A New Symmetric Key Cryptosystem Based on Feistel Network: Parallel Dependent Feistel Network (PDFN)

Indrajit Das[1] and R. Saravanan[2]

[1] SCSE, VIT University, Vellore-632014, India
[2] SITE, VIT University, Vellore 632014, India
indrajit.das23@gmail.com, rsaravanan@vit.ac.in

Abstract. In this paper a new Symmetric Key Cryptosystem, based on Feistel Networks has been proposed. The Cryptosystem demonstrates a few effective features like variable size key, variable size plain text encryption depending on data and key, padding by random variables etc. The cryptosystem is designed to be efficient on processors using simple operations.

Keywords: Symmetric Key Cryptography, Encryption, Decryption, Feistel Networks, Avalanche Effect.

1 Introduction

Secure communication and secure data storage is a necessary part of our everyday lives. A few examples are ecommerce, phone calls, transfer or storage of proprietary information, email, secure money transfers and military applications. It has all become possible because of the evolution of cryptography over the past couple of decades. Originally cryptography dealt with the science of secret writing, whence its name is derived. The first major step in the development of modern cryptography was Shannon's concept of perfect secrecy based upon an information theoretic model [1].

In this paper a new secret key cryptosystem is presented. The major points taken into consideration while designing the algorithm were

- It should guarantee substitution and permutation for all bits in each round.
- It should guarantee some randomness which will cause an unpredictable change on every encryption.
- It should be able to encrypt a block of variable plaintext. The size of the plain text to be encrypted should be determined on a dynamic basis.
- It should use a variable size key.
- It should be a word-oriented algorithm.
- It should use operations which are comparatively efficient on processors.

1.1 Symmetric Key Cryptography

Symmetric-key algorithms are a class of algorithms for cryptography that use identical cryptographic keys for both encryption and decryption and therefore

N. Meghanathan et al. (Eds.): CCSIT 2012, Part I, LNICST 84, pp. 592–601, 2012.
© Institute for Computer Sciences, Social Informatics and Telecommunications Engineering 2012

sometimes called *secret-key cryptography.* In contrast *public-key algorithms*, uses two keys - a public key to encrypt and a private key to decrypt messages. Symmetric Key Cryptosystems consists of the following five ingredients:

- *Plaintext*: This is the original intelligible message or data that is fed to the algorithm as input.
- *Secret Key*: The secret key is also input to the encryption algorithm. The exact substitutions and permutations performed depend on the key used, and the algorithm will produce a different output depending on the specific key being used at the time.
- *Encryption Algorithm*: The encryption algorithm performs various substitutions and permutations on the plaintext.
    - Input: (M, K)
    - $C \leftarrow Enc_K(M)$
    - Output: (C)
- *Cipher text*: This is the scrambled message produced as output. It depends on the plaintext and the key. The cipher text is an apparently random stream of data, as it stands, is unintelligible.
- *Decryption Algorithm*: This is essentially the encryption algorithm run in reverse. It takes the cipher text and the secret key and produces the original plaintext.
    - Input: (C, K)
    - $M \leftarrow Dec_K(C)$
    - Output: (M)

The main advantages of symmetric-key cryptography are

- High speed of data encryption.
- Shorter keys compared to public key algorithms.
- Symmetric-key ciphers can be used as primitives to construct various cryptographic mechanisms (i.e. pseudorandom number generators).
- Ciphers can be combined to produce stronger ciphers.
- Symmetric-key encryption is perceived to have an extensive history.

1.2 Feistel Network

A Feistel network is a general method of transforming any function into a permutation. It was invented by Horst Feistel and has been used in many block cipher designs [2, 3]. The most important part of a Feistel Network is a non-linear and irreversible function F. The function F of a general Feistel Network can be expressed as

$$F: \{0, 1\}^{n/2} * \{0, 1\}^{k} \rightarrow \{0, 1\}^{n/2}$$

Where n is the size of the block, F is the function taking n/2 bits and k bits of key of key as input and producing an output of length n/2 bits. One round of a general Feistel network can be described as

$$X_{i+1} = (F_{ki}\ (msb_{n/2}(X_i))\ XOR\ lsb_{n/2}(X_i)) \parallel msb_{n/2}(X_i)$$

Here X_i is the input to the round and X_{i+1} is the output. K_i is the key, n is the block length, lsb_u and msb_u are the least and most significant u bits of the block X respectively, XOR indicates modulo 2 addition and || indicates concatenation.

Diagram of a general Feistel round in given in Figure 1. The working of a Feistel Network is given below:

- Split each block into halves
- Left half becomes new right half
- New left half is the final result when the right half is XORed with the result of applying F to the Left half and the key.

