IMES DISCUSSION PAPER SERIES

A Strength Evaluation of the Data Encryption Standard

Koji Kusuda, Tsutomu Matsumoto

Discussion Paper No. 97-E-5

IMES

INSTITUTE FOR MONETARY AND ECONOMIC STUDIES

Bank of Japan

C.P.O BOX 203 TOKYO
100-91 JAPAN

IMES Discussion Paper Series 97-E-5

July 1997

A Strength Evaluation of the Data Encryption Standard

Koji Kusuda*, Tsutomu Matsumoto**

Abstract

The security of a cryptosystem can be assessed by subjecting it to various cryptanalytic attacks under circumstances considered favorable to the cryptanalyst. In this paper, we have attempted to present an exhaustive review of literature on such cryptanalytic attacks against DES (data encryption standard), including short-cut methods (differential and linear cryptanalyses) and brute force methods (exhaustive search and time-memory trade-off cryptanalysis). We have particularly focused on the importance of time-memory trade-off cryptanalysis and presented some new evidence based on our own numerical examples.

Key words: brute force methods, DES, strength evaluation, short-cut methods, time-memory trade-off cryptanalysis

JEL classification: L86, Z00

* Research Division 2, Institute for Monetary and Economic Studies, Bank of Japan

** Division of Artificial Environment Systems and Division of Electrical and Computer Engineering, Yokohama National University (E-mail: tsutomu@mlab.dnj.ynu.ac.jp)

Finished at the August 1996 for the sake of argument at the Working Group No. 7 of the Sub-committee No. 2 of the Technical Committee No. 68, International Organization for Standardization, this paper has reached this publication. Therefore, it should be noted that this paper does not include various information after the August 1996. The authors are grateful to many researchers, especially, Bart Preneel, Lars R. Knudsen, Serge Vaudenay, Yuliang Zheng, and Kazumaro Aoki, as well as to Kouichi Sakurai, Kazuo Takaragi, Naoya Torii, Michael J. Wiener and staff members of the Institute for Monetary and Economic Studies, Bank of Japan, and Matsumoto Laboratory, Yokohama National University. The authors remain solely responsible for the views presented here.

Contents

Chapter 1

Introduction

Background

Information security should meet a number of requirements such as integrity, which assures that messages are received as sent; authentication, which assures that a message comes from the alleged source; and confidentiality, which protects the content of a message from being disseminated to an unauthorized party. The most widespread tool used to provide such security measures is the cryptosystem, and the degree of security thus depends on how good the cryptosystem is. A cryptosystem should be evaluated not only when introduced but also frequently thereafter taking into account the fact that its strength is bound to be comprised by the rapid advance of computer and cryptanalytic techniques.

The most widely used contemporary cryptosystem, especially in terms of financial applications, is based on the Data Encryption Standard (DES) developed and adopted as the national cipher standard by the United States, which became the *de facto* international standard. DES is a common key cipher, whose algorithm was developed by IBM and adopted in 1977 by the National Bureau of Standards (NBS) now the National Institute of Standards and Technology (NIST), as a Federal Information Processing Standard (FIPS) incorporating changes recommended by the National Security Agency (NSA). Before its adoption as a FIPS, the proposed DES was evaluated by cryptographers at universities and laboratories. Because NSA did not release the design criteria or methodology and test results, the attention of some influential cryptanalysts such as Diffie and Hellman was drawn to the following: (i) a key length of 56 bits might not be long enough to withstand brute force attacks which exhaustively search all the keys; and (ii) the design criteria for the internal structure of DES being classified, there might be some trapdoors which only parties concerned, such as IBM and NSA, would know. In response to such concern, the major operating unit of NBS, the Institute of Computer

Sciences and Technology (ICST), held two workshops to assess the strength of DES before its approval as a FIPS. One workshop assessed the feasibility of a cryptosystem that could conduct a brute force attack, and concluded it would be impossible, until 1990, to produce a cryptosystem which could search all possible cipher keys in a day except at the cost of several tens of millions of dollars. The other workshop examined the possibility of trapdoors having been incorporated, but concluded that this was not the case. Based on the results of these workshops, the security of DES was confirmed, and it was subsequently approved as a FIPS; the standard has been subject to review by NIST every five years since 1983.

Since its adoption as the U.S. federal standard, there has been lingering concern about the level of security provided by it. In 1993, NIST reaffirmed DES as a FIPS for another five years, although there is no guarantee that it will be approved at the next review scheduled for 1998 owing to the rapid progress of both computer technology, which increases the threat of brute force attacks, and cryptanalytical techniques such as the development of differential and linear cryptanalyses. If DES is in fact rejected as a FIPS, users will face a turning point since they rely heavily on NIST's approval in evaluating the strength of DES. Since the strength of DES is obviously not what it was 20 years ago when first developed, some efforts to investigate alternatives have been undertaken. However, the authors do not think it feasible to immediately adopt such alternatives due to the absence of a cipher algorithm that is superior to DES from the viewpoint of processing strength and effectiveness as well as enormous cumulative worldwide investment in financial network security measures that utilize DES. Bearing in mind that the required strength of the cipher differs according to each user's usage, users themselves should be responsible for adopting DES. Therefore, with respect to criteria to evaluate DES, users or user groups should come up with their own evaluation yardsticks separate from NIST's approval of DES as a FIPS.

Based on such views, the International Organization for Standardization's Technical Committee No. 68 (ISO/TC68 – an international committee which promotes standardization in the banking, securities, and other financial service fields), decided at the annual meeting of its Sub-Committee No. 2 (SC2, which deals with operations and procedures including security), held in June 1994, to prepare a plan regarding the future usage of enciphering techniques for financial applications, and accordingly established Working Group No. 7 (WG7) to work on the subject.

Contribution

This paper has been written to provide ISO with a framework for discussion aimed

at evaluating the strength of DES. We have thus tried to give an exhaustive and comprehensive survey of studies regarding the strength of the DES algorithm. Also, for several popular cryptanalyses, we evaluated the strength of DES from the viewpoint of the probability of such cryptanalyses being successful, the time required, overall cost, and required number of plaintexts. In addition, we paid attention to time-memory trade-off cryptanalysis presented in 1980 by Hellman; optimized it from the viewpoints of success probability, time, and cost; and applied the resulting optimized cryptanalysis to test the strength of the DES algorithm.

Results

First, we introduced recent study results, which support the view that the DES algorithm is not closed algebraically and generates a sufficiently large subgroup, and that DES performs quite similarly to a random function. We have also shown some studies which have proved, from the viewpoint of differential and linear cryptanalyses, that cryptographic strength increases as the number of rounds increases. These results would imply that, most likely, DES does not have any fundamental shortcomings and is thus deemed to have a very superior internal structure.

Second, we examined cryptographic strength of the internal structure of DES, *i.e.* the confusion component (S-boxes) and diffusion components (P-box and key schedule). By verifying cryptographic strength of S-boxes based on various measures which have been proposed to date, we obtained results such as: S-boxes have fundamentally well-balanced structures except for their relatively low strength against linear cryptanalysis and, most likely, do not have any significant deficiencies such as trapdoors. Also, examining the diffusion function of the P-box and the key schedule from the viewpoint of rate of achieving completeness led us to the conclusion that, given the structure called the SP-network, these diffusion components are well designed.

Third, we examined the cryptographic strength of DES against various short-cut methods including both differential and linear cryptanalyses. The term "short-cut method" refers to cryptanalysis which tries to reduce the search range of keys based on some information concerning ciphertext and plaintext by taking advantage of analytical characteristics and the statistical bias of the cipher algorithm. By surveying various short-cut method studies, we have confirmed that improved Davies' attack, differential cryptanalysis, and linear cryptanalysis are the only three that can be regarded as superior, at least from some aspects, to brute force methods. Among the three short-cut methods, we have found linear cryptanalysis as most effective against DES, and we have actually applied it.

We obtained the result that if a cryptanalyst wants a 50% probability of success, linear cryptanalysis would be more effective than an exhaustive search by the brute force method, given that some 100 billion blocks (1 block = 8 bytes) of pairs of ciphertext and corresponding plaintext are available. Since it is impracticable, at least for now, to imagine a case where a cryptanalyst would obtain such a large number of ciphertext and plaintext pairs, it can be said that linear cryptanalysis does not pose a threat. However, considering the fact that such an assessment might not be strict enough, and that only three years have passed since the discovery of linear cryptanalysis and thus there might be improvements in the near future, we underline the importance of keeping a close eye on future developments in this area.

Fourth, we examined the cryptographic strength of DES against the brute force methods. In addition to exhaustive search, we focused on an optimized method of, rather overlooked, time-memory trade-off cryptanalysis.

We have compared the various techniques in exhaustive search using the data in published articles of trial computation performed by the presenters themselves. The result was that the apparatus presented by Wiener, equipped with special chips which realizes each round encryption by total pipeline processing, is the best in terms of cost-performance. If Wiener's trial computation is in fact true, it can be stated that crypt-analyzing DES in a short time by exhaustive search is no longer unrealistic. However, such a machine has not yet been actually built, and therefore the accuracy of his trial computation has not been verified.

