CHAPTER 3

ANALYSIS ON BLOCK CIPHER

3.1 Introduction

A block cipher is a symmetric encryption scheme that breaks up the plaintext messages to be transmitted into blocks of a fixed size, and encrypts one block at a time. Block cipher has formed the basis for many cryptosystem in the past. This chapter analyses the necessary principles and requirements to design a secure block cipher.

This chapter is organized as follows. We analyze various components that form a block cipher and their important properties. S-Box, the fundamental component that forms the nonlinear part of a block cipher, plays an important role in providing confusion to the block cipher. We investigate various cryptographic properties of the S-Box necessary to resist various kinds of cryptanalysis and statistical attacks. This is followed by a discussion of the diffusive element in block cipher that spreads the influence of individual plaintext or key bit over the whole ciphertext. We analyze two different diffusion techniques to find out their diffusive property. Key scheduling, another important component, which produces round subkeys for each iterative rounds is examined. Finally, we analyze the common block cipher structures and their effect on the block cipher.

3.2 Substitution Layer

Shannon's principle of confusion and diffusion is the cornerstone of good block cipher design. A block cipher is a substitution of plaintext value into arbitrary ciphertext value. Since a realistic block cipher has a block width that is far too large to hold the transformation in any physical table, it is necessary to use components that are far smaller than the desired block size in the construction of block ciphers. This makes it necessary to mix the data from each part so that changes in any part will produce the same statistical effect in the overall block. The substitution layer provides confusion and diffusion in each smaller parts of the block ciphers. The diffusion layer provides the overall diffusion needed in wide block cipher [3].

A substitution-box (S-Box) is a table-driven non-linear substitution operation used in most block ciphers. It is simply a substitution that maps m-bit inputs to n-bit outputs. S-Boxes are generally the only nonlinear step in a block cipher algorithm. Its main purpose is to provide confusion to obscure the relationship between the plaintext, the ciphertext and the key. It is the fundamental element that gives a block cipher its security.

The following paragraphs discuss some important cryptographic properties of an S-Box and various types of S-Boxes. There are various types of S-Boxes, such as man-made S-Box, algorithmic S-Box and random S-Box. Each S-Box has its own cryptographic properties. Since the strength of a block cipher ties directly to its S-Boxes, an in-depth study of the important cryptographic properties of an S-Box is crucial to the security of a block cipher.

3.2.1 S-Box Properties

Nonlinear S-Boxes provide resistance against linear and differential cryptanalysis. Since S-Boxes are the most critical parts in a block cipher that provide security of the block cipher, the criteria of S-Boxes must be studied. Some important cryptographic properties of S-Boxes are examined in this section.

1. Completeness

The property of completeness for a bijective function f: $\{0, 1\}^n \rightarrow \{0, 1\}^n$, was first defined by Kam and Davida [24]. f is complete if for all i, j γ $\{1, 2, \ldots, n\}$, there exists two n-bit vectors X_1, X_2 such that X_1 and X_2 differ only in the i^{th} bit and $f(X_1)$ and $f(X_2)$ differ in at least the j^{th} bit. This means that every output bit is a function of all input bits. An S-Box is complete if it satisfies this property. An SP network is complete if for every possible key value, every output bit c_i of the SP network depends upon all input bits $p_1,\ldots,p_n$ and not just a proper subset of the input bits.

2. Avalanche Criterion

Feistel defined a property of S-Boxes and SP networks known as avalanche criterion [25]. A function f : $\{0, 1\}^n \rightarrow \{0, 1\}^n$ satisfies the avalanche criterion if a one bit input change causes each output bit to change with a probability of ½. This means that an SP network satisfies the avalanche criterion if whenever one plaintext bit or one key bit is changed, on average half the ciphertext bits change.

