CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In the digital information age, digital communications and information resources affect every aspect of our life – business, government, medical, entertainment and education. The move towards electronic patient records (EPRs) is inevitable because of the desire to improve health care service, the push towards improved cost effectiveness and the growing use of telemedicine and telecare. But by putting patient's medical record online, the risk of exposing highly sensitive and private personal information to outsiders cannot be ignored.

The threats to patients information confidentiality comes not only from inside disclosures and break-ins from outside intruders, but also come from within secondary user settings. Threats from inside the patient care institution include accidental disclosure, insider curiosity and insider subornation. Judging from the success intruders now enjoy in attacking government, business and academic systems, preventing outsider intrusions into health care system is imperative. The threat from secondary users such as medical research, government administration and law enforcement must be very carefully controlled. Those who have access rights to patient information for a purpose in support of primary care may exploit that access for other purposes not envisioned in patients consent forms.

In order to counter various threats to health care information, security services must be put in place. This chapter starts with an overview of cryptography

and cryptanalysis. Cryptography provides means to ensure data confidentiality and prevents vital information from being accessible by unauthorized parties. Cryptanalysis is the science of recovering the plaintext of a message without access to the key. Then, we review cryptographic systems, which can be classified into two groups: private key cryptosystems, and public key cryptosystems. While public key algorithms have been widely used as a solution to the key distribution problem for private key cryptosystems, private key cryptosystems are still the only practical solution for cryptographic applications that require high data rates and low power consumption. Of these two cryptosystems, the focus in this chapter is given to private-key cryptosystem. Thus, we investigate the most recognizable private key cryptosystem, Data Encryption Standard (DES). While DES has been the standard private-key encryption for more than two decades, there was considerable criticism on its weaknesses, especially its small key size. We also review some well-known block ciphers and investigate their strengths and weaknesses. Finally, we discuss multiple encryption, a technique that uses an algorithm to encrypt the same plaintext block multiple times with multiple keys to make the algorithm stronger.

2.2 Cryptography

Cryptography is the science and study of designing systems to encode and decode information to protect it from an enemy. Up till the 1970s, cryptography had a monopoly in the military and diplomatic communities. Today, however, several factors have combined to stimulate great interest in commercial applications, leading to an increasing attention to the generation, transmission, processing and storage of information. The increasingly sensitive nature of business communications and data

crime in recent years has emphasized the need for techniques to protect information, especially when it is in electronic form [1].

Cryptographic techniques are important building blocks in the implementation of all security services. The most basic building block is called a cryptographic system (cryptosystem). A cryptosystem consists of five components:

1. A plaintext message, M
2. A ciphertext message, C
3. A key, K
4. An encryption transformation, E_k
5. A decryption transformation, D_k

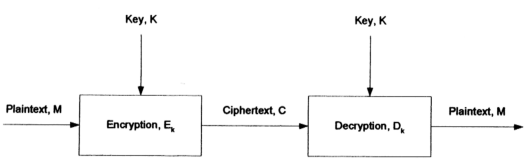

Figure 2.1 Basic Cryptosystem

A cryptosystem defines a pair of data transformations, i.e. the process of transforming an ordinary data item called plaintext into a corresponding unintelligible data item, known as ciphertext is called encipherment or encryption and the reverse process of transforming ciphertext into plaintext known as

decipherment or decryption [2]. The security of the system depends only on the secrecy of the keys and not on the encryption or decryption algorithms. This implies that the encryption and decryption algorithms must be strong. It should not be possible to break a cipher by just knowing the encryption and decryption algorithms. This is because the algorithms may be known to public; whence knowing K reveals E_k and D_k. However, the converse need not hold; that is, knowing E_k or D_k need not reveals K. There are two basic types of cryptosystems, i.e. symmetric systems and asymmetric systems.

2.3 Cryptanalysis

Cryptanalysis is the science of recovering the plaintext of a message without access to the key [3]. A cipher is breakable if it is possible to determine the plaintext or key from the cipher text, or to determine the key from plaintext-ciphertext pairs. An attempted cryptanalysis is known as an attack. There are five basic methods of attack: ciphertext only, known plaintext, chosen plaintext, chosen ciphertext and chosen text [4].

Under a ciphertext only attack, the cryptanalyst must recover the plaintext or determine the key solely from an intercepted ciphertext. For a known plaintext attack, a cryptanalyst knows some plaintext-ciphertext pairs. His job is to deduce the key used to encrypt the messages. A chosen-plaintext attack is more powerful compared to a known-plaintext attack because the cryptanalyst is able to choose the messages to encrypt. Chosen-ciphertext attack means that the cryptanalyst can choose different ciphertexts to be decrypted and has access to the decrypted plaintext. This type of attack is primarily applicable to public-key algorithms. A chosen-plaintext attack and a chosen-ciphertext attack are together known as chosen-

text attack. Table 2.1 is a summary of various types of cryptanalytic attacks based on the amount of information known to the cryptanalyst.

Table 2.1 Types of Attacks on Encrypted Messages

Method of Attack	Known Information
Ciphertext only	• Encryption algorithm • Ciphertext to be decoded
Known plaintext	• Encryption algorithm • Ciphertext of be decoded • One or more plaintext-ciphertext pairs formed with the secret key
Chosen plaintext	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen Ciphertext	• Encryption algorithm • Ciphertext to be decoded • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key
Chosen text	• Encryption algorithm • Ciphertext to be decoded • Plaintext message chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key • Purported ciphertext chosen by cryptanalyst, together with its corresponding decrypted plaintext generated with the secret key

2.4 Substitution Ciphers and Transposition Ciphers

There are two basic types of ciphers: substitution and transposition. Before computers were used, cryptography consisted of character-based algorithms. Different algorithms either substituted characters for one another or transposed characters with one another. With the use of computer, algorithms work on bits instead of characters, but most good algorithms still combine elements of substitution and transposition.

