The Design and Implementation of a Symmetric Encryption Algorithm Based on DES

ZhouYingbing
Dept. of Computer Science & Technology
Yanbian University
Yanji, 133002, China
965370611@qq.com

LI Yongzhen*
Dept. of Computer Science & Technology
Yanbian University
Yanji, 133002, China
lyz2008@ybu.edu.cn

Abstract—**In this age of explosive growth in information exchanges, there is indeed no time at which security does not matter. One of the symmetric encryption algorithms, DES, has kept its dominant position in the area of data encryption over the last few decades. However, with a rapid development in the field of computer hardware, DES has already been proved insecure. It takes a short time to translate the ciphertext to its corresponding plaintext using brute-force method at a reasonable cost. This is mainly due to the small key size DES employed. Given these issues, the objective of this article is to suggest an alternative on DES to obtain higher security and better execution efficiency by increasing the key size and updating the iteration technique. Comparisons were conducted with both DES and the advanced DES named triple DES (3DES). The results have demonstrated that the proposed algorithm outperforms both previous algorithms.**

Keywords—DES; key expansion; iterative algorithm

I INTRODUCTION

Symmetric encryption is a form of cryptosystem in which encryption and decryption are performed using the same key. It is widely adopted for its fast and efficient performance. DES (Data Encryption Standard), which was adopted in 1977 by the National Institute of Standards and Technology (NIST), has been the most widely used symmetric cipher [1]. However, coincident with development of computer technology has been the decline in security of DES, as the 56-bit key didn't get extended along with the computer technology. When using the brute-force method (a way to break the cipher by trying every possible key) for decryption, it only takes a few hours at a reasonable cost. In addition, because some of the design criteria of DES were not made public, there has been intense suspicion that it is possible for an opponent who knows the weakness of the system to decrypt.

Given these issues, several alternatives to DES were introduced among which the most two important were AES and triple DES. In this paper, we proposed two new methods for encryption based on the original DES and proved its operational efficiency and security.

II THE PRINCIPLE AND SECURITY OF DES

DES is a symmetric encryption algorithm, it needs two inputs: a plaintext and a key. The length of the plaintext is 64 bits, and the key is also 64 bits in length (only 56 bits are ever used among them, the other 8 bits can be used as parity bits or simply set arbitrarily).

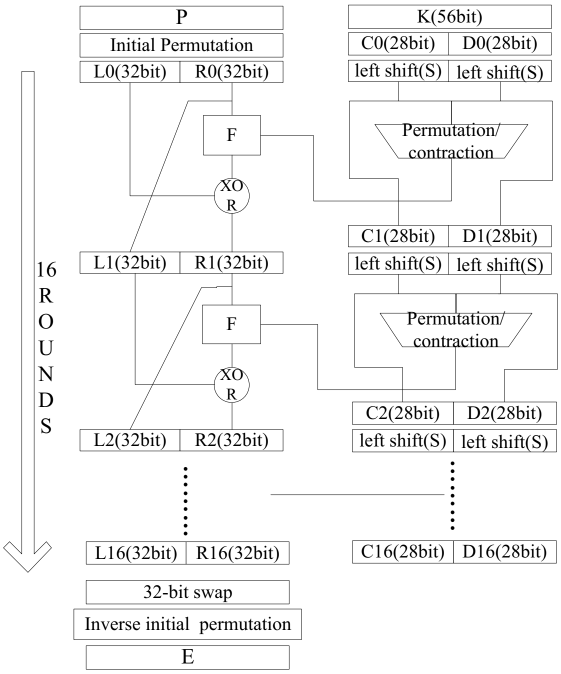

Figure 1. The schematic of the DES algorithm

The overall scheme for DES encryption is illustrated in Figure.1. From the left-hand side of the figure, it can be seen that the processing of the plaintext proceeds in three phases. In the first phase, the 64-bit plaintext passes through an initial permutation that realigns the bits to generate the permuted input. In the second phase, there is sixteen rounds of the same function, including permutation and substitution functions. The left and right halves of the output in each round are swapped to produce the input in the next round. At last, the output in the second phase is passed through a permutation, which is the inverse of the initial permutation function, to produce the 64-bit ciphertext.[2]

The right part of Figure.1 shows how the 56-bit key is used. Initially, the key is passed through a permutation function. Then, a subkey (K_i) is produced in each round by combining with a left circular shift and a permutation. The permutation function is identical for each round, but due to the