Therefore, in this article, we have chosen another approach. We assumed a simple parallel processing machine using GaAs chips presented by Eberle and performed a trial computation on the relation between searching time and cost (variable cost only) for exploring a half of the key space of DES. The result was that it was still very difficult to do the search in one day due to formidable cost. However, we obtained some figures implying that a search in a month can no longer be stated as totally unrealistic. Also, we got some figures suggesting that in the year 2001, a search in one day can no longer be stated as totally unrealistic.

In the ciphered data communication system among banks, the common view is that authentication is more important than confidentiality. Hence, there maybe an idea of increasing the frequency of key exchange to about once in an hour as a tentative measure to protect the efficacy of authentication from the attack of exhaustive search. This measure may be effective in a system where a cryptanalyst can obtain known plaintexts only rarely. However, in a system where a cryptanalyst can obtain known plaintexts easily, we showed that increasing the frequency of key exchange is not effective because of the possibility of getting an attack in waves.

For those systems in which fixed plintext unchanged for a long time is transferred either right after each key exchange or at the beginning of each session, or those systems in which a chosen plaintext attack is possible, an attack in time-memory trade-off cryptanalysis should be kept in mind. Time-memory trade-off cryptanalysis is a method which decreases the amount of computation after intercepting the ciphertext by storing a look-up table beforehand which was built by exhaustive search against predicted plaintext in an environment where a plaintext which will be transferred can be foreseen far before the actual interception. We have done a trial computation assuming an Eberle type machine as a key search machine, and the result was that as for building a look-up table in order to explore more than a half of the key space of DES, it takes 2.5 times long time compared with the total hours required for exhaustive search. But once the look-up table is prepared, we obtained some data implying that cryptanalysis in 10 minutes or less may no longer be stated as totally unrealistic. In addition, it was shown that in a ciphered data communication system where interception of a common fixed plaintext against multiple lines is relatively easy, a cryptanalyst can attack multiple lines one after another in the aim of masquerading oneself as an indefinite legitimate user. It was shown that in this case, the amount of precomputation will be substantially lesser, and the size of the machine be much smaller, both compared with exhaustive search.

Implications

Based on our research, we have shown that DES is not as strong as previously thought against cryptanalysis such as time-memory trade-off cryptanalysis. Therefore, when considering the future use of DES, users need to clarify their own necessary level of security, and verify whether the cryptographic strength of DES is sufficient. Considering the current situation with the financial industry investing heavily in information communication systems that use DES, a greater frequency in exchanging the key might be a practical solution for the time being, although this does not help decreasing analytic speed and would thus not result in improving cryptographic strength itself. As an alternative to improving cryptographic strength, Triple-DES would be a desirable solution, taking account of the advantages to users (low replacement cost from the current DES). However, apart from authentication and key management purposes, the government might think that Triple-DES is too strong from the viewpoint of assuring national security (criminal investigations and precautions against terrorism) trying to ensure the possibility of eavesdropping on private telecommunications system. Therefore, future discussions at the ISO should not be limited to verification of Triple-DES, but should also include wide range of research issues on the cryptographic strength of other various common key ciphers. It is worth noting that methods used to evaluate cryptographic strength in this

paper are also applicable to other common key ciphers.

Organization

The organization of this paper is as follows. Chapter 2 introduces cryptography basics. Chapter 3 presents a survey of studies concerning the cryptographic strength of the basic and internal structure of DES. Chapter 4 examines the cryptographic strength of DES against short-cut methods including two powerful attacks (differential and linear cryptanalyses). Chapter 5 provides an estimation of analyzing time, analyzing costs, and success probability when DES is cryptanalyzed by two brute force methods: exhaustive search and time-memory trade-off cryptanalysis. Chapter 6 overviews studies on cryptanalyzes against Triple-DES, followed by an estimation of duration when Triple-DES is adopted. Chapter 7 presents our findings and implications.

Chapter 2

Scope of Cryptanalysis

2.1 Dishonest Acts and Countermeasures

Messages transmitted over electronic lines are vulnerable to dishonest acts, which threaten their confidentiality and authenticity. There are six kinds of dishonest acts: *eavesdropping*, *tampering*, *masquerading*, *replay attack*, *deletion*, and *repudiation*.

- **Eavesdropping**

  This is the interception of the content of a message by an unauthorized person without the knowledge of the sender and receiver. Enciphering a message can prevent an unauthorized person from eavesdropping.

- **Tampering**

  The modification of message content by an unauthorized person. Enciphering a message can detect tampering, *i.e.* the integrity of the message can be confirmed (message authentication).

- **Masquerading**

  Access by an unauthorized person who disguises himself/herself as the rightful sender. Enciphering enables detection, *i.e.* the identity of the counterparty can be confirmed (entity authentication).

- **Replay Attack**

  This refers to behavior where an unauthorized person transmits a previously transmitted message again. Although enciphering alone does not prevent replay attack, cryptographic techniques such as adding a time-stamp to the message along with encipherment can detect such an attack.

- **Deletion**

  The deletion of an entire message by an unauthorized person. A communication protocol (whereby message receipt is acknowledged) can detect such deletion.

- **Repudiation**

  False denial of receipt (transmission) of message by the recipient (sender). A mechanism (enciphering technique assuming the existence of a trusted third party) for dealing specifically with this issue is currently being discussed by ISO/IEC JTC1/SC27 [58] to become a standard as CD 13888-2.

2.2 Cryptosystem Basics

The previous section indicates that encryption plays a central role in effecting countermeasures against dishonest acts. This section examines cryptosystem basics and Fig. 2.1 illustrates the basic model. The original intelligible message which a sender wants to send is called "plaintext" and, using a secret method of writing called encryption, it is transformed into "ciphertext" which is random nonsense to third parties. The process of transforming plaintext into ciphertext is termed "encipherment" or "encryption," and the reverse process of transforming ciphertext into plaintext is called "decipherment" or "decryption." Each such transformation consists of an enciphering/deciphering algorithm (referred to "algorithm" hereafter), and an enciphering/deciphering key (as "key" hereafter) which controls the algorithm.

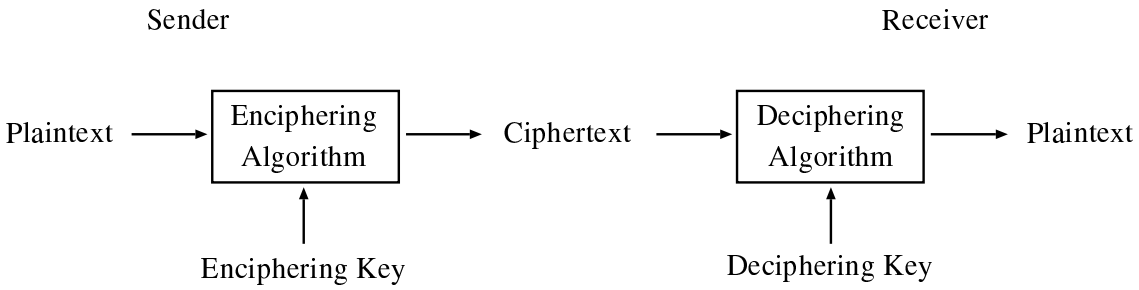

Figure 2.1: Basic cryptosystem.

Let us take a closer look at these basic concepts using the Caesar cipher as an example. As shown in Fig. 2.2, the Caesar cipher involves replacing each letter of the alphabet with the third one after it in the alphabet. In this case, replacing each letter with ones after

(before) them in the alphabet corresponds to the enciphering (deciphering) algorithm, and the number of places (three) the enciphering (deciphering) key.

before replacement:	A	B	C	D	E	F	$\cdots$	X	Y	Z
after replacement:	D	E	F	G	H	I	$\cdots$	A	B	C

For example: BANK OF JAPAN $\rightarrow$ EDQN RI MDSDQ

Figure 2.2: Caesar cipher.

Enciphering/deciphering involves both an algorithm and a key because once participants in a communication network share a cipher implementing a certain algorithm, they can decide a key among themselves, hence making the cryptosystem efficient even in a large network.

The security of a cryptosystem should depend on the key being kept secret, not the algorithm being kept secret. In fact, the algorithm can either be kept secret or made publicly available. It should be noted that, if the algorithm is kept secret as well as the key, it is then virtually impossible to verify the security of the cryptosystem, which is regarded as problematic.

Cryptosystems can be classified into those employing common key ciphers or public key ciphers.

- **Common Key Cipher**

  A system where enciphering and deciphering keys are the same.

- **Public Key Cipher**

  A system where enciphering and deciphering keys are different and, due to the huge amount of computation required to derive one key from the other, one of the keys can be made publicly available.

The public key cipher is convenient in a way because you can make known the key that enciphers the message addressed to you or use the secret key to compute a signature of the message. However, given the low speed in processing the public key cipher to encipher the message, major cryptosystems currently employ common key ciphers, which can be classified further into block and stream ciphers.