2.4.1 Substitution Cipher

A substitution cipher replaces each character in a plaintext with another character to produce the ciphertext. The receiver inverts the substitution on the ciphertext to recover the plaintext. There are four types of substitution cipher [3]:

- A **simple substitution cipher** or **monoalphabetic cipher** replaces each character of an ordered plaintext alphabet with the corresponding character of an ordered cipher alphabet.
- A **homophonic substitution cipher** is similar to the simple substitution cipher, except the mapping is one-to-many and each plaintext is enciphered with a variety of ciphertext characters.
- A **polyalphabetic substitution cipher** uses multiple mappings from plaintext to ciphertext characters. By using different monoalphabetic substitutions as one proceeds through the plaintext, polyalphabetic substitution cipher conceals the single-letter frequency distribution of the plaintext.
- A **polygram substitution cipher** is made up of multiple simple substitution ciphers. By enciphering large blocks of letters, polygram substitution cipher

makes cryptanalysis harder by destroying the significance of single-letter frequencies.

2.4.2 Transposition Cipher

A different kind of mapping can be achieved by performing some sort of permutation over the plaintext characters. The plaintext remains the same, but the order of characters is shuffled around. In a simple columnar transposition, the plaintext is written into a matrix by rows. The ciphertext is obtained by taking off the columns in some order.

Periodic permutation cipher is another type of transposition cipher, which permutes the characters of the plaintext with a fixed period. Like columnar transposition, it can be viewed as transpositions of the columns of a matrix in which the matrix is written in by rows. However, the ciphertext is also taken out by rows. This is far more efficient for computer applications because each block can be enciphered and deciphered independently [2].

Although transposition cipher is used by many modern algorithms, it is troublesome because of its high memory requirement and sometimes the messages can only be of a certain length.

2.5 Cryptographic Systems

Simmons classifies cryptosystems as symmetric (one-key) and asymmetric (two-key) [5]. In a private-key cryptosystem, the enciphering and deciphering keys are the same, and the key is kept secret, while in public-key cryptosystems, the two

keys differ in such a way that one key is computationally infeasible to determine from the other.

2.5.1 Symmetric Encryption

Symmetric encryption, also referred to as private-key encryption or single-key encryption, is the most prominent and important elements in many cryptographic systems. In symmetric encryption, the encryption key and decryption key are the same. Symmetric encryption enables parties in possession of a shared secret key to achieve the goal of data privacy. There are two categories of symmetric-key encryption schemes: *block ciphers* and *stream ciphers* which are discussed in the following sections.

(i) Block Cipher

A block cipher is an encryption scheme which breaks up the plaintext messages to be transmitted into blocks of a fixed length over an alphabet A, and encrypts one block at a time. It maps n-bit plaintext blocks to n-bit cipher-text blocks; n is called the *block length* [4]. Most well-known symmetric-key encryption techniques are block ciphers. Two important classes of block ciphers are *substitution ciphers* and *transposition ciphers*.

According to Shannon, the two basic techniques for obscuring the redundancies in a plaintext message are confusion and diffusion [6]. Confusion obscures the relationship between the plaintext and the ciphertext. The easiest way to achieve confusion is through substitution. Diffusion dissipates the redundancy of the plaintext by spreading it out over the ciphertext. The simplest way is through

transposition or permutation. Simple substitution and transposition ciphers individually do not provide a very high level of security. However, by combining these transformations it is possible to obtain strong ciphers. Product ciphers iterate several weak operations such as substitution, transposition, modular addition/multiplication, and linear transformation. A block cipher that incorporates layers of substitution and permutation is called a *substitution-permutation network*, or *SP network*. Examples of product ciphers include LUCIFER [7], DES [8], CAST-128 [9], LOKI [10], FEAL [11], IDEA [12], Khufu and Khafre [13].

(ii) Stream Cipher

A stream cipher is one that encrypts a digital data stream one bit or one byte at a time [4]. It is an important class of symmetric encryption schemes. Stream ciphers are useful due to the fact that the individual symbols of a plaintext message are encrypted one at a time using an encryption transformation that varies with time. By contrast, block ciphers tend to simultaneously encrypt groups of characters of a plaintext message using a fixed encryption transformation. Stream ciphers are generally faster than block ciphers in hardware. They are more appropriate when buffering of data is limited and the data must be processed one symbol at a time. In situations where transmission errors are highly probable, stream ciphers are advantageous because they have no error propagation [38].

2.5.2 Asymmetric Encryption

After symmetric encryption, another form of encryption known as public-key encryption has revolutionized the communications security. The concept of public-

key cryptography was introduced in 1976 by Whitfield Diffie and Martin Hellman of Stanford University [14].

Public-key algorithms are based on mathematical functions rather than on substitution and permutation. Unlike symmetric encryption that uses only one secret key, public-key cryptography is asymmetric involving the use of a complementary pair of keys to separate the functions of encryption and decryption. It relies on one key for encryption and a different, but related key, for decryption. It is computationally infeasible to determine one key from the other. One key, the private key, is kept secret like a secret key in symmetric encryption. The other key, the public key, is available publicly.