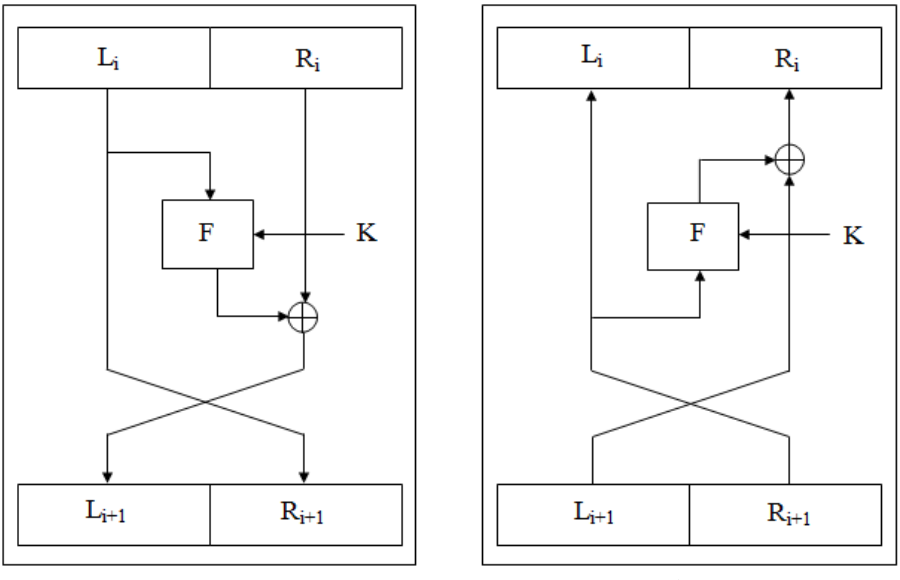

Fig. 1. a) Feistel Encryption Round b) Feistel Decryption Round

Let F be the round function and let k_0, k_1, k_2….k_n be the sub-keys for the rounds r_0, r_1, r_2 … $r_n$respectively. Then the basic operation is as follows:

Split the plaintext block into two equal pieces (L, R)

For each round r_i where i= 0, 1, 2… n compute

$R_{i+1} = L_i$

$L_{i+1} = R_i$ XOR F (L_i, K_i)

The cipher text is (L_{n+1}, R_{n+1}).

Decryption of a cipher text (L_{n+1}, R_{n+1}) is done by computing for round r_i where i=n, n-1, n-2,…0

$L_i = R_{i+1}$

$R_i = L_{i+1}$ XOR F (R_{i+1}, Ki)

Then (L_0, R_0) is the plaintext.

1.3 S-Box Design

The substitution layer in a Substitution-Permutation Network (SPN) is of critical importance to security since it is the primary source of nonlinearity in the algorithm (permutation is a linear mapping from input to output). S-Box substitutes the input with the output such that any change to the input results in a random-looking change to the output. Research into cipher design and analysis [4, 5, 6] suggests that s-boxes with specific properties are of great importance to avoid certain classes of attacks. However, it can be very difficult (and, in some cases, impossible) to satisfy some of these properties using "small" s-boxes.

Four general approaches can be made [12]

- **Choose Randomly.** It is clear that small random S-boxes are insecure, but large random S-Box may be good.
- **Choose and Test.** Some ciphers generate S-Boxes and then test them for the required properties.
- **Man Made.** This technique uses little mathematics: S-Boxes are generated using more intuitive techniques.
- **Math made.** Generate S-Boxes according to mathematical principles so that they have proven security against linear and differential cryptanalysis and good diffusive property.

2 Proposed Design: PDFN

The name “Parallel Dependent Feistel Network” is given to the algorithm since the algorithm can be viewed as two or three Feistel Networks working in parallel and are dependent on the each other. The proposed design PDFN encrypts variable size plain text and has variable length key and data dependent circular shift operation. The proposed design PDFN encrypts 128 bit block.

In the block size of 128 bits, the 96 higher order bits are plain text bits and the remaining 32 bits is a combination of plain text bits and random bits selected by the sender. This feature increases the security of the algorithm since the attacker, despite of the knowledge of the algorithm, is not able to predict the actual size of the plain text being encrypted. The size of the plain text is determined by the 96 higher order bits and the key. We calculate N where N= (96 plain text bits + Key) MOD 32. In the next 32 bits of the block, N bits are plaintext bits and rest 32 – N bits are random bits depending on the encoder. The same calculation is done by the decoder and the last 32-N bits are discarded after decryption.