Stream cipher is again divided further into synchronous and self-synchronous stream cipher. In a synchronous stream cipher, the key stream is generated independently of the message stream: if a ciphertext block is lost during transmission, the sender and receiver

must resynchronize their key generators before they can proceed. In a self-synchronous stream cipher, each key character is derived from a fixed number n of preceding ciphertext blocks: if a ciphertext character is lost during transmission, the error propagates forward for n blocks, but the cipher resynchronizes itself after n correct ciphertext blocks have been received (see, for example, Daemen [32] for further details).

- **Block Cipher**

  A block cipher enciphers/deciphers a certain length of input elements at one time, using the same key.

- **Stream Cipher**

  A stream cipher enciphers/deciphers a certain length of input elements continuously, using different keys.

As shown in Fig. 2.3 and Fig. 2.4, the block cipher transforms block $P_1\,P_2\,\cdots$ of plaintext P into block $C_1\,C_2\,\cdots$ of ciphertext by the same key K, whereas the stream cipher transforms block $p_1\,p_2\,\cdots$ of plaintext P into block $c_1\,c_2\,\cdots$ of ciphertext continuously by key stream $K = k_1\,k_2\,\cdots$.

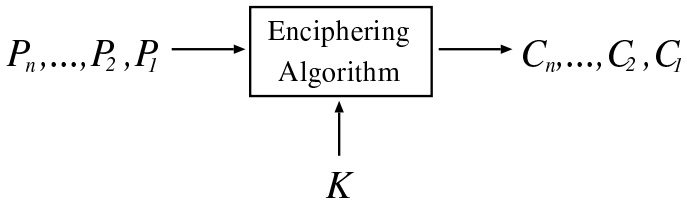

Figure 2.3: Block cipher.

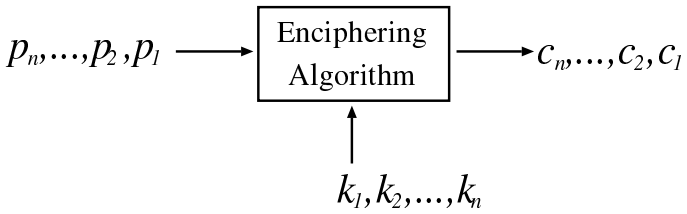

Figure 2.4: Stream cipher.

ISO 8732 [59] stipulates the use of 64 bit block ciphers. ISO 9160 [56] stipulates using stream ciphers of every one bit or eight bits for enciphering the circuit cryptosystem.

The subject of this paper, DES is a common key 64 bit block cipher.

2.3 Cryptanalysis

We define *cryptanalysis* as below, and call one who engages in it a cryptanalyst.

Definition 2.1 *Cryptanalysis is the process of attempting to find a key from some ciphertexts and corresponding plaintexts or only ciphertexts.*

Keys, which are targets for cryptanalysis, should preferably be changed after each and every correspondence or at least daily. Changing keys frequently reduces the number of ciphertexts being enciphered by a key which cryptanalysts can get, as well as messages on which cryptanalysts can eavesdrop even when the key which enciphers the messages has been found. The more chances to conduct cryptanalysis between key changes, the greater possibility for cryptanalysts to be dishonest. Also, changing keys frequently will reduce the probability of a key being acquired before it is changed. The acquiring of a key before it is changed is far more worrisome for users than if a key is acquired after change. Let's explain this further.

If a cryptanalyst acquires a key after change, he/she can commit dishonest acts by making use of the ciphertexts he/she has intercepted and stored before the key change. Dishonest acts would be possible if the cryptanalyst could: (i) know the plaintext corresponding to the ciphertext; (ii) suspend the telecommunication, tamper with the message, or send it to the receiver as if it had been sent during the period when the key was valid; and (iii) tamper with the ciphertexts which the receiver keeps for authentication. If a cryptanalyst acquires a key before it is changed, in addition to the above, the following dishonest acts become possible to effect until the change in the required key: (iv) tamper with the message which is being transmitted; and (v) even masquerade as an authorized person. Among these dishonest acts, countermeasures against (ii) and (iii) are rather easy to take. In the case of (ii), a dishonest act can be detected once a time-stamp is added to the message and a system to check the delayed receipt of a message is established. With respect to (iii), a countermeasure can be taken by enciphering the ciphertexts-to-be-saved as well as limiting access to them. If such countermeasures were taken, therefore, dishonest acts which could be conducted after key change would be limited to case (i), and before key change to (i), (iv), and (v).

The choice of cryptanalysis which a cryptanalyst can carry out depends on what kind of messages he/she can acquire. In this report, we classify cryptanalyses into five types corresponding to information available to a cryptanalyst.

- **Ciphertext only attack**

  cryptanalysis where a cryptanalyst can acquire only some ciphertexts

- **Known plaintext attack**

  cryptanalysis where a cryptanalyst can acquire some ciphertexts and corresponding plaintexts

- **Chosen plaintext attack**

  cryptanalysis where a cryptanalyst can acquire some plaintexts which he chose and corresponding ciphertexts

- **Chosen ciphertext attack**

  cryptanalysis where a cryptanalyst can acquire some plaintexts corresponding to ciphertexts which he freely chose

- **Adaptive chosen plaintext attack**

  cryptanalysis where a cryptanalyst can take into account information contained in ciphertexts corresponding to acquired plaintexts

Adequate security is indispensable against the ciphertext only attack because such an attack can easily be conducted by wiretapping the telecommunication route.

Similarly, security is also essential against the known plaintext attack because one often encounters cases where standard format is used for message transmission.

It is also desirable to have adequate security against the chosen plaintext, chosen ciphertext, and adaptive chosen plaintext attacks. These attacks are now confined to special cases such as: the operator of message transmission secretly communicating with cryptanalysts; or cryptanalysts attacking their own cryptosystem. The latter is likely to become common parallel with the development of electronic money systems based on IC cards and personal computers.

2.4 Operating Modes

When a block cipher is used as DES in the original operation called Electronic Codebook (ECB) mode, the following problems emerge:

- The ECB mode enables a cryptanalyst to carry out the chosen plaintext attack because some special messages (plaintexts) which he/she has chosen are directly used as the input of the original algorithm.

- The known plaintext attack is comparatively easy for the ECB mode. Unless a key is changed, same plaintexts are encrypted to same ciphertexts. The frequent emergence of ciphertext blocks in ciphertexts makes corresponding plaintext blocks inferable.

- A ciphertext can be tampered with in a block unit: a block of a ciphertext can be replaced with another one, or a block of message can be inserted or deleted.

Three operating modes which make use of the block chaining technique are mentioned below. ISO 10126 [60], which prescribes the procedure to encipher (wholesale) banking messages, requires the use of either the Cipher Block Chaining (CBC) mode or the Cipher Feedback (CFB) mode. The error-propagation properties of the CBC mode and CFB mode are not very problematic, and the Output Feedback (OFB) mode was developed to cope with applications in which error-propagation might occur. These three modes are adopted as FIPS [98], ANSI X3.106 [5], ISO 8372 [61], ISO/IEC 10116 [57], and ISO 10126 [60].

2.4.1 CBC mode

Fig. 2.5 illustrates the CBC mode. To begin encryption, an n-bit block Initial Value (IV) is exclusive-ored with the first n-bit plaintext ("exclusive-or" is a bit-by-bit module 2-addition, and referred to as "XOR" hereafter). The result is presented to the block cipher algorithm and an n-bit ciphertext block is produced in the output register. In response to the transmission, the ciphertext is sent to a bit-by-bit modulo 2 addition with the second plaintext block. The result of this addition passes through the block cipher algorithm as before. To decipher, the ciphertext block is presented to the block cipher algorithm and the resultant n-bit block is XORed with IV to produce the plaintext. Each ciphertext block is, in turn, deciphered and then added to the previous ciphertext block. ISO 10126 requires a change of IV in each message transfer.

2.4.2 CFB mode

Fig. 2.6 illustrates the CFB mode with m bit feedback. We begin with a situation where an m bits input register is filled with n bits of block IV. This IV is presented to the block cipher algorithm instead of data and an n bit output block produced. We take m ($1 \leq m \leq n$) bits of output block and XOR them to m bits of plaintext data. In response to the transmission, the resultant m bits of ciphertext are also sent to become the last m bits of the n bits input register. In order to make room for these new bits in the register, the original $n - m$ bits are moved to the left by m places.