Potentially, there are two modes of use of public-key cryptosystems, depending on whether the public key is used as an encryption key or decryption key. By using the recipient's freely disclosed, unique public key as the encryption key, the sender can send a confidential message to the recipient. Only the holder of the corresponding private key can decrypt and read the message. This is the encryption mode [3].

Public key cryptography can be used to provide authentication. It is also possible to encrypt messages with the sender's private key, allowing anyone who knows the sender's public key to decrypt the message. The reader knows that only the holder of the corresponding private key could have created the message. Therefore, public-key cryptography can be used for data origin authentication and for ensuring the integrity of a message [38].

2.5.3 Comparison of Symmetric Cryptography and Asymmetric Cryptography

Symmetric-key and public-key encryption schemes have various advantages and disadvantages, some of which are common to both. This section highlights a number of these and summarizes features pointed out in previous sections.

(i) Advantages of symmetric-key cryptography

1. Symmetric-key ciphers can be designed to have high rates of data throughput
2. Keys for symmetric-key ciphers are relatively short.
3. Symmetric-key ciphers can be employed as primitives to construct various cryptographic mechanisms including pseudorandom number generators, hash functions, and computationally efficient digital signature schemes, etc.
4. Symmetric-key ciphers can be composed to produce stronger ciphers. Simple transformations which are easy to analyze, and on their own weak, can be used to construct strong product ciphers [38].

(ii) Disadvantages of symmetric-key cryptography

1. In a two-party communication, the key must remain secret at both ends.
2. In a large network, there are many key pairs to be managed. Consequently, effective key management requires the use of an unconditionally trusted third party.

3. In a two-party communication between entities A and B, sound cryptographic practice dictates that the key be changed frequently and perhaps for each communication session.

(iii) Advantages of public-key cryptography

1. Only the private key must be kept secret (authenticity of public keys must, however, be guaranteed).
2. Depending on the mode of usage, a private key/public key pair may remain unchanged for considerable periods of time, e.g. many sessions.
3. Many public-key schemes yield relatively efficient digital signature mechanisms. The key used to describe the public verification function is typically much smaller than for the symmetric-key counterpart.
4. In a large network, the number of keys necessary may be considerably smaller than in the symmetric-key scenario.

(iv) Disadvantages of public-key encryption

1. Throughput rates for the most popular public-key encryption methods are several orders of magnitude slower than the best-known symmetric-key schemes.
2. Key sizes are typically much larger than those required for symmetric-key encryption, and the size of public-key signatures is larger than that of tags providing data origin authentication from symmetric-key techniques.
3. No public-key scheme has been proven to be secure. The most effective public-key encryption schemes found to date have their security based on the presumed difficulty of a small set of number-theoretic problems [38].

While public-key algorithms have been widely used as a solution to the key distribution problem for private-key cryptosystems, private-key cryptosystems are still the only practical solution for cryptographic applications that require high data rates and low power consumption. Private-key cryptosystems, with their attractive features such as high data rates and low power consumption, provide a practical solution for a variety of applications. Symmetric-key block ciphers are the most prominent and important elements in many cryptographic systems. Individually, they provide confidentiality. As a fundamental building block, their versatility allows construction of pseudorandom number generators, stream ciphers, MACs (Message Authentication Codes), and hash functions. Furthermore, they may serve as a central component in message authentication techniques, data integrity mechanisms, entity authentication protocols, and symmetric-key digital signature schemes [38]. The following sections review Data Encryption Standard (DES), the most recognizable private-key cryptosystem.

2.6 Data Encryption Standard (DES)

The work of the IBM research team which produced Lucifer led to the development of the Data Encryption Standard (DES), which is believed to be the most widely used cryptosystem in the world for unclassified information. On March 17, 1975, DES was first published in the Federal Register. DES was adopted as a standard [8] for unclassified data on January 15, 1977. It is reviewed by the National Bureau of Standards every five years. Its most recent renewal was in January 1994 when it was renewed until 1998. On January 1997, the National Institute of Standards and Technology (NIST) announced the development of a Federal

Information Processing Standard for Advanced Encryption Standard (AES). In the following sections, we discuss about the DES algorithm and the problems and weaknesses of DES.

2.6.1 DES Overview

DES is a 64-bit Feistel cipher with a 56-bit key. The DES algorithm transforms every 64-bit block of the plaintext into a 64-bit block of ciphertext. It uses S-boxes and permutations to achieve Shannon's mixing transformation but performs these operations on only half the block at a time. The single key used with the DES algorithm is a 64-bit string that allows every 8^{th} bit to be set for parity. Therefore, the algorithm expects 64 bits, but only uses 56 of them for encryption. Although the DES algorithm was made public, many of the design decisions were not revealed until the early 1990s [3].

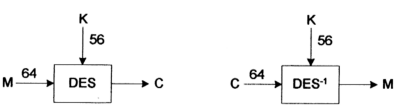

Figure 2.2 DES Input-Output

DES algorithm is a combination of the two basic encryption techniques, i.e. confusion and diffusion. An input block is first transposed under an initial permutation, IP. After it has passed through 16 iterations of a function, *f*, it is transposed under the inverse permutation IP^{-1} (Figure 2.3).