3. Strict Avalanche Criterion (SAC)

In 1985, Webster and Tavares combined the concepts of completeness and the avalanche effect to define a new property, known as the strict avalanche criterion [26]. A function f: $\{0, 1\}^n \rightarrow \{0, 1\}^n$ satisfies SAC if for all i, j γ $\{1, 2, \ldots, n\}$, flipping input bit i changes output bit j with probability exactly ½. This means that if a cryptographic function is to satisfy the strict avalanche criterion, then each output bit should change with a probability of one half whenever a single input bit is complemented. A more precise definition of the criterion is as follows. Consider X and X_i, two n-bit binary plaintext vectors, such that X and X_i differ only in bit i, 1 T i T n. Let

$$V_i = f(Y) \oplus f(Y_i)$$

where $Y = f(X)$, $Y_i = f(X_i)$, and f is the cryptographic transformation under consideration. If f is to meet the strict avalanche criterion, the probability that each bit in V_i is equal to 1 should be one half over the set of all possible plaintext vectors X and X_i. This should be true for all values of i.

4. Nonlinearity

An important cryptographic property for block cipher is nonlinearity. Since the S-Boxes are the only nonlinear elements of an SP network, the level of nonlinearity of S-Boxes is crucial in determining the security of the overall SP network. S-Boxes with high nonlinearity are needed to make an SP network immune to linear cryptanalysis. An S-Box has high linearity if each output bit has low correlation to a linear system equation [27]. Linear cryptanalysis attempts to find a linear approximation that is deduced by combining a number of probable linear expressions of the involved S-Boxes.

Meier and Stafflebach introduced perfect nonlinearity. They discussed nonlinearity criteria for Boolean functions, which are defined to be at maximum distance from linear structures [28]. These functions are the same as the previously known bent functions. To construct perfect nonlinear S-boxes, it is necessary that each output bit is a perfect nonlinear function of the input.

5. Low Probability Differential Characteristics

Differential cryptanalysis [29] is a powerful cryptanalysis technique introduced by Biham and Shamir. The differential attacks exploit the property that the XOR of two inputs to the F function leads to a non-uniform distribution of the XOR of the output of the corresponding outputs. Differential cryptanalysis requires knowledge of the XOR tables of the S-Boxes. Differential cryptanalysis exploits input/output XOR pairs with large table entries. An SP network can be secured against differential cryptanalysis by selecting S-Boxes with low XOR table entries, ideally all 0 or 2, with the exception of entry (0, 0), which has value 2^n.

6. Cyclic Properties

For an invertible n x n S-Box, there are two important cyclic properties to be considered:

i) Number of fixed points

ii) Number of cycles

The number of fixed points of S is defined as the number of $X \gamma \{0, 1\}^n$ such that $S(X) = X$. A cycle is a sequence of elements in $\{0, 1\}^n$, $X_1, X_2, \ldots, X_L$, such that $X_{l+1} = S(X_l)$ for $1 \leq l \leq (L-1)$, and $X_1 = S(X_L)$, but $X_1 \neq S(X_l)$ for $2 \leq l \leq (L-1)$. The length of the cycle is L.

The cyclic properties of an S-Box are related to its cryptographic properties. Youssef et. al. shows the experimental results for the average nonlinearity and the average maximum XOR table entry as a function of the number of fixed points [30]. The results clearly indicate a strong correlation between the cryptographic properties and the number of fixed points and suggest that the S-Boxes should be chosen to contain few fixed points.

7. Bit Independence

Bit independence criterion (BIC), defined by Webster and Tavares, is another property desirable for any cryptographic transformations besides SAC [31]. A function f: $\{0, 1\}^n \rightarrow \{0, 1\}^n$ satisfies BIC if for all i, j, k γ $\{1, 2, \ldots, n\}$ with $j \neq k$, inverting input bit i causes output bits j and k to change independently. In other words, for a given set of avalanche vectors generated by the complementing of a single plaintext bit, all the avalanche variables should be pair wise independent. The degree of independence between a pair of avalanche variables can be measured by calculating their correlation coefficient. A correlation coefficient of 0 means that the variables are independent.

If either the strict avalanche criterion or the bit independence criterion is not satisfied, then a cryptanalyst can gain some information about the statistical properties of the function, which he could conceivably use to his advantage in an attack on the system.

8. Static Information Theoretic Properties

This property specifies that partial knowledge of plaintext or key bits should not reveal any information about the ciphertext [32].

9. Dynamic Information Theoretic Properties

This property specifies that partial knowledge of plaintext or key bits changes should not reveal any information about the ciphertext bit changes [32].