It can be observed that during encryption, every time the algorithm encrypts a block of different size and the random bits used for padding results in a random change in the value of data in every round. When the algorithm is used in Cipher

Block Chaining mode (CBC) the Initialization Vector (IV) can be used to influence the value of N.

There are total 14rounds and each round uses 5 sub keys hence 70 sub keys each for both encryption and decryption. The large number of sub keys increases the security of the algorithm. The advised minimum length of key is 320 bits and the upper limit is 2240 bits which is very large making it unrealistic for a brute force attack.

The keys are calculated before encryption or decryption.

- Each Sub Keys is of 32 bit and are named as:
  SK0, SK1, SK2, SK3,……, SK69
- There are four 32-bit S-boxes with 256 entries each:
  $S_{1,0}$, $S_{1,1}$,$S_{1,2}$,....., $S_{1,255}$
  $S_{2,0}$, $S_{2,1}$,$S_{2,2}$,….., $S_{2,255}$
  $S_{3,0}$, $S_{3,1}$,$S_{3,2}$,....., $S_{3,255}$
  $S_{4,0}$, $S_{4,1}$,$S_{4,2}$,….., $S_{4,255}$

2.1 Encryption Process

For encryption there are 14 identical rounds. Each round divides the 128 bit block into four 32 bit words which are XORed with the corresponding sub key. Each of these words then undergoes a cyclic shift operation where the number of shifts is governed by the data and the sub key. The next part of the encryption process is a Feistel Network [2, 3]. The algorithm for encryption round is as follows

```
Encryption_PDFN (128 bit data block X)      /*Round i*/
{
128 bit data block B is divided into four 32 bit blocks: A, B, C and D
A = A XOR SK5i+0
B = B XOR SK5i+1                            // round i
C = C XOR SK5i+2
D = D XOR SK5i+3
A = A << (B XOR SK5i+4)                     //circular shift operations
B = B << (C XOR SK5i+4)
C = C << (D XOR SK5i+4)
D = D << (A XOR SK5i+4)
A = A XOR F (B)
B = B XOR F (C)
C= C XOR F (D)
TEMP = A
A = B
B = D
D = C
C = TEMP
Return X
}
```

The block diagram for encryption is given in figure 2.

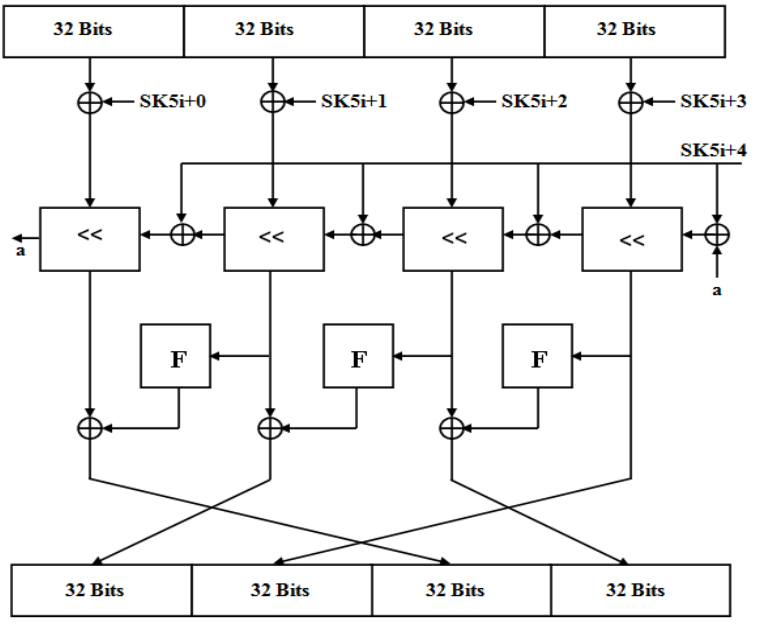

Fig. 2. Block diagram of PDFN Encryption

2.2 Decryption Process

For decryption there are 14 identical rounds. The algorithm for decryption round is as follows

Decryption_PDFN (128 bit data block X) /*Round i*/
{
128 bit data block B is divided into four 32 bit blocks: A, B, C and D
TEMP=D
D=B
B=A
A=C
C=TEMP
C=C XOR F (D)
B=B XOR F (C)
A=A XOR F (B)
D = D >> (A XOR SK5i+4)
C = C >> (D XOR SK5i+4)
B = B >> (C XOR SK5i+4)
A = A >> (B XOR SK5i+4)
A = A XOR SK5i+0
B=B XOR SK5i+1