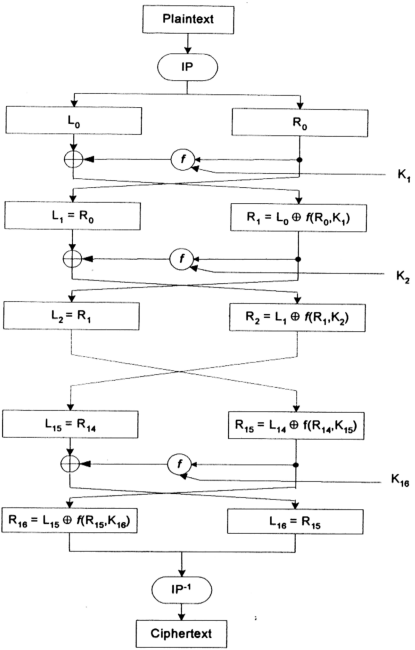

Figure 2.3 DES

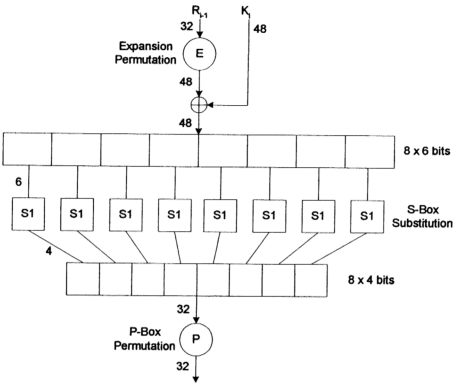

Figure 2.4 DES inner *f* function

Between the initial permutation and final permutation, the algorithm performs 16 iterations of a function, *f*, which combines the following operations (Figure 2.4):

- Key Transformation;
- Expansion Permutation;
- S-Box Substitution;
- P-Box Permutation.

The key transformation produces sixteen 48-bit round subkeys, K_i from K for every round of DES transformations. The 64-bit DES key is reduced to a 56-bit key by ignoring every eighth bit. Then, a different 48-bit subkey is generated for each of the 16 rounds of DES as follows: The 56-bit key is divided into two 28-bit halves. Then, the halves are circularly shifted by either one or two bits, depending on the

round. Compression permutation is then performed to permute the order of the bits as well as to select a subset of bits. This operation produces a subset of 48 bits. Because of the shifting operation, a different subset of key bits is used in each subkey.

The expansion permutation expands the right half of the data, R_i from 32 bits to 48 bits. This operation makes the right half the same size as the key for the XOR operation and provides a longer result that can be compressed during the substitution operation. Its main cryptographic purpose is to achieve an **avalanche effect**. By allowing one bit to affect two substitutions, the dependency of the output bits on the input bits spreads faster.

After the expansion permutation, the 48-bit result moves to a substitution operation. The substitutions are performed by eight different substitution-boxes or S-boxes; each has a 6-bit input and a 4-bit output. The S-box substitution is the critical step in DES because it provides confusion. It is the only nonlinear operation in DES algorithm which gives DES its security. The other operations in DES algorithm are linear and easy to analyze. The result of the S-box substitution is eight 4-bit blocks that are combined into a single 32-bit block.

The 32-bit block is permuted according to a P-box. It is a straight permutation that maps each input bit to an output position. The output of the P-Box permutation is XORed with the left half of the initial 64-bit block. Then the left and right halt are switched and another round begins. After the last round of DES, the left and right halves are not exchanged. This does not affect the security of DES. The purpose is to produce a very useful property: The same algorithm works for both encryption and decryption. The only difference is that the keys must be used in the reverse order for decryption.

In order to apply DES in a variety of applications, four modes of operation have been defined [3], i.e. *electronic codebook (ECB)*, *cipher block chaining (CBC)*, *cipher feedback (CFB)* and *output feedback (OFB)*. These four modes cover virtually all possible applications of encryption for which DES could be used. These same modes can be applied to any symmetric block cipher. Table 2.2 gives a summary of these modes.

Table 2.2 DES Modes of Operation

Mode	Description	Typical Application
ECB	Each block of 64 plaintext bit is encoded independently using the same key.	• Secure transmission of single values (e.g. an encryption key)
CBC	The input to the encryption algorithm is the XOR of the next 64 bits of plaintext and the preceding 64 bits of ciphertext.	• General-purpose block-oriented transmission • Authentication
CFB	Input is processed J bits at a time. Preceding ciphertext is used as input to the encryption algorithm to produce pseudorandom output, which is XORed with plaintext to produce next unit of ciphertext.	• General-purpose stream-oriented transmission • Authentication
OFB	Similar to CFB, except that the input to the encryption algorithm is the preceding DES output.	• Stream-oriented transmission over noisy channel

2.6.2 Problems of DES

There are many desirable characteristics for block ciphers. These include:

- Each bit of the ciphertext should depend on all bits of the key and all bits of the plaintext; there should be no statistical relationship evident between plaintext and ciphertext.

- Altering any single plaintext or key bit should alter each ciphertext bit with a probability of 0.5.
- Altering a ciphertext bit should result in an unpredictable change to the recovered plaintext block.

Empirically, DES satisfies these basic objectives. However, there has been much speculation on the security of DES. Some known weaknesses and anomalies of DES are given below.

(i) Key Length

There is a growing concern that the DES algorithm may be broken using an exhaustive search. Diffie and Hellman argued that with 56-bit keys, DES might be broken under a known-plaintext attack by exhaustive search. In 1977, they show that an exhaustive DES key search machine consisting of one million LSI chips, with an estimated cost US$20 million, could try all $2^{56} \approx 7 \times 10^{16}$ keys in 12 hours [15]. In 1984, DES chips capable of performing 256,000 encryptions per second had been produced [16]. By 1987, chips performing 512,000 encryptions per second were being developed [17]. In 1993, Michael Wiener provided a gate-level design for a US$1 million machine using 57,600 DES chips that could complete a brute-force attack against DES in an average of 3.5 hours.