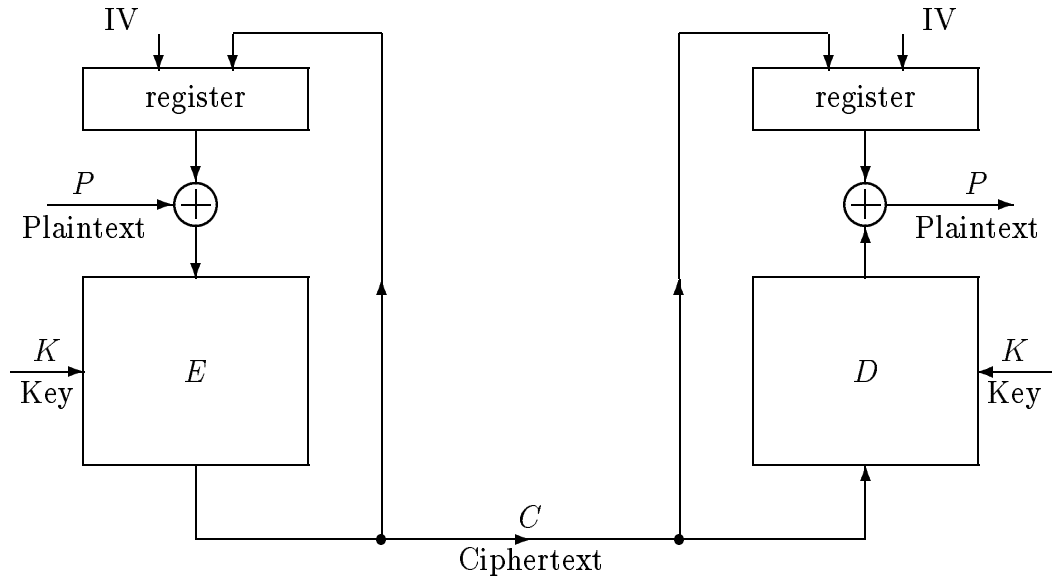

Figure 2.5: Cipher block chaining mode.

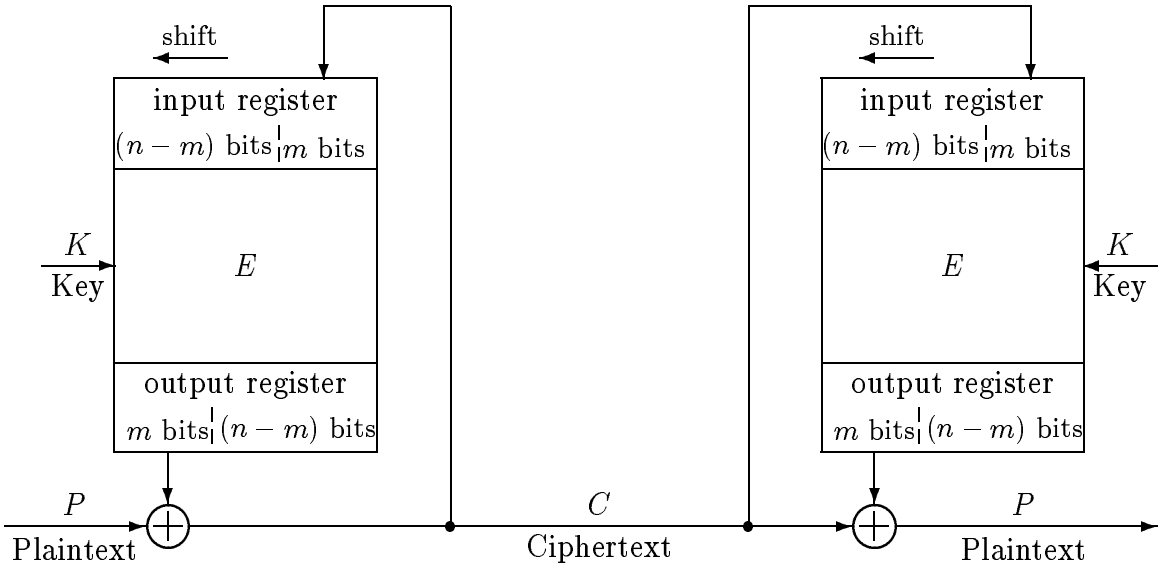

Figure 2.6: m bit cipher feedback mode.

2.4.3 OFB mode

The OFB mode has the property of simply transferring errors in the ciphertext to corresponding bits of the plaintext output. Fig. 2.7 shows the OFB mode with m bits which is identical with the CFB mode with m bits except the place from which the feedback is taken. Since the OFB mode does not feature self-synchronization, it needs *ad hoc* initial and continuous synchronization schemes.

However, as described later in Section 3.3.2, security of the OFB mode was pointed out to be in doubt when $m < n$, and the feedback width was thus fixed at 64 ($= n$) in ISO 8372.

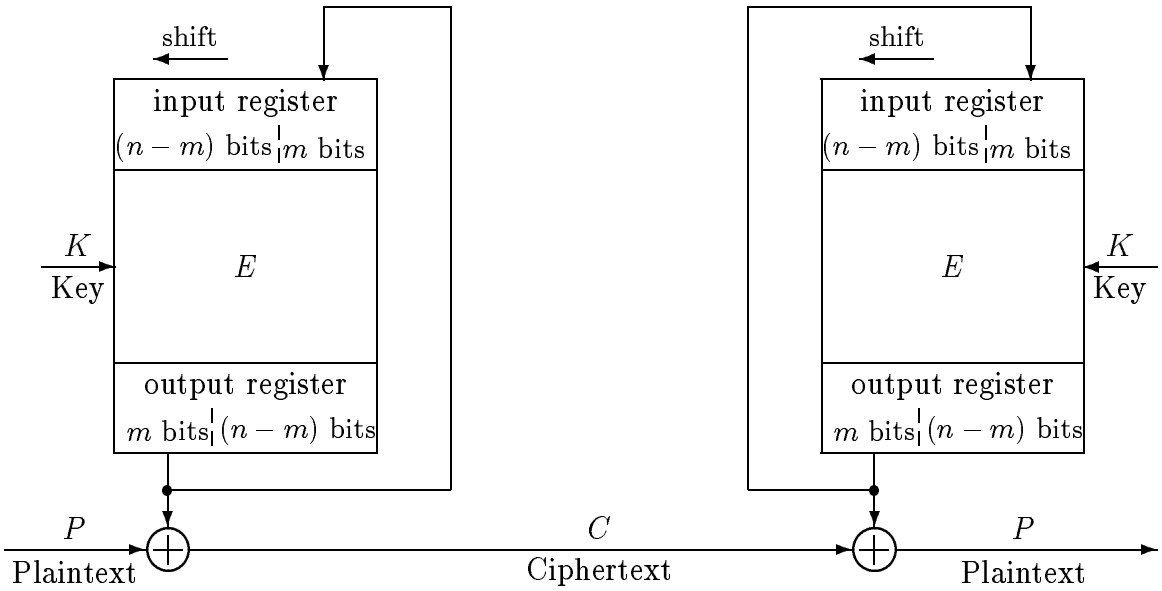

Figure 2.7: m bit output feedback mode.

2.5 Relation Between Operating Modes and Attacks

When the IV is predictable, the relation between each operating mode ($n = 64$) and practicability of each attack mentioned above is shown in Table 2.1. In this case, CBC mode and n-bit CFB-mode are equivalent to ECB mode for cipher chosen plaintext attack.

At CRYPTO '93, Preneel *et al.* [112] first presented a study on a short-cut method

Table 2.1: Relation between operating modes ($n = 64$) and attacks.

	ECB	CBC	OFB	CFB	
			$m = 64$	$m = 64$	$m = 1, 8$
KPA	◯	◯	◯	◯	◯
CPA	◯	×	×	×	×
CCA	◯	◯	×	◯	×

KPA: known plaintext attack
CPA: chosen plaintext attack
CCA: chosen ciphertext attack
◯: applicable
×: not applicable

against the CFB mode of DES. They showed that (1) differential cryptanalysis (modified for CFB-mode) would be successful if $m \geq 3$ with 10 rounds DES of m-bit CFB mode, by using 2^{55} chosen plaintexts, and (2) linear cryptanalysis would be successful if $m \geq 6$ with DES reduced to 8 rounds, to find 7 key bits by using only 2^{31} known plaintexts.

Chapter 3

Studies on DES Structures

Chapter 3 examines the structure of the DES algorithm and studies on the basic and internal structure of DES. This chapter requires advanced knowledge of differential and linear cryptanalyses. Readers who are not familiar with such analyses may want to first read sections 4.2 and 4.3, which discuss them in detail.

3.1 Basic Notations and Definitions

In this paper, we use the following basic notations and definitions. In addition, note that bit-wise exclusive-or is simply called exclusive-or, and well-known expression as XOR is frequently used. Those which are not mentioned here are defined when necessary.