Of all the properties of an S-Box specified, there are some properties that are very important to the security of a block cipher. Analysis of the security of the block cipher can be based on these properties. Completeness, avalanche criterion and strict avalanche criterion are very important to ensure the diffusion of the block cipher. Although the diffusion layer is the main layer that provides diffusion to the block cipher, the S-Box plays an important role in contributing to the overall diffusion of the block cipher. The diffusion layer is responsible for transposing the outputs received from all S-Boxes. It would normally take several rounds to achieve the desired effect. The number of rounds required can be significantly reduced if the outputs of the S-Boxes are highly diffusive.

Other properties such as nonlinearity, cyclic properties and low probability differential characteristics are required to provide confusion for a block cipher. These properties are very critical to the overall security of the block cipher. To provide immunity against differential cryptanalysis, differential distribution table for each S-Box should be uniform to smooth out differentials in any particular round. To be resistant against linear cryptanalysis, Boolean functions used in S-Box should not be linear or affine. There should be no correlations between input and output bits.

3.2.2 Types of S-Boxes

The design of an S-Box is the most critical part in the process of designing a block cipher since the strength of a block cipher ties directly to its S-Boxes. The analysis on the important cryptographic properties of an S-Box in the previous section gives a guideline in the requirements of a secure block cipher. There are many ways to design an S-Box. One can design based on mathematical rules or he can construct an S-Box intuitively based on ad-hoc criteria. Since choosing good S-

Boxes is crucial to the security of the block cipher, some of the common types of S-Boxes used in various block ciphers are investigated.

a) Man-made S-Boxes

Man-made S-Boxes use little mathematics. S-Boxes are generated using more intuitive techniques and tested for the required properties. These types of S-Boxes are specified as lookup tables. Man-made S-Boxes have the advantage that there is no mathematical structure that can be used for cryptanalysis.

b) Algorithmic S-Boxes

The S-Boxes are derived algebraically according to mathematical principles. The S-Boxes can be generated with given cryptographic properties. They can be generated such that they have proven security against linear and differential cryptanalysis and good diffusive properties. Algorithmic S-Boxes have the advantage that they can be implemented easily in applications with low memory requirement compared to S-Boxes using large tables. Examples of block cipher using algorithmic S-Boxes are IDEA, SAFER, SQUARE, RIJNDAEL and LOKI.

c) Random S-Boxes

The S-Boxes are generated randomly. Randomly constructed S-Boxes are unlikely to be secure and can be vulnerable to differential cryptanalysis. Small random S-Boxes are insecure, but the security of random S-Boxes can be improved by using larger inputs. Another method to improve the security of random S-Boxes is to make the S-Boxes key-dependent. Examples of block cipher using random S-Boxes are Khafre, CMEA and NewDES.

d) Key-dependent S-Boxes

S-Boxes are either fixed or key-dependent. There are three general approaches in constructing key-dependent S-Boxes:

(i) Construct the S-Box randomly, e.g. Blowfish and REDOC-II.

(ii) Construct the S-Box specifically such that no two entries are identical, e.g. Khufu and WAKE.

(iii) Construct the key-dependent S-Boxes from secure fixed S-Boxes and a series of strict mathematical rules, e.g. Biham-DES.

The fact that S-Boxes are unknown to the cryptanalyst is one of the principle strengths of key-dependent S-Boxes since this provides protection against unknown cryptanalysis.

There is no one S-Box that is superior to others. Each has advantages and disadvantages. Algorithmic S-Box has the advantage of being optimal against known attacks, such as linear and differential cryptanalysis. However, it may be vulnerable to unknown attacks. Random S-Box, on the other hand, may not be optimal against known attacks, but is likely to be more optimal against unknown attacks compared to algorithmic S-Box. To make random S-Box more resistant against known attacks, it can be made sufficiently large. In terms of memory requirement, algorithmic S-Box has the advantage that it can be implemented easily in applications with low memory requirement if compared to S-Box using large tables.

3.3 Diffusion Layer

In the previous section, we analyze various important cryptographic properties of the substitution layer to provide confusion and diffusion in each smaller part of a block ciphers. The next step is to mix the data from each part so that changes in any part will produce the same statistical effect in the overall block. The diffusion layer provides the overall diffusion needed in wide block cipher.