With the current advances in technology and the estimate that cost will drop by a factor of 5 every ten years, DES will only become less secure as times goes on.

(ii) Design of the S-Boxes

Since DES was proposed as a standard, there was considerable criticism. The unpublished S-box design criteria are the main issues in this criticism [39]. There are no apparent reason as to why and what all the constants in the S-box are for. There is

fear that the S-Boxes may have hidden trapdoors. NSA (National Security Agency) was accused of modifying the contents of the S-Boxes designed originally by IBM. People have pointed to this as evidence that the NSA embedded a trapdoor into the algorithm so that they would have an easy means of decrypting messages.

(iii) Complement Keys

If the original key encrypts a block of plaintext, then the complement of the key will encrypt the complement of the plaintext block into the complement of the ciphertext. Let E denote DES, and x' the bitwise complement of x. Then, $C = E_K(M)$ implies $C' = E_{K'}(M')$. In DES, the subkeys are XORed with the right half after the expansion permutation in every round. Complementation property of the key is a result of that fact. This means that a chosen-plaintext attack against DES only has to test half the possible keys; 2^{55} keys instead of 2^{56} [40].

(iv) Weak Keys and Semi-Weak Keys

A DES *weak key* is a key K such that $E_K(E_K(x)) = x$ for all x, i.e., defining an involution. The four DES weak keys are listed in Table 2.3.

Table 2.3 DES Weak Keys

Key Value
00000000 00000000
00000000 FFFFFFFF
FFFFFFFF 00000000
FFFFFFFF FFFFFFFF

Due to the way in which DES generates subkeys for each round of the algorithm, certain initial keys are weak keys [41]. The initial key is split into two halves, and each half is shifted independently. If all the bits in each half are either 0 or 1, then the key used for any round of the algorithm is the same for all the rounds of the algorithm. This can happen if the key is entirely 1s, entirely 0s or if one half of the key is entirely 1s and the other half is entirely 0s.

Table 2.4 DES Semi-weak Key Pairs

01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E0F1 E0F1 F1E0 F1E0
01E0 01E0 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

A pair of DES *semi-weak* keys is a pair (K_1, K_2) with $E_{K1}(E_{K2}(x)) = x$. This means that encryption with one key of a semi-weak pair operates as does decryption with the other. In other words, one key in the pair can decrypt messages encrypted with the other key in the pair. This property occurs when subkeys K_1 through K_{16} of the first key, respectively, equal subkeys K_{16} through K_1 of the second. This requires that a 1-bit or 2-bit circular left-shift of each of left and right halves for K_1 results in the same value as right-rotating those for K_2 the same number of positions. DES has six semi-weak keys as shown in Table 2.4 [42]. If the number of weak keys or semi-weak keys is relatively small, they may not compromise the cipher when used to

assure condentiality. However, several hash modes use block ciphers where an attacker can choose the key input in an attempt to and a collision; in these modes the block cipher should not have any weak or semi-weak keys.

(v) Fixed Points

For each of the four DES weak keys K, there exist 2^{32} *fixed points* of E_K, i.e., plaintexts x such that $E_K(x) = x$. Similarly, four of the twelve semi-weak keys K each have 2^{32} *anti-fixed points*, i.e. x such that $E_K(x) = x'$, where x' is the bitwise complement of x. The four semi-weak keys are the first two pairs shown in Table 2.4 since for these keys, $K_1 = K_{16}'$, $K_2 = K_{15}'$ and so on.

Weak keys, semi-weak keys and fixed points can significantly affect the security of DES. Although the odd of picking one of these weak keys is very low, weak keys should be checked during key generation.

Now that we have discussed the various weaknesses of DES, the next section focuses on some important cryptanalysis techniques developed to exploit the weaknesses of DES. These techniques provide ways to determine the strength and weaknesses of a block cipher.

2.6.3 Cryptanalysis of DES

Cryptanalysis provides a way to exploit weaknesses in a cipher with a complexity less than brute-force. Since differential cryptanalysis was introduced in 1990, many cryptanalysis techniques have been invented. In this section, a few common classes of cryptanalysis techniques on DES are discussed. These cryptanalysis techniques show the strength and weaknesses of DES.

respectively. An *n-round characteristic* is defined as a sequence of difference pairs: $\Omega = \{(\Delta U_1, \Delta V_1), \ldots, (\Delta U_n, \Delta V_n)\}$. A characteristic has a probability which is the probability that a random pair with the chosen plaintext XOR has the round and ciphertext XOR specified in the characteristic. Certain differences in plaintext pairs have a high probability of causing certain differences in the resulting ciphertext bits. Characteristics extend over a number of rounds and essentially define a path through these rounds.

These characteristics can be found by generating a table for each S-Box, representing the number of times a particular output XOR occurs for a given input XOR. Every input XOR of an S-Box suggests a probabilistic distribution of the possible output XORs. Differential cryptanalysis requires one to find good characteristics, i.e. to find pairs of messages, such that the difference of the output of the *n*th round during encryption of these messages is predictable with a relatively high probability. It is based on the fact that in many S-Boxes certain input XORs lead to certain output XORs with fairly high probability, and to certain other output XORs with very low or zero probability. Chosen-plaintext attacks can be mounted that take advantage of the relatively high probabilities to reduce the search space for the key in use.