- Z_2^n : the n-dimensional vector space over the finite field $Z_2 = GF(2) = \{0, 1\}$
- $\oplus$: the addition over Z_2^n, or, the bit-wise exclusive-or
- $\mathcal{F}_n$: the set of all functions $Z_2^n \to Z_2$
- $\mathcal{F}_n^m$: the set of all functions $Z_2^n \to Z_2^m$
- $\mathcal{A}_n$: the set of all affine functions $Z_2^n \to Z_2$
- $hwt(\cdot)$: Hamming weight function
- $d(\cdot, \cdot)$: Hamming distance function
- $p_i(F)$: Projection of the Function F of the ith output bit
- $Prob\cdot$: probability of a particular event
- $\#\cdot$: the cardinality of a set

Definition 3.1 (Inner product) *For vectors* $\mathbf{x} = (x_1, \cdots, x_n)$ *and* $\mathbf{y} = (y_1, \cdots, y_n) \in Z_2^n$, *the inner product of* $\mathbf{x}$ *and* $\mathbf{y}$ *is defined as*

$$\mathbf{x} \cdot \mathbf{y} = x_1 y_1 \oplus x_2 y_2 \oplus \cdots \oplus x_n y_n.$$

Definition 3.2 (Entropy) *When* n *discrete output symbols are generated by an information source* S, *and* $p_i (i = 1, \cdots, n)$ *is the probability of the* i*th symbol, the entropy of the information source* S *is defined as*

$$H(S) = -\sum_{i=1}^{n} p_i \log_2 p_i.$$

Definition 3.3 (Walsh-Hadamard transformation) *For* $f \in \mathcal{F}_n$, *the Walsh-Hadamard transformation of* $f : Z_2^n \to R$ *is defined as*

$$\hat{f}(\mathbf{w}) = \sum_{\mathbf{x} \in Z_2^n} (-1)^{f(\mathbf{x}) \oplus \mathbf{w} \cdot \mathbf{x}}$$

where R *denotes the set of all real values.*

Definition 3.4 (Bent function) *A function* $f \in \mathcal{F}_n$ *is called a bent function if and only if*

$$\hat{f}(\mathbf{w}) = 2^{\frac{n}{2}}.$$

Definition 3.5 (Difference distribution table) *A table that shows the distribution of input and output XORs of all possible pairs of an S-box is called the difference distribution table of the S-box. In this table, each row corresponds to a particular input XOR, each column to a particular output XOR, and the entries themselves count the number of possible pairs with such an input XOR and output XOR.*

3.2 Structure of the DES Algorithm

DES enciphers (and deciphers) 64-bit blocks of data with a 56-bit key. The algorithm which is used to encipher (and decipher) is illustrated in Fig. 3.1.

3.2.1 Enciphering and Deciphering

A plaintext block P is first transposed by an initial permutation IP, which is given as $P_0 = IP(P)$. After it has passed through 16 iterations of a function (called *round function*) including a function f (called *F-function*) which is controlled by a 48-bit subkey produced from 56-bit key K, it is transposed by an inverse permutation IP^{-1} to give the ciphertext block C.

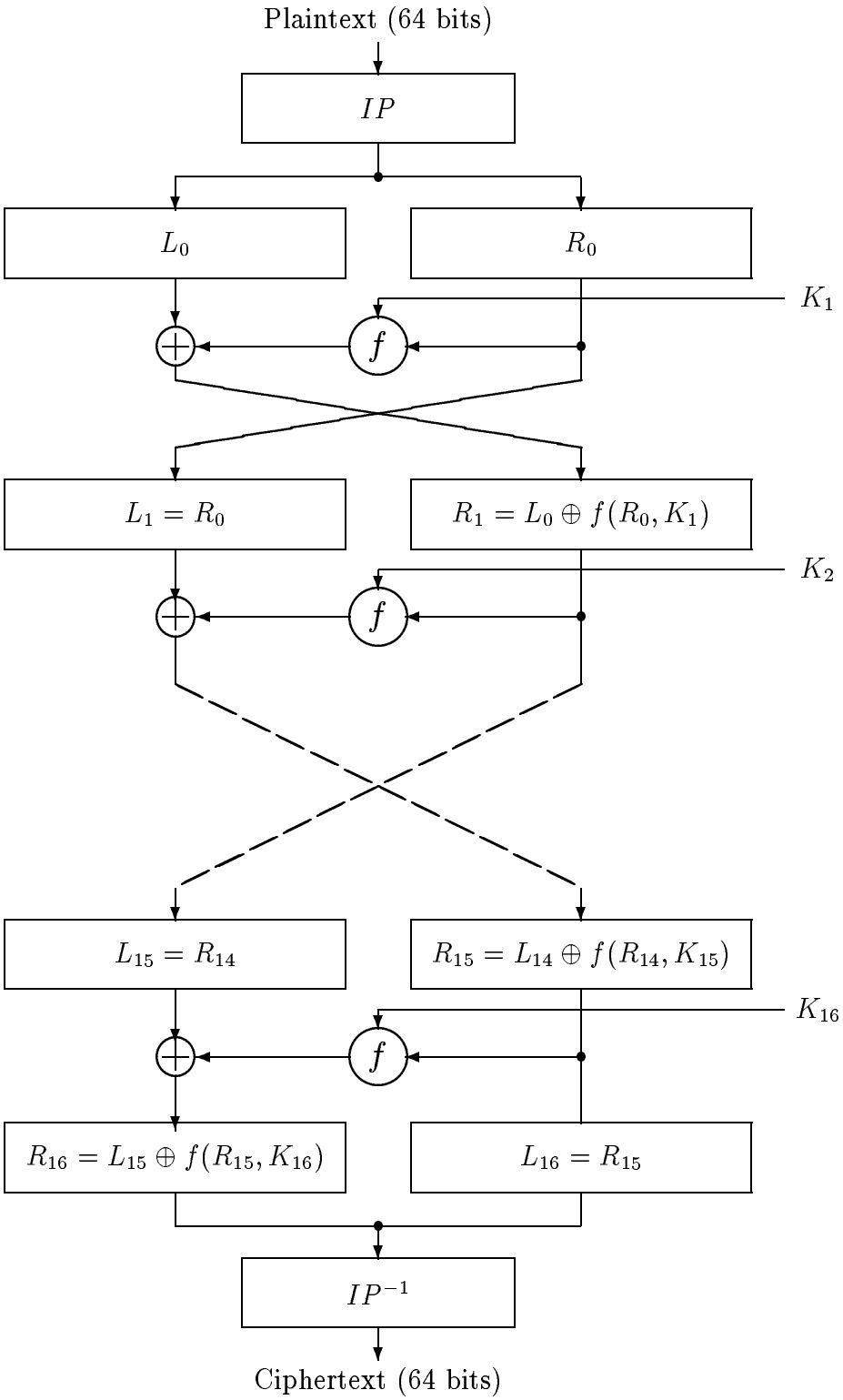

Figure 3.1: DES enciphering algorithm.

Let P_i denote the result of the ith round function, and let L_i and R_i denote the left and right halves of P_i, respectively. Then,

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f\,(R_{i-1},\, K_i) \end{aligned}$$

where $1 \leq i \leq 16$ and K_i is a 48-bit subkey produced by key scheduling. For the details of key scheduling of DES, see [97].

Deciphering is performed using the same algorithm, except that K_i is used in the $(17-i)$th round. This is because the final permutation IP^{-1} is the inverse of the initial permutation IP, and

$$\begin{aligned} R_{i-1} &= L_i \\ L_{i-1} &= R_i \oplus f\,(L_i,\, K_i) \end{aligned}$$

where $1 \leq i \leq 16$.

3.2.2 F-function

The F-function consists of eight *substitutions* (called *S-boxes*) and a *bit permutation* (called the *P-box*). Substitution and bit permutations are defined as follows:

- **Substitution** — Transformation which replaces a number with a different number.
- **Bit permutation** — Transformation of a binary vector, which modifies the order of its component bits without affecting their value.

The idea to combine substitution with permutation finds its origin in Shannon [126] in which he proposed the *product cipher*. When substitution or permutation are used alone in a cryptosystem, many such cryptosystems are very weak. Shannon [126] implied that a product cipher is superior to an individual transformation from a cryptographic point of view. Feistel actualized the idea of a product cipher by building a block cipher called Lucifer [129]. Transformation by Lucifer is generally called *SP-network*, which consists of subsequent substitutions and bit permutations (referred to as "permutation" hereafter). Since DES was constructed based on the structure of Lucifer, it inherits the structure of an SP-network as well. Fig. 3.2 shows a sketch of function $f(R_{i-1}, K_i)$. First, R_{i-1} is expanded to a 48-bit block $E(R_{i-1})$ using the bit selection table E (for details