[*] Corresponding Author: Li Y.Z. E-mail: lyz2008@ybu.edu.cn

978-1-4799-3279-5 /14/$31.00 ©2014 IEEE

repeated shifts of the key bits, a different subkey is produced. Since the development of DES, it has been subjected to intense criticism over the security that DES provided. Concerns mainly lie in such three areas:

a) *A short key length:* The original DES design was intended to be implemented on single chips. At the time of the design, 56-bit keys were selected because they are suitable for single chips' computational capabilities, and 56-bit keys were enough to withstand brute-force attacks. However, considering today's technological advancement, a 56-bit key length can no longer securely defend against brute-force attacks.

b) *Complementary symmetry:* The complementary symmetry exists in the relationship among the plaintext, ciphertext and the key. It can be explained as follows.

If

$$E(k, m) = C \tag{1}$$

Then

$$E(\bar{k}, \bar{m}) = \bar{C} \tag{2}$$

If we encrypt the plaintext "m" using key "k", we then get the ciphertext "C". $\bar{m}, \bar{k}, \bar{C}$ are the bitwise complements of m, k, and C respectively. It indicates that if we replace m, k, C with their bitwise complements, the equation still holds. Given this property of DES, the time to break the cipher using brute-force method is reduced to half of the original, which further emphasized the necessity to strengthen the construction of DES[3].

c)*Weak keys:* After a permuted choice, the original 64-bit key is reduced to 58 bits. This 58-bit key is then divided into two parts with each of 28 bits. Each part moves independently, processing a circular left shift of 1 or 2 bits to generate the sub-key for each round. If all the bits of each part is 0 or 1, then the resulting sub-keys would be the same to any round. This sort of key is regarded as weak keys. Other weak keys can be referred to[4][5].

To address these issues, many efforts were made to find an alternative to DES; however, years of practice have proven that the internal structure of DES is strong and it's the key length that determinately made DES not applicable any more.

There are in general two ways to find substitutions for DES. One is to design a completely new algorithm, of which AES is a prime example. Another alternative, which would preserve the internal structure of DES, is to make changes on the original DES design.

Currently, changes on DES mainly happen in two areas: one is to extend the key length; the other is to redesign the S-box. However, replacing the original S-boxes may destroy the structure of function F, and may cause unpredictable possibilities for attackers to decipher in ways of differential cryptanalysis [6] and linear cryptanalysis[7]. Therefore, the most effective way to improve the security is to increase the key size which led to the creation of the 3DES. In this paper, by running two single DES processes simultaneously and having the two processes communicated with each other, we actually achieved the extension of the key size. In the simulation results, compared with both 3DES and DES, new methods exhibited prominent performance in terms of both executional efficiency and security.

III Two Improvements on DES

To extend the key length, new encryption methods employed two keys as its input, as in 3DES. Although 3DES has enhanced the security of DES, it redoubled the time complexity and computation complexity at the same time, which would lay heavy burden on the processing units. In this section, two new encryption algorithms, based on DES, that doubled the length of both the plaintext and the key and also preserved the efficiency, are presented.

Extending the key size to acquire higher security would directly lead to the idea of double DES, which simply encrypts the plaintext using two DES projects successively. As described in following formulas.

$$X=E(K_1, P) \tag{3}$$

$$C=E(K_2, X) \tag{4}$$

P is the plaintext; k1 is the first key; through the DES encryption, the interim result X is produced and serves as the input to the next encryption procedure, using the second key k2; C is the final result.

Based on the property of symmetric cipher, that the decryption algorithm is essentially the encryption algorithm run in reverse. In the second phase, the ciphertext C and the secret key K_2 produces the intermediate value X. So there must have a relationship like

$$E(K_1, P)=X=D(K_2, C) \tag{5}$$

This attack is Meet-In-The-Middle Attack[8].

To defend against this type of attack, 3DES applied three stages of encryption/decryption process with two or three keys, which in turn led a marked fall in the operational efficiency. To satisfy the demand of an enough secure and efficient encryption algorithm, we proposed two new methods that employed a fundamentally different mechanism from 3DES. In the new methods, we had two single DES procedures executed at the same time; for the sake of higher security, the two parts communicated with each other after each of the first 15 rounds.

A. System with alternating outputs

The overall process for this new algorithm is illustrated in Figure 2. A 128-bit plaintext is divided into two parts, each of which passes through a single DES encryption process respectively. Communication occurs at the end of each round except for the 16th round. A communication asks for the swap between the two individual DES processes of their right parts. After the completion of the total 16 rounds, the two 64-bit outputs are combined into one 128-bit ciphertext as the final result.