Differential cryptanalysis uses this property as a tool to identify key bits. The algorithm counts the number of times that a subkey consisting of a subset of the key bits is consistent with the ciphertext difference, ΔC, assuming the characteristic has occurred. After an appropriate number of trials, the correct subkey is counted significantly more times than incorrect subkeys.

(ii) Linear Cryptanalysis

In 1993, M. Matsui introduced a new method for cryptanalysis of block cipher, known as linear cryptanalysis [19], which is essentially a known-plaintext attack. Linear cryptanalysis exploits linear approximations of some bits of the plaintext, ciphertext and key. The attacker uses a known-plaintext attack technique to extract key information by finding a linear equation consisting of plaintext, ciphertext and key terms which is statistically likely to be satisfied. In order to attack an SP network using the linear cryptanalysis technique, the cryptanalyst is interested in the best R-round linear approximation of the form:

$$P_{i1} \oplus \ldots \oplus P_{ir} \oplus C_{j1} \oplus \ldots \oplus = K_{k1} \oplus \ldots \oplus K_{kr}$$

Let p_L represents the probability that the above equation is satisfied. In order for the linear approximation to be valid, $p_L \neq ½$ over all keys and the best expression is the equation for which $|p_L - ½|$ is maximal. If the magnitude $|p_L - ½|$ is large enough and sufficient plaintext-ciphertext pairs are available, the equivalent of one key bit, expressed by the XOR sum of the key bits on the right side of the equation may be guessed as the value that most often satisfies the linear approximation.

In Lemma 2 of [19], Matsui develops an expression for the number of plaintexts required to give a 97.7% confidence in the correct key bit may be approximated by N_L, where $N_L = |p_L - ½|^{-2}$. This shows that N_L can be increased by decreasing $|p_L - ½|$. Therefore, selecting S-Boxes for which $p_L \rightarrow ½$ will increase the difficulty for linear cryptanalysis.

In the next section, we discuss about some of the popular block ciphers besides DES.

2.7 Review of Other Block Cipher

DES dates back from the 1970s and has become inadequate in particular because of its key length of only 56 bits. DES was designed for hardware implementations and requires tricks to operate moderately fast in 32-bit software implementations. All these factors stimulated the development of other block cipher as alternatives for DES. Of the many block ciphers currently available, the focus of this section is given to some of high profile and well-studied algorithms, i.e. FEAL, IDEA and RC5. All the algorithms are based on the two basic principles of cryptography, i.e. confusion and diffusion. However, each algorithm uses different techniques to achieve these principles. The concepts of the algorithms, as well as its strength and weaknesses are reviewed.

2.7.1 FEAL

The Fast Data Encipherment Algorithm (FEAL) is a family of algorithms that has played a critical role in the development and refinement of various advanced cryptanalytic techniques, including linear and differential cryptanalysis. FEAL-N uses a 64-bit block and a 64-bit secret key. It is an N-round Feistel cipher similar to DES, but with a far simpler f-function, and augmented by initial and final stages which XOR the two data halves as well as XOR subkeys directly onto the data halves.

The f-function $f(A,Y)$ maps an input pair of 32 x 16 bits to a 32-bit output. Within the f function, two byte-oriented data substitutions (S-boxes) S_0 and S_1 are each used twice; each maps a pair of 8-bit inputs to an 8-bit output. S_0 and S_1 add a

single bit d $\in$ {0,1} to 8-bit arguments x and y, ignore the carry out of the top bit, and left rotate the result 2 bits (ROT2):

$$S_d(x,y) = ROT2(x + y + d \bmod 256)$$

As the operations of 2-bit rotation and XOR are both linear, the only nonlinear elementary operation in FEAL is addition mod 256.

FEAL was designed for speed and simplicity, especially for software on 8-bit microprocessors. It uses byte-oriented operations (8-bit addition mod 256, 2-bit left rotation, and XOR), avoids bit-permutations and table look-ups, and offers small code size. Needing fewer rounds, the algorithm would run faster. The initial commercially proposed version with 4 rounds (FEAL-4), positioned as a fast alternative to DES, was found to be considerably less secure than expected. FEAL-8 was similarly found to offer less security than planned. FEAL-16 or FEAL-32 may yet offer security comparable to DES, but throughput decreases as the number of rounds rises. Moreover, whereas the speed of DES implementations can be improved through very large lookup tables, this appears more difficult for FEAL.

2.7.2 IDEA

IDEA (International Data Encryption Algorithm) is a block cipher that encrypts 64-bit plaintext to 64-bit ciphertext blocks, using a 128-bit input key. Based in part on a novel generalization of the Feistel structure, it consists of eight computationally identical rounds followed by an output transformation. As with other block ciphers, IDEA uses both confusion and diffusion. The ingenious design of IDEA is supported by a careful analysis of the interaction and algebraic incompatibilities of operations across the groups. The three algebraic groups are:

- XOR;
- Addition modulo 2^{16};
- Multiplication modulo $2^{16} + 1$.

All these operations operate on 16-bit sub-block. Consequently, it performs quite well compared to DES on 16-bit processors. However, its key setup is quite slow in decryption mode as multiplicative inverses have to be calculated. It has also to be noted that a class of weak keys has been discovered. IDEA is based on a firm theoretical foundation. Although cryptanalysis has made some progress against reduced-round variants, no attacks are feasible against full 8-round IDEA.