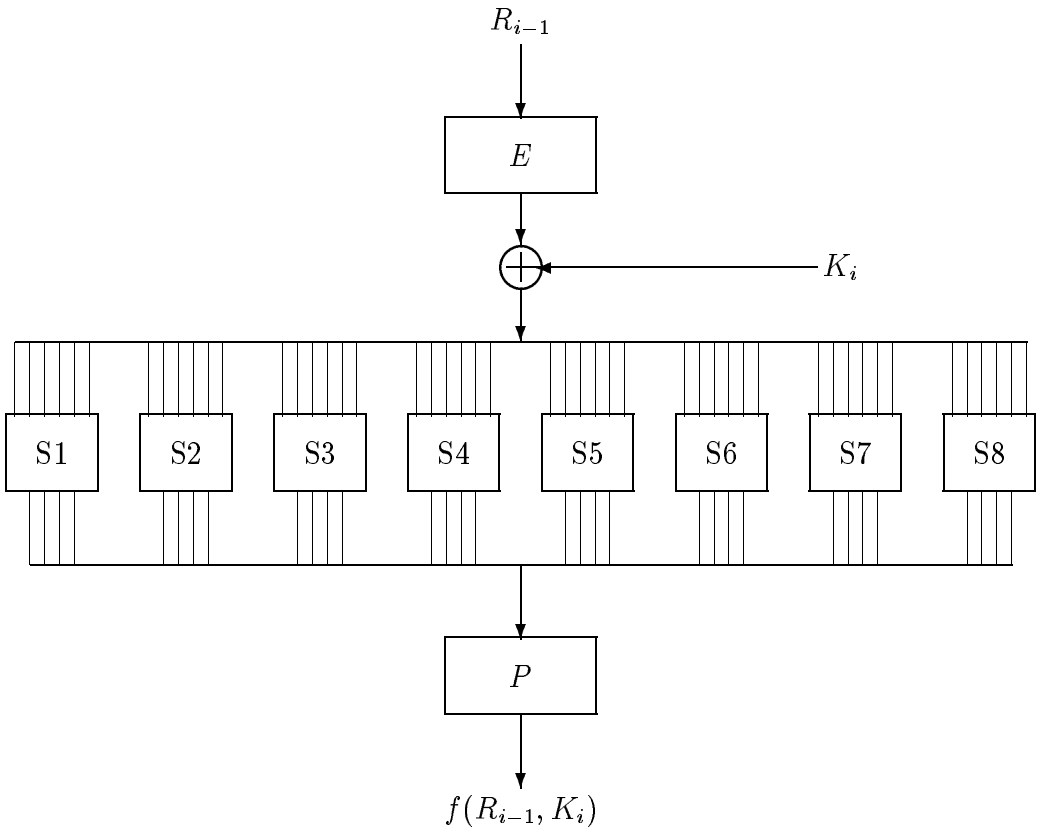

Figure 3.2: F-function of DES.

of E, see [97]). Second, the exclusive-or of $E(R_{i-1})$ and K_i is calculated and the result broken into eight 6-bit blocks $B_1, B_2, \cdots, B_8$, *i.e.*,

$$E(R_{i-1}) \oplus K_i = B_1 B_2 \cdots B_8.$$

Each 6-bit block B_j is then used as an input to an S-box S_j, which returns a 4-bit block $S_j(B_j)$. These blocks are concatenated together, and the resulting 32-bit block is transposed by P-box P. Thus, the block returned by $f(R_{i-1}, K_i)$ is

$$P(S_1(B_1)S_2(B_2)\cdots S_8(B_8)).$$

Each S_j maps a 6-bit block $B_j = b_1 b_2 \cdots b_6$ into a 4-bit one. This is done in the following order: the integer corresponding to $b_1 b_6$ determines a row in the table, while the integer corresponding to $b_2 b_3 b_4 b_5$ determines a column. The value $S_j(B_j)$ is then a 4-bit representation of the integer in that row and column (for details of S-boxes and P-box, see

[97]).

Typical common key cryptosystems other than DES would include: FEAL which NTT [93], [94] proposed in 1987; MULTI2 (Hitachi [130]; 1988); LOKI (Brown *et al.* [22] in Australia; 1990); and IDEA (Lai *et al.* [78]; 1991). Table 3.1 compares the main characteristics of published DES-like cryptosystems.

Table 3.1: Comparison of DES-like cryptosystems.

	DES	FEAL-N/-NX	MULTI2	LOKI	IDEA
Year published	1977	1987	1988	1990	1991
Proposed by	IBM	NTT	Hitachi	Brown *et al.*	Lai *et al.*
Plaintext (bits)	64	64	64	64	64
Key (bits)	56	64/128	64*	64	128
Number of rounds	16	N**	8*	16	8
Intermediate key (bits)	768	$16N + 128$	256	576	768
F-function	SP-network	$+, \oplus, Rot_2$	$+, -, \oplus$ or, Rot_v***	SP-network	$+, \oplus, \odot$****

* ISO registration form stipulates key length as more than 64 bits and number of rounds positive integer. For commercial usage, the number of rounds in MULTI2 is 32 or more, i.e., only MULTI2-32 or upper versions have been used for products.

** $N = 2^n$ $(n = 3, 4, \cdots)$.

*** v is 1, 2, 4, 8, and 16.

**** multiplication modulo $2^{16} + 1$ of 16-bit integers with the zero subblock corresponding to 2^{16}.

3.3 Basic Structure of DES

Here we survey the studies on the basic structure of DES, such as group theoretic issue, cycle length, strength of asymptotic extensions of DES, and cryptographic strength and number of rounds.

3.3.1 Group Theoretic Issue

This section surveys a series of studies which concluded that "DES is not closed" and "DES generates a large group." Before moving on to details we will define the concepts used.

DES defines an indexed set of bijective functions acting on message set $\mathcal{M} = \{0,1\}^{64}$. Let $\mathcal{K} = \{0,1\}^{56}$ denote the set of keys. Each key $K \in \mathcal{K}$ represents a transformation E_K, with inverse E_K^{-1}.

If this set of bijective functions were closed under functional composition (we say that *DES is closed*), then DES would be vulnerable to a known plaintext attack that, on average, runs in 2^{28} steps. Moreover, the most popular proposals for strengthening DES through multiple encryption would be equivalent to single encryption. Granted that DES is not closed, however, a meet-in-the-middle attack would be possible if the size of the subgroup generated by the set of bijective functions is "small" (Kaliski *et al.* [63]).

Closure Test

Kaliski *et al.* [63] proposed two closure tests, namely the *cycling closure test* (CCT) and *meet-in-the-middle closure test* (MCT), which can detect features such as algebraic closure. In addition, they proposed two cryptanalytic methods based on algebraic features. CCT experiments performed by them showed that DES is not closed.

However, their work relied upon randomness assumptions about DES and, consequently, it is quite difficult to use the results of their cycling tests to make any claims about the probability of DES not being closed.

Morita *et al.* [96] proposed the *Switching Closure Test* (SCT) as a practical version of MCT. SCT tests a cryptosystem by detecting that it could have an equivalent key pair (K_1, K_2) to any key K $(K, K_1, K_2 \in \mathcal{K})$ such that $E_{K_2}(E_{K_1}(P)) = E_K(P)$ for any $P \in \mathcal{M}$. After SCT finds the candidates of equivalent key pairs for fixed P, those candidates are tested for any P. They applied SCT to DES, and showed that if you select the null hypothesis that DES is closed, then the hypothesis is rejected with a level of $3 \times 10^{-43}\%$ error probability.

According to Campbell and Wiener [24], in an as yet unpublished paper, Coppersmith described his work on finding a lower bound on the size of the subgroup $\mathcal{G}$, generated by DES bijective functions. He takes advantage of the special properties of $E_{\mathbf{0}}$ and $E_{\mathbf{1}}$ (DES enciphering functions with all 0's and all 1's key).

In Coppersmith's earlier work [31], he explained that the bijective function $E_{\mathbf{1}}E_{\mathbf{0}}$ contains short cycles (of about 2^{32}). This makes it practical to find the length of the cycle produced by repeatedly applying $E_{\mathbf{1}}E_{\mathbf{0}}$ to some starting message. Each of these cycle lengths must divide the order of $E_{\mathbf{1}}E_{\mathbf{0}}$. Therefore, the least common multiple of the cycle lengths for various starting messages is a lower bound on the order of $E_{\mathbf{1}}E_{\mathbf{0}}$. Also, the order of $E_{\mathbf{1}}E_{\mathbf{0}}$ divides the size of $\mathcal{G}$. This makes it possible to get a lower bound on the size of $\mathcal{G}$.

Coppersmith found the cycle length for 33 messages which proved that $\mathcal{G}$ is at least 10^{277}. Campbell and Wiener [24] found the cycle lengths for 295 additional messages. Combining Campbell and Wiener's results [24] with Coppersmith's yields a lower bound

on the size of the subgroup generated by the DES bijective function of 1.94×10^{2499}. This is greater than the number of DES bijective functions, which proves that DES is not closed.

Size of Subgroup Generated by DES

Which subgroup DES generates is important from a cryptographic viewpoint. If the generated group is "too small" in a certain sense, then the algorithm might be vulnerable to meet-in-the-middle attacks (see Kaliski *et al.* [63]).

The experimental result conducted by Kaliski *et al.* [63] supports the hypothesis that the size of the subgroup generated by DES is not "small." Subsequently, as mentioned above, Campbell and Wiener [24] showed that it is greater than 10^{2499}, which is too large for the meet-in-the-middle attack on DES.

Also, Wernsdorf [139] proved that one-round bijective functions would generate an alternating group, and that the size of such a group is "large."

3.3.2 Cycle Length

The operation which iterates a cipher, whose enciphering function is E, n times is recursively defined as

$$E^n(K, X) = E(K, E^{n-1}(K, X))$$

where K and X stand for key and plaintext, respectively. Then we call minimum N which satisfies

$$E^N(K_0, X_0) = X_0$$

as the *cycle of the cipher on plaintext* X_0 *and key* K_0.

When DES is used in OFB mode, it serves as a key sequence generator of a stream cipher. Since a stream cipher attributes its security to the unpredictability of key sequence, DES needs to be a pseudorandom generator with a long cycle and of good quality.

Davies and Parkin [34], Gait [49], and Jueneman [62] *et al.* studied the properties of randomness of the key sequence in OFB mode in view of the cycle length. In OFB mode, m bits of the left side output of DES are converted into key sequence as well as sent to the input register as feedback.