Diffusion can be achieved by permutation or linear transformation. The diffusion layer should provide an avalanche effect such that small input changes should cause large output changes, and conversely, to produce a small output change, a large input change should be necessary. Permutation is used by DES to allow one bit to affect two substitutions. Hence, the dependency of the output bits on the input bits spreads faster. While bit permutation operates efficiently on hardware platform, it is not optimized for software implementation. Linear transformation provides an alternative to achieve diffusion in a block cipher. Linear transformation consists of linear operations that spread the output of one S-Box to other S-Boxes. Unlike bit permutation, which is bit-oriented, linear transformation can be designed such that it is byte-oriented or word-oriented. This makes linear transformation well suited for software as well as hardware implementations. Linear transformation is used in several well-known block ciphers. SQUARE uses maximum distance separable error correcting codes (MDS codes) to guarantee a good diffusion while SAFER uses Pseudo-Hadamard Transform (PHT).

Permutation and linear transformation are two different techniques to achieve diffusion. We analyze the characteristics and performance of these techniques in the following sections. In the analysis, we consider two cases. For the first case, the S-Box satisfies the completeness property such that every output bit of the S-Box is a function of all input bits. For the second case, the S-Box does not satisfy the completeness property. We consider the worst case where the S-Box map into itself, i.e. S(x) = x. The block cipher effectively consists of permutation box only. From Case I, we can determine the minimum rounds required for the block cipher to satisfy completeness. From the Case II, we can determine the maximum number of rounds required.

3.3.1 Analysis on Permutation

In this analysis, we use a P-Box that maps each input bit to an output position. Consider a block cipher of 32-bit block size with four 8 x 8 bits S-Boxes. The outputs of the S-Boxes are combined into a 32-bit block to be fed to the P-Box as shown in Figure 3.1. Assume that the P-Box has the property that every bit will be fed to all S-Box as soon as possible. Thus, every bit will go through all four S-Boxes after four rounds.

Figure 3.1 S-Boxes and P-Box Interconnection

Case I

For the first case where the S-Box satisfies the completeness property, every output bit of the S-Box is a function of all input bits. After one round of transformation, every output bit is a function of eight input bits. After two rounds, every output bit is a function of 16 input bits. Completeness for the block cipher can be satisfied after four rounds when every ciphertext bit is a function of all plaintext bits.

Case II

For the worst case, every output bit is a function of one input bit after one round of permutation. After the second round, every output bit is a function of 2 input bits. After three rounds, every output bit is a function of 4 or 2^2 input bits. The cipher can achieve completeness if every output bit is a function of 32 or 2^5 input bits. Hence, 6 rounds of permutations are required.

3.3.2 Analysis on Linear Transformation

Linear transformation can be performed using different techniques, such as MDS codes and Pseudo-Hadamard Transform (PHT). We perform analysis on linear transformation using PHT. PHT can be considered as a mixing operation that takes the outputs from two S-Boxes as inputs and mix them to get the outputs. If the inputs to a PHT are a_1 and a_2, then the outputs are:

$$b_1 = (2a_1 + a_2) \bmod 256$$

$$b2 = (a_1 + a_2) \bmod 256$$

For the analysis, we consider a block cipher of 32-bit block size with four 8 x 8 bits S-Boxes and two PHT mixers as shown in Figure 3.2. Every output of a single PHT is a function of every input of the PHT. If we view the PHT connection shown in Figure 3.2 as a single mixer that has four inputs and four outputs, every outputs of the mixer is a function of all four inputs. This can be proved by the following equations:

$$b_1 = 2a_1 + a_2$$

$$b_2 = a_1 + a_2$$

$$b_3 = 2a_3 + a_4$$

$$b_4 = a_3 + a_4$$

Then,

$$b_1' = 2b_1 + b_3 = 4a_1 + 2a_2 + 2a_3 + a_4$$

$$b_2' = b_1 + b_3 = 2a_1 + a_2 + 2a_3 + a_4$$

$$b_3' = 2b_2 + b_4 = 2a_1 + 2a_2 + a_3 + a_4$$

$$b_4' = b_2 + b_4 = a_1 + a_2 + a_3 + a_4$$

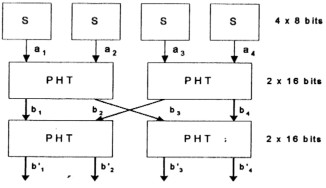

Figure 3.2 S-Boxes and PHT Interconnection

Case I

For the first case, we made the assumption that the S-Boxes satisfy completeness property. Since every output bit of the S-Box is a function of all input bits and every output of the mixer is a function of all its input, then it can be concluded that the block cipher can satisfy completeness property in one round of transformation. This is much faster than block cipher that uses permutation.