2.7.3 RC5

The RC5 block cipher has a word-oriented architecture for variable word sizes w=16, 32, or 64 bits. It has an extremely compact description, and is suitable for hardware or software. The number of rounds r and the key byte-length b are also variable. Plaintext and ciphertext are blocks of bit length 2w. Each of r rounds updates both w-bit data halves, using 2 subkeys in an input transformation and 2 more for each round. The only operations used, all on w-bit words, are addition mod 2^w, XOR, and rotations. The XOR operation is linear while the addition may be considered non-linear depending on the metric for linearity. The data-dependent rotations featured in RC5 are the main nonlinear operation used. The rotations, which vary across rounds, distinguish RC5 from iterated ciphers that have identical operations each round.

There is a differential attack that requires 2^{45} chosen plaintexts for 5 rounds, 2^{53} for 12 rounds and 2^{68} for 15 rounds. Since there are only 264 possible chosen

plaintexts, this attack will not work for 15 or more rounds. But Knudsen and Meier [20] presented differential attacks on RC5 and showed that RC5 has weak keys (independent of the key schedule) for which these differential attacks perform even better. Linear cryptanalysis estimates indicate that it is secure after five rounds.

Based on the reviews of the three algorithms, we concluded that each algorithm has its strength and weaknesses. While FEAL appears to remain relatively secure when iterated a sufficient number of rounds (e.g., 24 or more), this defeats its original objective of speed. IDEA is a 16-bit oriented cipher which performs quite well compared to DES on 16-bit. However, its key setup is quite slow in decryption. It has also to be discovered that a class of weak keys exist in IDEA. While no practical attack on RC5 has been found, the studies provide some interesting theoretical attacks, generally based on the fact that the rotation amounts in RC5 do not depend on all of the bits in a register.

No block cipher is ideally suited for all applications, even one offering a high level of security. This is a result of inevitable tradeoffs required in practical applications, including those arising from, for example, speed requirements and memory limitations, constraints imposed by implementation platforms (hardware and software) and differing tolerances of applications to properties of various modes of operation. In addition, efficiency must typically be traded off against security.

2.8 Multiple Encryption

While DES is the most widely used private-key cryptosystem, DES controversy was born at the same time as DES. The main controversy is the 56-bit key size is too short. In order to increase security without going through the trouble

of designing a new algorithm, multiple encryption is introduced. Multiple encryption is a combination technique that uses an algorithm to encrypt the same plaintext block multiple times with multiple keys. In the following sections, we discuss two important cases of multiple encryption, i.e. double and triple encryption. Then, we focus on meet-in-the-middle attack, which could break multiple encryption with much less time at the cost of substantial space.

2.8.1 Double Encryption

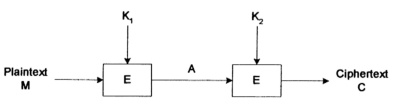

Figure 2.5 Double Encryption

Double encryption is a way of improving the security of a block algorithm by encrypting a block twice with two different keys. First, encrypt a block with the first key, and then encrypt the resulting ciphertext with the second key. Double encryption is defined as:

$$C = E_{K2}(E_{K1}(M))$$

Where E_K denotes a block cipher E with key K.

Decryption is the reverse process:

$$M = D_{K1}(D_{K2}(C))$$

Independent stage keys K_1 and K_2 are typically used in double encryption.

2.8.2 Triple Encryption

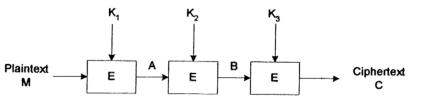

Figure 2.6 Triple Encryption

Triple encryption is defined as $E(x) = E_{K3}(E_{K2}(E_{K1}(x)))$. In triple encryption, dependent stage keys are often used in order to reduce on key management and storage costs. Three types of triple encryption are commonly used: triple encryption with two keys, triple encryption with three keys and triple encryption with minimum key (TEMK).

Triple Encryption with Two Keys

This is a multiple encryption scheme, which encrypts a block three times with two keys. The plaintext block is encrypted with the first key, followed by the second key and finally the first key. Decryption is the reverse process.

$$C = E_{k1}(E_{k2}(E_{k1}(M)))$$

$$M = D_{k1}(D_{k2}(D_{k1}(C)))$$

Tuchman suggested that the sender first encrypts with the first key, then decrypts with the second key and finally encrypts with the first key. In this case, it is commonly known as encrypt-decrypt-encrypt (EDE) mode.

$$C = E_{k1}(D_{k2}(E_{k1}(M)))$$

$$M = D_{k1}(E_{k2}(D_{k1}(C)))$$

In this mode, dependent stage keys are used to save on key management and storage costs. If the block algorithm has an n-bit key, then this scheme has a 2n-bit key. Setting two keys equal to each other is identical to encrypting once with the key. Hence, there is no security improvement over double encryption. This mode has been adopted to improve the DES algorithm in the ISO 8732 standard [21].

Triple Encryption with Three Keys

In this scheme, the plaintext block is encrypted with the first key, followed by the second key and finally the first key. If the block algorithm has an n-bit key, then this scheme has a 3n-bit key. Decryption is the reverse process.

$$C = E_{k3}(E_{k2}(E_{k1}(M)))$$

$$M = D_{k1}(D_{k2}(D_{k3}(C)))$$

This mode of encryption is more secure compared to triple encryption with two keys because three independent keys are used.