This mechanism will add confusion to the

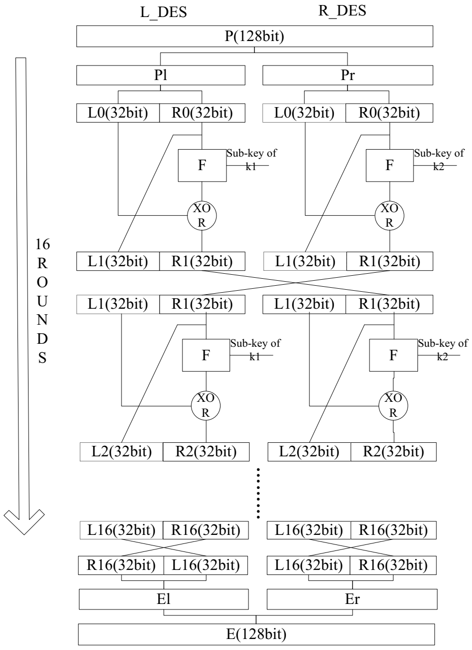

Figure 2. DESnew1

B. System with alternating subkeys

Different from the first method, in this one, we alternate the subkeys served for each round. This will actually supply a single DES procedure with two keys for encryption.

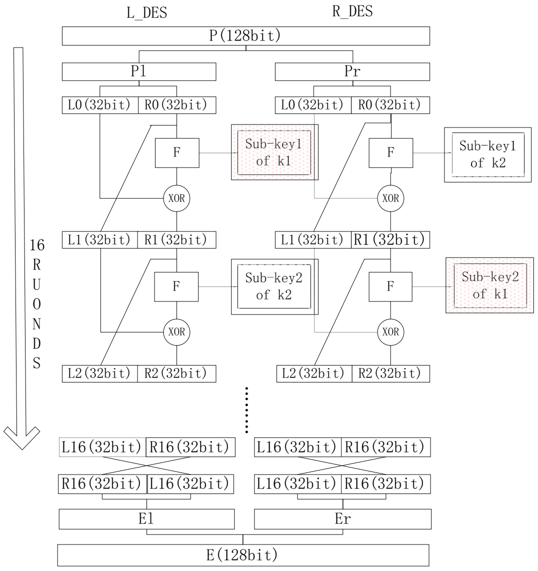

Figure 3. DESnew2

The whole process of this second approach is illustrated in Figure 3. Similar to the first one, the 128-bit plaintext is divided into halves that each passes through a DES process with an input of 56-bit key. As depicted in the figure 2, the subkeys generated from k1 and k2 serve alternately as the input keys for each round. k1 and k2 are initial keys input to the DES procedures; the way of the generation of the subkeys is the same with DES, i.e. taking the permutation function with the initial key to generate k1; the rest subkeys are produced by rotating the previous subkey of 1 or 2 bits. Look at the left-hand side of the figure, the subkey input to the first round is derived from k1; for the second round, the subkey is generated from k2 used in the first round from the right DES process; in the same manner, the subkeys for the rest rounds are deduced. This pattern also works for the right part except for that the subkey begins with k2. From this way, we actually made use of a 112-bit key to encrypt a 64-bit plaintext that made the cryptanalysis become more complicated. The final output is the concatenation of the two 64-bit ciphertext produced by the two parts.

IV EXPERIMENT RESULTS AND ANALYSIS

In experiments, we compared the two new methods with DES and 3DES from the perspective of the operational efficiency. The codes of the methods were programed in java and implemented on eclipse. The encryption target was a file of 10kb. The file was encrypted by the four algorithms of 100 times, 500 times and 1000 times, respectively. We conducted 5 same experiments for each different scale of encryption. The total time of the encryption for each scale are listed in Table Ⅰ.

TABLE Ⅰ THE CONTRAST OF ALGORITHM EFFICENCY

	100rounds（ms）	500rounds(ms）	1000rounds(ms）
DES	687、657、703、688、704	3361、3327、3345、3350、3375	6672、6703、6705、6782、6782
DESNew1	657、640、671、640、672	3187、3156、3172、3172、3141	6344、6328、6312、6280、6296
DESNew2	688、686、704、704、703	3407、3375、3376、3391、3392	6719、6657、6687、6626、6703
3DES	2046 、 2048 、 2063 、 2010、2016	9860、9878、9907、9922、9937	19718 、 19798 、 19829 、 19780、19843

It should be pointed out that, in order to avoid the impact of running the codes irrelevant to the encryption part on the final operational time, such as the file reading part, the total running time was calculated no other than the core codes of the encryption section. In addition, for the sake of executing the algorithms on an equal level, we programed both the new and previous algorithms without invoking API functions.