Case II

Each output bit of a single PHT is a function of 2 input bits. For the worst case, each output is a function of 4 or 2^2 input bits. After two rounds, every output bit is a function of 16 or 2^4 input bits. After three rounds of transformation, every output bit is a function of all input bits. Hence, completeness can be satisfied after three rounds for the worst case.

Table 3.1 Number of Rounds Required to Satisfy Completeness Property

	Permutation	Linear Transformation (PHT)
Case I	4	1
Case II	6	3

From the analysis performed, we conclude that linear transformation using PHT has a higher level of diffusion compared to permutation. The numbers of rounds required for a block cipher with linear transformation to satisfy completeness property are lower than a block cipher with permutation layer for both cases. Linear transformation is an alternative to permutation to achieve diffusion in a block cipher. Its operations that are not bit oriented make it suitable for software implementation as well as hardware implementation. Suitability for both hardware and software implementations as well as its fast diffusive characteristic makes linear transformation a better choice than permutation. However, the linear operations must be carefully selected so that the linear transformation possesses high diffusive characteristic.

3.4 Key Schedule

After the analysis of substitution layer and diffusion layer, which provides confusion and diffusion to a block cipher, we investigate another important component in a block cipher, the key schedule. A block cipher is a product cipher, which iterates a simple round function many times, each with its own subkey to make the block cipher stronger and more secure against statistical attack. A round function implements the Shannon's principles of confusion and diffusion to transform plaintext to ciphertext. A round function is a weak block cipher. A key schedule is an algorithm that expands a relatively short master key to a relatively large expanded key for later use in an encryption and decryption algorithms. It is used to specify the round keys of a product cipher. The round subkeys should be slightly different per round to prevent a slide attack.

In the following sections, we discuss several attacks that exploit the weak key schedule in a block cipher. We begin with the major attack, related-key cryptanalysis and follow by other common attacks. We analyze the properties of key schedule and explain ways to design a strong key schedule.

3.4.1 Related-Key Cryptanalysis

Related-key cryptanalysis is a powerful attack that exploits weaknesses in key schedule of a block cipher. Related-key cryptanalysis is similar to differential cryptanalysis, but it examines the difference between keys. The attacker knows or chooses the relationship between a pair of keys, but does not know the actual key values. Data is encrypted with both keys. Related-key cryptanalysis assumes that the attacker learns the encryption of certain plaintexts not only under the original (unknown) key K, but also under some derived keys K' = f(K). In a chosen-related-key attack, the attacker specifies how the key is to be changed. Known-related-key attacks are those where the key difference is known, but cannot be chosen by the attacker. In its most advanced form of differential related-key cryptanalysis, both plaintexts and keys with chosen differentials are used to recover the keys.

Here is an overview of a general technique used to perform related-key cryptanalysis. Let SK_i be the i^{th} round subkey generated from master key K under the key schedule. Related-key attack takes advantage of a related key-pair (K,K_0) such that $SK_i = SK'_{i+1}$ for nearly all i, by noting that the action of rounds 1 to r-1 with master key K is equivalent to the action of rounds 2 to r with master key K0. This method is successful against LOKI and Lucifer [33].

One very useful cryptanalytic technique considers a differential attack in which the keys, as well as the plaintexts, are chosen with specific differences. Given a vulnerable key schedule, we can often insert a chosen difference into the middle of a cipher rather than having to pass through all rounds of the cipher.

Related-key cryptanalysis is independent of the number of rounds of a cryptographic algorithm. It has considerable success against block ciphers with simple key schedules, e.g. 3-Way and GOST. Related-key cryptanalysis is a practical attack on key-exchange protocols that do not guarantee key-integrity - an attacker may be able to flip bits in the key without knowing the key and key-update protocols that update keys using a known function: e.g., K, K + 1, K + 2, etc.