Triple Encryption with Minimum Keys (TEMK)

TEMK is a more secure way of using triple encryption with two keys. The trick is to derive three keys from two, X1 and X2:

$$K_1 = E_{X1}(D_{X2}(E_{X1}(T_1)))$$

$$K_2 = E_{X1}(D_{X2}(E_{X1}(T_2)))$$

$$K_3 = E_{X1}(D_{X2}(E_{X1}(T_3)))$$

T_1, T_2 and T_3 are constants which do not have to be secret. This scheme guarantees that the best attack for any particular pair of keys is a known-plaintext attack.

2.8.3 Meet-in-the-middle Attacks on Multiple Encryption

Multiple encryption is much harder to break using an exhaustive search. Instead of 2^k attempts, where k is the key length, it would require 2^{2k} attempts for double encryptions. But this is not to be true for a known-plaintext attack. A time-to-memory trade-off, developed by Merkle and Hellman, could break multiple encryption with much less time at the cost of substantial space [22]. The attack is known as a meet-in-the-middle attack.

Double Encryption

For a block cipher with a k-bit key, a known-plaintext meet-in-the-middle attack defeats double encryption using on the order of 2^{k+1} operations and 2^k storage. It works by encrypting from one end, decrypting from the other, and matching the results in the middle.

Given a plaintext-ciphertext pair, (P, C), compute Mi = E_i(P) under all 2^k possible key values K_1 = i; store all pairs (M_i, i), sorted or indexed on M_i. Decipher C under all 2^k possible values K_2 = j, and for each pair (M_j, j) where M_j = Dj(C), check for hits M_j = M_i against entries M_i in the first table. This can be done by creating a second sorted table, or simply checking each M_j entry as generated. Each hit identifies a candidate solution key pair (i, j), since E_i(P) =M = D_j (C). Using a second known-plaintext pair (P', C'), discard candidate key pairs that do not map P' to C'. The maximum number of encryption trials required is $2x2^n$, or 2^{n+1}.

The meet-in-the-middle attack requires a lot of memory, i.e. 2^k blocks. For DES, this translates to 2^{56} 64-bit blocks, or 10^{17} bytes. But it is enough to convince cryptographer that double encryption is not worth using.

Triple Encryption

Triple encryption with two keys is not susceptible to meet-in-the-middle attack using known-plaintext. The use of different keys alternately prevents this kind of attack. But Merkle and Hellman developed another time-memory trade-off that could break this technique [22]. The meet-in-the-middle attack requires in 2^{k-1} steps using 2^k blocks of memory and 2^m chosen plaintext, where m is the block size. Paul van Oorschot and Michael Wiener converted this to a known-plaintext attack, requiring p known plaintexts [23]. This attack requires $2^{n+m}/p$ time and p memory.

For Triple encryption with three keys, the best time-memory trade-off attack takes 2^{2k} steps and requires 2^k blocks of memory. This is equivalent to exhaustive search for double encryption.

The meet-in-the-middle attack on double encryption and triple encryption with two keys concludes that multiple encryption, if used, should be at least three-fold with independent keys. In order to improve the security of a block cipher, triple encryption with three independent keys is an option. But this can also be achieved by doubling the key size of the block cipher. The key required is shorter and the performance is faster than three encryptions with three different keys.

2.9 Conclusion

Cryptographic systems can be classified into two groups: private-key cryptosystems and public-key cryptosystems. In a private-key cryptosystem, the enciphering and deciphering keys are the same, and the key is kept secret, while in public-key cryptosystems, the two keys differ in such a way that one key is computationally infeasible to determine from the other. While public-key algorithms

have been suggested as a solution to the key distribution problem of private-key cryptosystems, private-key cryptosystems are still the only practical solution for cryptographic applications that require high data rates and low power consumption. Private-key cryptosystems, with their attractive features such as high data rates and low power consumption, provide a practical solution for a variety of applications. Of these two cryptosystems, the focus in this chapter is given to private-key cryptosystem.

DES is the most recognizable private-key cryptosystem. Since DES was proposed as a standard, there was considerable controversy. The small key size and unpublished S-box design criteria are the main issues in this criticism. Today, the cryptographic community is quite aware of the fact that breaking DES is much more feasible than it was previously thought. DES alternatives are sought not only due to the desire for a key length exceeding 56 bits, but also because its bit-oriented operations are inconvenient in conventional software implementations, often resulting in poor performance. Multiple encryption with multiple keys provides a way to strengthen a DES without having to modify the algorithm. However, the only mode that can be considered secure is triple encryption with triple keys since double encryption and triple encryption with two keys are susceptible to meet-in-the-middle attacks. The use of triple-DES, while having good security, is excessively inefficient especially for software implementations. While the need for a serious alternative to DES increases, the dramatic failure of some other proposed alternatives, such as FEAL, illustrates the difficulty of achieving this. No block cipher is ideally suited for all applications, even one offering a high level of security. This is a result of inevitable tradeoffs required in practical applications, including those arising from, for example, speed requirements and memory limitations, constraints imposed by

implementation platforms, e.g. hardware and software. In addition, efficiency must typically be traded off against security. Thus it is beneficial to have a number of candidate ciphers from which to draw.

In order to address 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. In the next chapter, we analyze various components that form a block cipher and their important properties. We investigate various cryptographic properties of each component necessary to resist various kinds of cryptanalysis and statistical attacks. Based on the results of the analysis, we propose a new block cipher that will address these problems.