C=C XOR SK5i+2
D=D XOR SK5i+3
Return X
}

The block diagram for encryption is given in figure 3

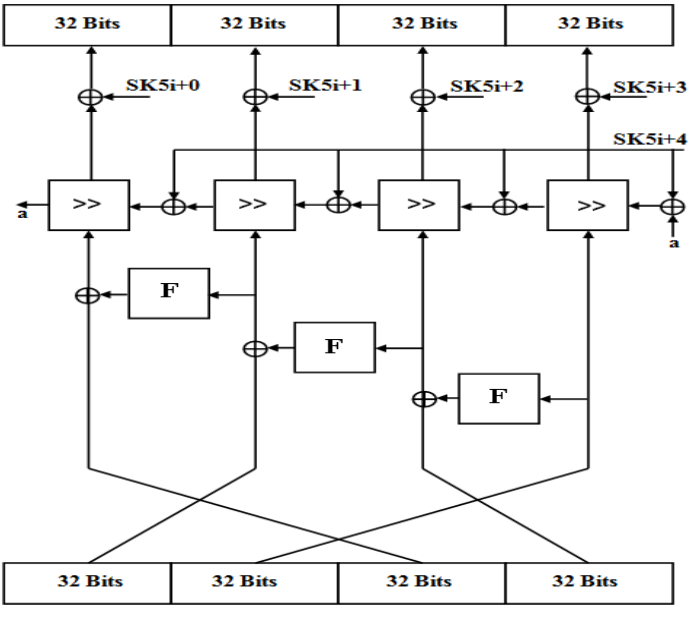

Fig. 3. Block diagram of PDFN Decryption

2.3 Function F

The most important part of a Feistel Cipher is the non-reversible function. This function uses simple operations like XOR and Modular Addition. The algorithm for function F is given below

Function_PDFN (32 bit block)
{
32 bit block is divided into four 8 bit blocks: A, B, C and D
Val_A= S_BOX_1 [A]
Val_B= S_BOX_2 [B]
Val_C= S_BOX_3[C]
Val_D= S_BOX_4 [D]
Val_AB= Val_A XOR Val_B
Val_CD= Val_C XOR Val_D
Val_F = (Val_AB+Val_CD) mod 32
Return Val_F
}

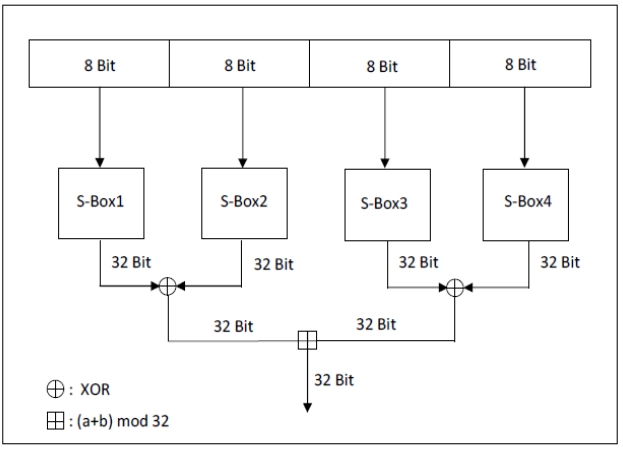

Fig. 4. Block diagram of PDFN function F

2.4 Sub Key and S-Box Generation

Sub Key and S-Box generation play a very important role providing high security to a symmetric key cryptosystem. The key schedule should satisfy completely or partially Strict Avalanche Criterion (SAC) [3, 6] and Bit Independence Criterion (BIC) [3, 6] in order to avoid certain key clustering attacks. Grossman and Tuckerman [7] showed that cryptosystems like the Data Encryption Standard (DES) where the key does not vary with successive rounds can be broken. It is therefore required that the primary key bits used to create sub key for a particular round *i* are different from those used in round i+1. PDFN uses 70 Sub Keys and four S-Boxes of dimension [1x256] of 32 bit each for encryption and decryption. The algorithm for Sub Key and S-box is as follows

SK_SBox_PDFN (variable size key)
{

Step1. Initialize each element of the four [1x256] S_Box, 70 Sub keys SK and 128 bit block B i.e. 4 words in order with decimal values of e^{π}.

Step2. Initialize array X of size 1098 words with decimal values of exponential e.

Step3. Initialize array Y of size 1098 words with decimal values of pi.

Step4. Initialize array SEQ of size 35136 bits or 1098 words by using key as a trajectory to select bits from X and Y. Repeat the key if necessary. Exp: If key bit is 0 select bit from X, if 1 select from Y. There are two approaches while selecting bits from X and Y which are described later.