The following figure displays the average operational time for each scale of encryption.

Figure 4. The contrast of algorithm efficiency

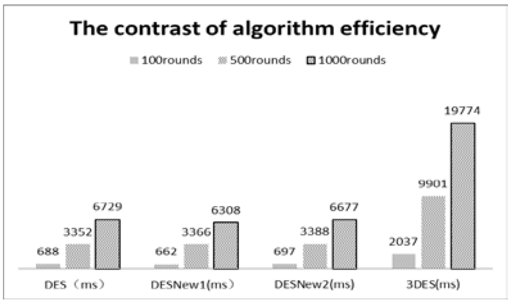

Figure. 4 indicates that the running time of new methods and DES are nearly the same. This is intelligible since the new methods take an execution of two DES process concurrently to produce a 128-bit block that was a doubled length of that produced by DES. In other words, the amount of labor to produce one bit ciphertext is on average the same. In the other comparison, it is clearly seen that new methods take only one third of the running time of 3DES. This is also intuitive that 3DES takes a triple-DES process to produce the ciphertext of the same length with that from new methods, i.e. 3DES triples the labor or the time needed to produce one bit by new methods.

In regard to the security provided by the four methods, the new methods are, at minimum, as secure as 3DES, and they are consequently more secure than DES. This is obvious; since new methods complicated the DES procedure, thus added difficulty to_cryptanalysis. In addition, by letting the two parts in the encryption program communicates with each other, the new methods actually extended the key size; consequently, the attacker would need more time to decipher the messages using brute-force method. When compared with 3DES, as new methods extended the key length to 112 bits (the second method) or more (the first method), they theoretically obtained an equal level of security of 3DES and practically higher operational efficiency than 3DES.

V CONCLUSION

This paper analyzed the weakness exited in DES and proposed two new methods that concurrently operated with two single DES processes. The first method swapped the round results which added confusion to the ciphertext; the second method alternated the subkey between the two parts to obtain an effect of employing a doubled key length (112-bit) to encrypt a 64-bit block. The simulation results indicated that in terms of operational efficiency, new methods distinctly outperformed 3DES and kept a same level with DES. In respect to the security, new methods provided a more secure performance than DES did and at the least as secure as 3DES did.

REFERENCES

[1] FIPS 81 - DES MODES OF OPERATION. http://www.itl.nist.gov/fipspubs/fip81.htm , June 2009.

[2] William Stallings, Cryptography and Network Security: Principles and Practices, Fourth Edition,vol.3. Beijing: Publishing House of Electronic Industry, 2006, pp. 73-87.

[3] Moore, Judy H., and Gustavus J. Simmons, “Cycle Structure of the DES with Weak and Semi-Weak Keys.” In Advances in Cryptology-CRYPTO’ 86: Proceedings, vol. 1, Andrew M. Odlyzko, Eds. Germany: Springer Berlin Heidelberg, 1987, pp. 9-32.

[4] Zhoujianqing and Helingyun. “Key extension of DES encryption algorithm” .Bulletin of Science and Technology, vol. 27, no.2, 2011, pp. 263-267.

[5] Humeiyan and Liuranhui. “DES algorithm security Analysis and Research”. Journal of Inner Mongolia University. Inner Mongolia, vol. 36, pp. 693-697, 2005.

[6] Preneel, Bart, Christof Paar, and Jan Pelzl. Understanding Cryptography:A Textbook for Students and Practioners. Beijing: Springer Press,vol. 3, Sept 2012, pp. 55-75.

[7] Mitsuru Matsui. “Linear Cryptanalysis Method for DES Cipher.” in Advances in Cryptology — EUROCRYPT ’93, vol. 765, Tor Helleseth Eds, Germany: Springer Berlin Heidelberg, May 1993, pp. 386-397.

[8] Diffie W and Hellman M.E, “Exhaustive Cryptanalysis of the NBS Data Encryption Standard,” Computer. vol. 10, no. 7, pp. 74-84, June 1977.