3.4.2 Other Key Schedule Attacks

Besides related-key cryptanalysis, there are several attacks on key schedule. While these attacks cannot break the underlying algorithms in all forms, they show a theoretical weakness that may be exploited under certain circumstances.

a) Meet-in-the-Middle Attack

This attack occurs when the first part of a cipher depends upon a different set of key bits than does the second part, which allows an attacker to attack the two parts independently, and works against double-encryption with a block cipher and two different keys [34].

b) Weak Keys

A weak key is a key for which encryption is the same function as decryption. A pair of semiweak keys encrypts plaintext to the identical ciphertext. In other words, one key can decrypt messages encrypted with the other key in the pair.

c) Linear Factors

A fixed set of key bits whose complement leaves the XOR of a fixed set of ciphertext bits unchanged; this weakness can be used to speed up an exhaustive key search.

d) Simple Relations

A simple relation occurs between two different keys, manifesting itself as a relationship between the resulting plaintexts and ciphertexts. This also allows the key space to be reduced in a search. DES and LOKI have a simple relation known as the complementation property. If K encrypts P to C, then the bitwise complement of K encrypts the bitwise complement of P to the bitwise complement of C.

$$E_K(P) = C \quad \Rightarrow \quad E_{M'}(P') = C'$$

where P', C' and K' are the bitwise complement of P, C and K respectively.

This reduces the effective key space by one bit. DES and LOKI have pairs of keys for which a simple relation exists, for at least a fraction of all plaintexts.

e) Equivalent Keys

A pair of equivalent keys, K, K^*, is a pair of keys that encrypts all plaintexts into the same ciphertexts. This can be considered a special kind of simple relation.

f) Detectable Key Classes

One way to reduce the effective key space is to divide the key space into classes, and then find an attack that reveals to which class the key belongs. In some cases, the workload of identifying a key with a specific class is very small; these too are sometimes referred to as weak keys. It can be due to weak mixing in the key schedule.

g) Attacks on One-Wayness

A key schedule is one-way if given several round subkeys; it is infeasible for an attacker to gain any new information about the master key or about other unknown round subkeys. It may be easier to find weak keys and related keys for key schedules that are not one-way. In the DES key schedule, recovering a few round subkeys allows one to recover most of the master key. This can be exploited to optimize differential attack on DES.

3.4.3 Requirements for Strong Key Schedule

All the attacks discussed in the previous section show that key schedule, if not being designed properly, might cause weaknesses in a block cipher algorithm that allows it to be exposed to various attacks. A block cipher may have S-Boxes that are strong against various known attacks and a diffusion layer that provides good avalanche effect. However, a weak key schedule will still cause the block cipher to be vulnerable to certain attacks.

i) Key schedules should be hard to invert, that is given some of the round keys, it should be difficult to recover any new information about other bits of the key.

ii) Key schedules should possess some form of collision freedom to avoid equivalent keys. Equivalent keys considered a special kind of simple relation that allows the key space to be reduced in a search.

iii) Avoid using linear key scheduling. Although some cryptosystems like DES have successfully incorporated linear key schedules, designing linear key schedule appears to be a difficult task. Many ciphers' linear key schedules have been shown to be quite weak. Ciphers like LOKI, LUCIFER, 3-WAY, TEA and SAFER, which uses linear key schedule, have been successfully cryptanalysed.

iv) Ensure that every key bit is about equally powerful in terms of its effect on the round keys. This will maximize avalanche in the subkey to immunize the key schedule against related-key attacks.

v) Key schedule should not be able to produce controlled changes in the round keys since this may ease related-key analysis against the key schedule.

vi) Avoid independent round subkeys since this can dramatically lower the cipher's resistance to related-key attacks. In general, when independent round subkeys are in use, the strength of a cipher against related-key attacks will be approximately proportional to the strength of one round standing on its own.

3.5 Common Block Cipher Structures

A block cipher consists of three major components, i.e. substitution layer, diffusion layer and key schedule. In the previous sections, we analyzed each individual component of the block cipher. In this section, we focus on the block cipher structure which is the basic building block that combines every component in a block cipher.

An n-bit block cipher with a k-bit key uses its key to select among 2^k possible different permutations on the 2^n possible inputs/outputs to the cipher. A round function implements the Shannon's principles of confusion and diffusion to transform plaintext to ciphertext. A round function is a weak block cipher. A product cipher, or sometimes known as an iterative block cipher, iterates a simple round function many times, each with its own subkey. This makes the block cipher stronger and more secure against statistical attack.