In OFB mode with $m = 64$, it is well known that the average cycle of DES is $O(\#\mathcal{M}/2)$ when one assumes DES to be a completely random bijective function (see Harris [52]). A more precise average cycle is $\#\mathcal{M}/(\ln \log_2 \#\mathcal{M}) \simeq 1.9 \times 2^{61}$ (see Konheim [76]). Based on numerical experiments, Konheim [76] confirmed the assumption that DES is a random bijective function. In addition, when $m < 64$, DES can be regarded as a random function; and it is known that the average cycle of DES is $O(\#\mathcal{M}^{1/2})$ (see

Purdon and Williams [113]). A more precise average cycle, under a certain assumption, is evaluated by the following approximation (see Knuth [75]):

$$\text{average cycle} \simeq \left(\frac{\pi \# \mathcal{M}}{8V}\right)^{\frac{1}{2}} + \frac{1}{3}$$

where

$$V \simeq \begin{cases} 1 - 2^{-m} & (\text{for } m \text{ is small}) \\ 1 - 2^{m-64} & (\text{for } m \text{ is large}) \end{cases}$$

Gait [49] experimented with the OFB mode with a feedback width of 64 bits up to 1,000 enciphering operations, and confirmed that cycles do not exist. Davies and Parkin [34] tried using 8 bits of DES, and verified the validity of the above theories. In other words, they confirmed that if DES is used in OFB mode with a feedback width of 64 bits, then the resulting key sequence will cycle in approximately 2^{62} steps. However, if a feedback width is less than 64 bits, then the expected cycle length is approximately 2^{32} (strictly speaking, results will slightly differ according to feedback width). Also, with $m = 64$, a cycle of less than 2^{36} has not been found.

Therefore, we conclude that: when we use the OFB mode, which uses DES as a key sequence generator of a stream cipher, the cycle will be long and secure enough if $m = 64$; but the cycle will become considerably shorter and less secure if $m < 64$.

3.3.3 Asymptotic Extension of DES

One way to evaluate the superiority of the basic structure of a cryptosystem is to: extend the cryptosystem naturally by its message length, key length, and number of rounds; and then examine whether such an asymptotically extended cryptosystem would become random bijective functions. In this section, we survey a series of studies which examined whether asymptotic extension of DES would become random bijective functions.

Luby and Rackoff [82] explained that such studies were relevant to the security of DES: they argued that the security of DES depends on whether DES passes the "black box" test, which is informally explained as follows:

Construct two black boxes that compute functions from n bits to n bits as follows. One of the functions is selected uniformly at random from all the $2^{n \cdot 2^n}$ possible functions from n bits to n bits. The other function is the DES function with the key set to a uniformly random m-bit string. We say that DES "passes" the black box test if, every algorithm that examines the two black boxes by repeatedly feeding inputs to them and examining the outputs, can obtain a "significant" idea about the difference between the two boxes after

a "feasible" amount of computation (Cleve [28] p.531 ll.14-21).

If DES passes the black box test, then DES is said to be secure against an adaptive chosen plaintext attack (see Luby and Rackoff [82]).

With n fixed at 64, the black box test as stated above is quite informal, because the terms "feasible" and "significant" are imprecise, and it appears difficult to quantify precisely when the values of n and m are fixed. To be specific about these points, it is useful to rely on the variety of cryptographic theories which deal with the asymptotic behavior of asymptotically large cryptosystems. Among them, the black box test was applied to functions mapping n-bit strings to n-bit strings for arbitrarily large values of n. The word "feasible" means polynomially bounded with respect to n, and "significant" quantity means bounded below by a quantity that is larger than the inverse of a polynomial quantity. A cipher refers to any function which maps an $m(n)$-bit string (called the key) and an n-bit string (plaintext) to an n-bit string (ciphertext). It is reasonable to require a cipher to be feasibly computable. The actual asymptotic approach differs by researcher. In order to organize the studies which dealt with this issue, Cleve [28] provided a framework for discussion as the followings.

First, Cleve [28] defined a set of functions called *function generators*, and classified functions which belong to the same set according to the time and space complexity required for computing each.

Definition 3.6 (Function generators) *A function generator G is a function of the form $G = \cup_{i=1}^{\infty} G^n$, where, for each n, $G^n : \{0,1\}^{m(n)} \times \{0,1\}^n \to \{0,1\}^n$.*

We call the first input to G^n ($m(n)$ bit string) the "key".

Definition 3.7 *P is the class of function generators computed in polynomial time.*

Definition 3.8 *$FAN-IN[k(n)]$ is the class of function generators that, after polynomial time preprocessing of key bits, is computed by a function with fan-in bounded above by $k(n)$, where the fan-in of a function is the maximum number of input bits that outputs depend on.*

Definition 3.9 *NC^1 is the class of function generators that, after polynomial time preprocessing of key bits, is computed by a logarithmic depth Boolean circuit.*

Definition 3.10 *$SPACE[w(n)]$ is the class of function generators that, after polynomial time preprocessing of key bits, is computed by $w(n)$ space computations, where $w(n)$ space computations are analogous to non-uniform Turing machine computations that use $w(n)$ space and run in polynomial time.*

Theorem 3.1 shows the relationship among the concepts introduced above.

Theorem 3.1

$$FAN - IN[O(1)] \subset FAN - IN[O(\log n)] \subset NC^1 \subset SPACE[O(\log n)] \subset P$$

After defining definitions 3.11 and 3.12 based on definitions 3.6 to 3.10, Cleve [28] further defined an asymptotic extension of DES as Definition 3.13.

Definition 3.11 (Bijective function generators) *A bijective function generator is a function of the form* $A = \cup_{i=1}^{\infty} A^n$, *where,* $A^n : \{0,1\}^{m(n)} \times \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$, *and, for each* $z \in \{0,1\}^{m(n)}$, $A^n(z) : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ *is a bijective function.*

We will call the first input G^n as the "key", the second input as the "plaintext", and the output as the "ciphertext", respectively.

Definition 3.12 *For any function generator* $G : \{0,1\}^{m(n)} \times \{0,1\}^n \rightarrow \{0,1\}^n$, *we define the associated bijective function generator* $T_G : \{0,1\}^{m(n)} \times \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ *as*

$$T_G(z)(x,y) = (x, y \oplus G(z,x))$$

for all $z \in \{0,1\}^{m(n)}$, *and* $x, y \in \{0,1\}^n$.

Definition 3.13 (Asymptotic extension of DES) *For any class of function generators* B, *we define* $DES[B]$ *as the class of bijective function generators of the form*

$$T_{G_1} \circ S \circ T_{G_2} \circ S \circ \cdots \circ S \circ T_{G_{r(n)}},$$

where $S : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ *as* $S(x,y) = (y,x)$, $r(n)$ *is polynomial in* n, *and* $G_1, G_2, \cdots,$ $G_{r(n)}$ *are all in* B.

DES can be expressed in the form of the above definition with $n = 64$, $m(n) = 56$, $r(n) = 16$, and $B = FAN - IN[12]$ (6 key bits and 6 plaintext bits contribute to fan-in). Cleve [28] considered any class that contains a realistic asymptotic extension of DES to be in $DES[FAN - IN[k(n)]]$, where $k(n)$ is somewhere between $O(1)$ and $O(\log n)$. We also believe that his idea is valid.

In the following, we will survey, within the framework of Cleve [28], a series of studies by Luby and Rackoff [82], Coppersmith and Grossman [30], Even and Goldreich [43], and Cleve [28].

Based on the assumptions that the F-function is a pseudo-random function and that a one-way function exists, Luby and Rackoff [82] proved that a pseudo-random bijective function can be obtained by conducting round function three times. If we reiterate their results within the framework of Cleve [28], it can be summarized as the following theorem.

Theorem 3.2 *If there exists a one-way function then there exists a pseudo-random bijective function generator in* $DES[P]$.

In DES, however, the F-functions are not pseudo-random in any sense. Any function that has the property where none of its output bits depend on any input bits, can very easily be distinguished from a random function. In fact, in the case of DES, the F-functions can be broken in a much stronger sense. In other words $DES[P]$ is too powerful to reflect the design principles of DES.

We are primarily interested in $DES[FAN-IN[O(1)]]$ and $DES[FAN-IN[O(\log n)]]$. If we could show that one of these classes is equivalent to $DES[P]$, then, by the result of the Theorem 3.2, it would follow that there exists a pseudo-random bijective function in $DES[FAN-IN[O(1)]]$ or $DES[FAN-IN[O(\log n)]]$ (provided that a one-way function exists).

Coppersmith and Grossman [30] investigated certain special bijective functions on sets of strings. They showed that, by composing sufficient numbers of their special bijective functions, any bijective function of even parity can be constructed. Even and Goldreich [43] made the link between the work of Coppersmith and Grossman [30] and DES a more explicit one. Instead of an unrealistic setup used in the theorem of Luby and Rackoff [82], they made the F-function quite realistic: F-functions have the property that each of their output bits depends on at most a constant number (in this case two is sufficient). And they showed that a pseudo-random bijective function could be obtained if both the number of rounds and the key length were allowed to be exponentially long. However, exponentially long rounds of bijective functions seem too long to be realistic reflecting the design of DES. Cleve [28] reiterated their results as a proof of Theorem 3.3.