Step5. XOR every element of S-Box, Sub key and B with SEQ.

Step6. XOR consecutive word (32 bits) of key with consecutive sub key i.e. XOR first 32 bits of key with the first sub key, second 32 bits of key with second sub key and so on. Repeat key if necessary.

Step7. Encrypt block B with same algorithm and sub keys to get 128 bit cipher text i.e. 4 words.XOR the 4 cipher text words with S-Box and repeat step 6 using 128 bit cipher text block B until all the elements of S-Box and Sub-Key have been XORed.

}

Selection of bits from array X and Y as stated above, can be done in two ways which differ in speed and security of the algorithm.

- **FixedBitPosition:** In this approach, the i^{th} bit of trajectory Key determines the selection between j^{th} bits of array X and Y where j= n *length (key) + i and n is the number of times the key is repeated. This selection process can be executed effectively using parallelism but has a weakness - if the bit is 0 or 1 in both X and Y then the value of bit in SEQ is predetermined.
- **NextBitPosition:** In this approach, the i^{th} bit of the trajectory SEQ determines the selection between the $p+1^{th}$ and $q+1^{th}$ bits from array X and Y where p + q = i and p, q are the bit positions from where the last bit was selected from X and Y respectively. This selection process is sequential but distributes the entropy of the key throughout SEQ.

3 Analysis

- Parallel dependent Feistel network is designed in such a way that every higher order 32 bit gets affected by lower order 32 bits and the sub keys thus increasing the security of PDFN.
- The algorithm encrypts variable plain text size on every encryption even if the key is same because the size of plain text to be encrypted depends upon both key and plain text.
- Data and Key dependent cyclic shift operations makes the algorithm more secured.
- Function F is non-reversible in nature which gives PDFN quality avalanche effect.
- Use of four different S-boxes avoids symmetries when input bytes are equal.
- The key-dependent S-boxes protect against differential and linear cryptanalysis. The structure of the S-boxes is unknown to the attacker. Key-dependent S-boxes are easier to implement and can be created on demand, reducing the need for large data structures stored with the algorithm.
- The sub key generation process is designed in such a way that all the sub keys are affected by the key. One major advantage of the sub key generation algorithm is that parallel execution can be done for Step 4, Step 5 and Step6.
- If compared to Blowfish, PDFN uses key dependent string to encrypt and finalize the S-Boxes and sub key.
- Padding is done by random values known only to the sender which causes random change in the data which adds to the strength of the algorithm.

4 Future Work

- In proposed system the Key Generation algorithm can be modified for achieving more parallelism.
- The algorithm could be modified for variable number of rounds.
- The function F could be modified for better speed and security.

5 Conclusion

In this paper a new cryptosystem based on Feistel network has been proposed. The algorithm exploits Feistel structure to provide better security. Its variable size key makes the key space very large and hence brute force attack is not possible. Variable block size makes it difficult for the attacker to guess the size of the plaintext. Padding by random variables causes an undeterministic change in data in all rounds of encryption. Its testing against Differential and Linear attacks is yet to be done.

References

[1] Shannon, C.E.: Communication theory of secret systems. Bell Systems Technical Journal 28, 656–715 (1949)
[2] Feistel, H.: Cryptography and Computer Privacy. Scientific American 228(5), 15–23 (1973)
[3] Feistel, H., Notz, H.W., Lynn Smith, J.: Some cryptographic techniques for machine-to-machine data communications. IEEE Proceedings 63(11), 1545–1554 (1975)
[4] Webster, A.F., Tavares, S.: On the design of S-Boxes. In: Williams, H.C. (ed.) CRYPTO 1985. LNCS, vol. 218, pp. 523–534. Springer, Heidelberg (1986)
[5] Evertse, J.-H.: Linear Structures in Blockciphers. In: Price, W.L., Chaum, D. (eds.) EUROCRYPT 1987. LNCS, vol. 304, pp. 249–266. Springer, Heidelberg (1988)
[6] Youssef, A., Tavares, S., Mister, S., Adams, C.: Linear Approximation of Injective S-boxes. IEEE Electronics Letters 31(25), 2168–2169
[7] Grossman, E., Tuckerman, B.: Analysis of a Feistel-like cipher weakened by having no rotating key, Technical Report RC 6375, IBM (1977)
[8] Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1991)
[9] Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
[10] Rivest, R.L.: The RC5 Encryption Algorithm
[11] Schneier, B.: Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)
[12] Schneier, B.: Applied Cryptography. John Wiley & Sons, New York (1994)