S. Park, "A Survey and Taxonomy of Lightweight Intrusion Detection Systems," *Journal of Internet Services and Information Security*, vol. 2, issue 1/2, February 2012, pp. 119-131.
3. S.K. Pandey and R. Barua, "Efficient Construction of Identity Based Signcryption Schemes from Identity Based Encryption and Signature Schemes," *Journal of Internet Services and Information Security*, vol. 1, issue 2/3, August 2011, pp. 161-180.
4. http://zh.wikipedia.org/wiki/DES
5. C.H. Yang, Network Security: Theory and Practice, XBOOK MARKETING Co. Ltd., September, 2008.
6. http://en.wikipedia.org/wiki/Advanced_Encryption_Standard
7. J. Hunker and C. W. Probst, "Insiders and insider threats—an overview of definitions and mitigation techniques," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 2, no. 1, March 2011, pp. 4–27.
8. Y.F. Huang, F.Y. Leu, C.H. Chiu and I.L. Lin, "Improving Security Levels of IEEE802.16e Authentication by Involving Diffie-Hellman PKDS," Journal of Universal Computer Science, vol. 17, no.6, March 2011, pp. 891-911.
9. A.P. Moore, D.M. Cappelli, T.C. Carony, E. Shaw, D. Spooner, and R.F. Trzeciak, "A preliminary model of insider theft of intellectual property," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 2, no. 1, March 2011, pp. 28–49.
10. E. Barkan and E. Biham, "In How Many Ways Can You Write Rijndael?" in Proc. ASIACRYPT 2002, ser. Lecture Notes in Computer Science, Y. Zheng, Ed., vol. 2501. Berlin, Germany: Springer-Verlag, Dec. 2002, pp. 160-175.
11. Federal Information Processing Standards Publication 197, "Announcing the ADVANCED ENCRYPTION STANDARD (AES)" November 26, 2001.
12. J. Daemen and V. Rijmen, "AES Proposal : Rijndael," The First Advanced Encryption Standard Candidate Conference, September 1999.
13. http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation
14. T. Eisenbarth, S. Kumar, L. Uhsadel, C. Paar and A. Poschmann, "A Survey of Lightweight-Cryptography Implementations," IEEE Design & Test of Computers, Dec. 2007, pp. 522–533.
15. M. Dworkin, "Recommendation for BlockCipher Modes of OperationMethods and Techniques," Natl. Inst. Stand. Technol. Spec. Publ. 800-38A 2001 ED, 66 pages, December 2001.
16. Mils Electronic, "One Time Key Encryption" http://www.mils.com/
17. F.Y. Yang, Z.W. Liu, and S.H. Chiu, "Mobile Banking Payment System," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 2, no. 3, September 2011, pp. 85-95.
18. S.H. El-Ramly, T. El-Garf and A.H. Soliman, "Dynamic Generation of S-boxes in Block Cipher Systems" Radio Science Conference, August 2002, pp.389-397.