Most of the well-known block ciphers use this iterative round structure. There are a few types of iterative round structures, and they are discussed below.

3.5.1 SP Network

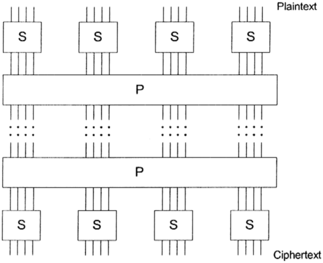

Figure 3.3 Substitution-Permutation (SP) Network

An SP network is a practical realization of Shannon's principles of confusion and diffusion. One round of SP network consists of a substitution layer followed by a permutation layer. The substitution layer confuses the statistics of the input by breaking it into smaller pieces and performing substitutions on them. The permutation layer diffuses the statistics of the input by rearranging the bits output from the substitution layer.

An SP network with a block size of N bits and key K is an invertible mapping

$$F_K : \{0,1\}^N \to \{0,1\}^N.$$

An SP network consists of R rounds, each implements a substitution layer and a permutation layer. In the substitution layer, the n-bit block is fed into M S-Boxes.

The same set of S-Boxes may be used for every round, or the S-Boxes may change from round to round. An n x m S-Box is a mapping of

$$S : \{0,1\}^n \rightarrow \{0,1\}^m,$$

for integers n and m.

The substitution layer is followed by a permutation layer, which permutes the N bit output from the substitution layer. Each output bit should be a function of more input bits. Decryption is accomplished by reversing the order of the rounds. In each round, inverse permutation is performed followed by inverse substitution. Examples of block ciphers using SP network are SAFER, Square and Shark.

3.5.2 Feistel Network

Feistel network was invented by Horst Feistel in his design of LUCIFER. Feistel ciphers are block ciphers. The ciphertext is calculated by recursively applying a round function to a plaintext. For an r-round Feistel cipher with block size of N bits, the round function is defined as follow.

$$\text{Round}_i : (L_i, R_i) \rightarrow (R_{i-1}, F(K_i, R_{i-1}) \oplus L_{i-1})$$

for i = 1 to r, where L_i and R_i are of length N/2. K_i is a round subkey generated from the master key using a key scheduling algorithm. F, the fundamental building block of a Feistel network, is a key-dependent mapping of an input string onto an output string.

$$F : \{0,1\}^{N/2} \text{ x } \{0, 1\}^K \rightarrow \{0, 1\}^{N/2}$$

The F function takes two arguments, N/2 bits of the block and K bits of a key as input, and produces an N/2 bit output. In each round of the iteration, half of the source block is the input of the F function. The output of F is then XORed with the other half, after which these two blocks swap places before entering the next round.

A block cipher using Feistel network construction is guaranteed to be invertible as long as the inputs to F in each round can be reconstructed. This means that F need not be invertible. In order to decrypt a ciphertext, starts at the tail end of the network and works backwards. This is possible since

$$F(K_i, R_{i-1}) \oplus L_{i-1} \oplus F(K_i, R_{i-1}) = L_{i-1}$$

Feistel offers the advantage that the F function needs only be calculated in one direction. It can be designed to be very complicated without having to calculate its inverse function for decryption since Feistel network takes care of its own inverse. On the contrary, an SP network must use different functions to perform its encryption and decryption.

There are three variants of the Feistel network, namely incomplete Feistel Network, source-heavy Feistel network and target-heavy Feistel network.

a) Incomplete Feistel Network

In incomplete Feistel network, the F function has an input and output size that is some fraction of N other than N/2. For the input and output size of N/4:

$$F : \{0,1\}^{N/4} \text{ x } \{0, 1\}^{K} \rightarrow \{0, 1\}^{N/4}$$

Some examples of block ciphers using incomplete Feistel network are Skipjack, Khufu and Khafre.

b) Source-heavy Feistel Network

For this variant of Feistel network, the source and target blocks are of different size, where the source block is larger than the target block, e.g. RC2 has a

48-by-16-bit source-heavy Feistel network, and SHA-1 is a 128-by-32-bit source-heavy structure.

c) Target-heavy Feistel Network

This variant of Feistel network is an opposite of the source-heavy Feistel network, where the F function has a larger output size than the input size. For example, Tiger has a 64-by-192-bit target-heavy structure.

3.5.3 Analysis on Block Cipher Structures

SP network and Feistel network are the most commonly used block cipher building blocks. Comparison between these two structures shows that each has its own advantages and drawbacks.

Feistel network has a nice feature that it is invertible for all choices of the F function. The structure of a Feistel network guarantees that encryption and decryption are similar process, independent of the exact specification of the F function. Hence, the design of the F function can concentrate on the desired propagation properties without restriction imposed by invertibility. Unfortunately, Feistel network has an important drawback. Each round of transformation always keeps one half of the block cipher constant, resulting in a large amount of linearity in the round function. This leads to many attacks such as linear and differential cryptanalysis.

SP network, on the other hand, has to take care of its own invertibility. It has a uniform round structure, consisting of an invertible substitution layer and an invertible linear transformation. Each round function transforms the whole round input. The strong diffusion and uniform nonlinearity allows the reduction of the number of rounds, but the amount of work per round is greater compared to Feistel network.

3.6 Conclusion

Shannon's principles of confusion and diffusion play important roles in good block cipher design. The substitution layer provides confusion and diffusion in each smaller parts of the block ciphers. The diffusion layer provides the overall diffusion needed in wide block cipher.

S-Boxes forms the substitution layer of the block cipher. Nonlinearity, cyclic properties and low probability differential characteristics are required to provide confusion for a block cipher. These properties are very critical to the overall security of the block cipher. To prevent differential cryptanalysis on block cipher, differential distribution table for each S-Box should be uniform to smooth out differentials in any particular round. To be resistant against linear cryptanalysis, Boolean functions used in S-Box should not be linear or affine. There should be no correlations between input and output bits. Completeness, avalanche criterion and strict avalanche criterion are very important to ensure the diffusion of the block cipher. The number of rounds required can be significantly reduced if the outputs of the S-Boxes are highly diffusive.

Permutation and linear transformation are two techniques commonly used in block cipher as the diffusion layer. From the analysis performed, we conclude that linear transformation using PHT has a higher level of diffusion compared to permutation. The number of rounds required for a block cipher with linear transformation to satisfy completeness property are lower than a block cipher with permutation layer for both cases. Linear transformation, if properly designed, is superior to permutation in achieving diffusion in a block cipher. Its operations can be designed to be byte-oriented or word-oriented to suit software, as well as hardware, implementations. Suitability for both hardware and software implementations as well as its fast diffusive characteristic makes linear transformation a better choice than permutation.

Without a strong key schedule, a block cipher that has cryptographically secure S-Boxes and highly diffusive diffusion layer can still be vulnerable to certain attacks. Key schedule should be hard to invert, that is given some of the round keys, it should be difficult to recover any new information about other bits of the key. Nonlinear and highly diffusive key schedule should be used to maximize avalanche in the subkeys. This will immunize the key schedule against related-key attacks.

The block cipher structure is the basic building block that combines every component in a block cipher. We investigate two common structures, i.e. Feistel network and SP network. While Feistel structures are used by DES and many other block ciphers, it has an important drawback. Each round of transformation always keeps one half of the block cipher constant, resulting in a large amount of linearity in the round function. This leads to many attacks such as linear and differential cryptanalysis. SP network on the other hand has a uniform round structure. Each round function transforms the whole round input. The strong diffusion and uniform nonlinearity allows the reduction of the number of rounds.

The Data Encryption Standard is nearing the end of its useful life for many applications. Its 56-bit key is too short for commercial security. Although Triple-DES can solve the key length problem, it is too slow for applications that require high performance. The 64-bit block size used by DES and most other popular encryption algorithms makes it vulnerable to attacks when large amount of data are encrypted under the same key.

In order to overcome these problems, we propose a new block cipher with improved performance and security. It is a block cipher with a 128-bit block size and a 256-bit key. With the increased key length and block size, the proposed block cipher has greater security over DES. Based on the analysis performed, the proposed block cipher is an iterated block cipher, using a standard substitution-permutation network. It consists of an algorithmic S-Box and a highly diffusive linear transformation layer. In the next chapter, we discuss the design strategy of the various components of the block cipher, including the structure of the cipher, S-Box, diffusion layer and key scheduling.