Theorem 3.3

$$DES[FAN-IN[2]] = DES[FAN-IN[O(\log n)]]$$

Cleve [28] was able to show that, for two interesting complexity classes B that are much more powerful than $FAN-IN[O(\log n)]$, $DES[FAN-IN[2]] = DES[B]$, *i.e.*

Theorem 3.4

$$DES[FAN-IN[2]] = DES[FAN-IN[NC1]] = DES[SPACE[\frac{1}{2}\log\left(\frac{n}{3}\right)]].$$

As far as we know, the equality of $DES[FAN-IN[2]]$ and $DES[P]$ has not been proved. We rather think that the equality does not hold with high probability. Hence, it seems quite difficult to prove, directly from Theorem 3.2, that Cleve's idea of the

asymptotic extension of DES is a pseudo-random function. Although, we would like to point out that, since the strong assumption of the round function in Theorem 3.2 could be weakened, the inequality of $DES[FAN - IN[2]]$ and $DES[P]$ would not necessarily prove that Cleve's asymptotic extension of DES is not a pseudo-random function. And we would also like to note that, even if Cleve's asymptotic extension of DES is not a pseudo-random function, it does not mean that the security of DES is low.

3.3.4 Strength and Number of Rounds

The security of iterated cryptosystems is based on the presumption that a cryptographically "strong" function can be obtained by iterating a cryptographically "weak" function over sufficient number of times.

Lai *et al.* [78] showed that the effect of input difference on an iterated can be analyzed provided that the cipher is a *Markov cipher*, and that the above presumption holds with respect to the strength of a cryptosystem against differential cryptanalysis. Their argument can be summarized in Theorem 3.5, but before introducing it we need some definitions.

First, the Markov ciphers can be defined as:

Definition 3.14 *An iterated cipher with round function $Y = F(X, K_i)$ is a Markov cipher if and only if there is a group operation $\otimes$, for all choices of α ($\alpha \neq e$), β ($\beta \neq e$), and γ when the subkey K_i is uniformly random.*

$$Prob\{\Delta Y = \beta \mid \Delta X = \alpha, X = \gamma\} = Prob\{\Delta Y = \beta \mid \Delta X = \alpha\}$$

where $\Delta X = X \otimes X^*$.

DES without the key schedule portion is a Markov cipher under the definition of $\Delta X = X \oplus X^*$, but DES subkeys are neither independent nor uniformly random. However, if we assume the success probability of $(r-1)$-round differential as equal for most of the subkeys (hypothesis of stochastic equivalence), we can argue as follows. For a Markov cipher with independent and uniformly random round subkeys, the probability of an r-round characteristic is given by the Chapman-Kolmogorov equation for a Markov cipher as

$$Prob\{\Delta Y_1 = \beta_1, \Delta Y_2 = \beta_2, \cdots, \Delta Y_r = \beta_r \mid \Delta X = \beta_0\} = \prod_{i=1}^{r} Prob\{\Delta Y_i = \beta_i \mid \Delta Y_{i-1} = \beta_{i-1}\}.$$

Then, the probability of an r-round differential (β_0, β_r) is

$$Prob\{\Delta Y_r = \beta_r \mid \Delta X = \beta_0\} = \sum_{\beta_1} \sum_{\beta_2} \cdots \sum_{\beta_{r-1}} \prod_{i=1}^{r} Prob\{\Delta Y_i = \beta_i \mid \Delta Y_{i-1} = \beta_{i-1}\}.$$

For any Markov cipher, let Π be the *transition probability matrix* of the homogeneous Markov chain $\Delta X = \Delta Y_0, \Delta Y_1, \cdots, \Delta Y_r$. The (i,j) entry in Π $Prob\{\Delta Y_1 = \alpha_j \mid \Delta X = \alpha_i\}$ where $\alpha_1, \alpha_2, \cdots, \alpha_M$ is some agreed-upon ordering of the M possible values of ΔX and $M = 2^m - 1$ for an m bit cipher. Then, for any r, the (i,j) entry in $\Pi^r, p_{ij}^{(r)}$ equals $Prob\{\Delta Y_r = \alpha_j \mid \Delta X = \alpha_i\}$, *i.e.* $p_{ij}^{(r)}$ is the probability of the r-round differential (α_i, α_j).

The following theorem shows that an iterated cipher, which is a Markov cipher under the definition of difference, is secure against differential cryptanalysis after sufficient many rounds.

Theorem 3.5 *For a Markov cipher of block length m with independent and uniformly random round subkeys, if there is a probability vector $(p_1, p_2, \cdots, p_M)$, such that, for all α_i, $\lim_{r\to\infty} Prob\{\Delta Y_r = \alpha_j \mid \Delta X = \alpha_i\} = p_{ij}$, then $\lim_{r\to\infty} Prob\{\Delta Y_r = \beta \mid \Delta X = \alpha\} = \frac{1}{2^m - 1}$ for every differential (α, β).*

O'Connor and Golić [105] also introduced a Markov chain for linear cryptanalysis and proved that the chain converges to uniform distribution for almost all round functions F.

These papers imply that in the independent random subkey model, almost all iterated ciphers become immune to both differential and linear cryptanalyses after a sufficiently large number of rounds.

3.4 Design Criteria for DES S-boxes

The NSA (National Security Agency) advised the designers that a certain part of the design criteria of DES was considered secret and hence IBM was requested not to reveal design criteria. Critics have suggested that special properties might have been incorporated in the design of substitutions and permutation (trapdoors) which could turn out to be a cryptanalytic advantage to a knowledgeable party. In response to questions raised at the second DES workshop, the NSA revealed several aspects of the design criteria:

C-1. No S-box is a linear or affine function of its input.

C-2. Changing one bit in the input of an S-box results in changing at least two output bits.

C-3. The S-boxes are chosen to minimize the difference between the number of 1's and 0's when any single input bit is held constant.

C-4. $S(\mathbf{x})$ and $S(\mathbf{x} \oplus 001100)$ differ in at least two bits.

C-5. $S(\mathbf{x}) \neq S(\mathbf{x} \oplus 11ef00)$ for any choice of e and f.

A DES S-box consists of four rows of 4-bit bijective functions. The input to an S-box is 6 bits. The left and right outermost bits (row bits) determine which function the four remaining bits (column bits) are to be given. This fact gives us a sixth property of DES S-boxes.

C-6. $S(\mathbf{x}) \neq S(\mathbf{x} \oplus 0abcd0)$ for any a, b, c and d, $abcd \neq 0000$.

In addition to these criteria, after differential cryptanalysis had been presented by Biham and Shamir [8], Coppersmith [31] published additional DES S-box design criteria:

C-7. For a given nonzero input XOR and output XOR, no more than 8 of the outputs may exhibit the given output XOR among the 32 pairs of inputs exhibiting the given input XOR.

C-8. Stronger restrictions for zero output XOR for the case of 3 active S-boxes.

The implications of these properties are explained in the following sections.

3.4.1 Criterion C-1

Criterion C-1 is essential to ensure the strength of a cryptosystem. If the S-boxes are affine functions, the entire cipher becomes affine since the remaining functions in the DES algorithm are all linear. Needless to say, the S-box seems to be chosen to be not only non-affine but also a function which has strong nonlinearlity.

For example, the S-box is in a form which prevents the F-function from being expressed by a simple Boolean function. In fact, Schaumüller-Bichl's attempt [119] to express the whole DES algorithm with a simple Boolean function ended in failure. Similarly, S-boxes except for S4 are expressed by two to four nonlinear functions.

However, S4 can be expressed by only one nonlinear function and its three last output functions can be derived from the first (Hellman *et al.* [53]). Also, Pieprzyk and Finkelstein [109], using minimum Hamming distance to the set of affine functions as a criterion, compared the nonlinearity of each row's bijective functions of DES S-boxes with that of their randomly generated bijective functions, and proved the former exhibited lower nonlinearity than the latter.

These properties are not believed to result in DES weakness. For a cryptosystem to be comfortably strong against any predictable cryptanalyses, there are plenty of desirable properties which S-boxes should fulfill. Due to the fact that there are no functions which satisfy all such desirable properties, a function which carries a well-balanced portion

of each property would be deemed as desirable. The nonlinearity criteria pointed out by Pieprzyk and Finkelstein [109], is no more than one desirable property of a strong cryptographic function.

3.4.2 Relationship among C-2, 3, 4, and 6

Criterion C-3 can be stated more explicitly. For an S-box S, let $p_i S$ be the projection of S of the ith output bit, *i.e.* if $S(\mathbf{x}) = (y_1, y_2, y_3, y_4)$, then $p_i\, S(\mathbf{x}) = y_i$. Since four component functions of the S-box are bijective functions, the list of $p_i\, S$ for all 6 bit inputs, $\mathbf{x}$, will contain exactly 32 ones and 32 zeros. Consider the same list with one of the input bits fixed. For example, consider the list for all 6 bit inputs $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5, x_6)$ such that $x_i = 0$. If $i = 1$ or 6, then the list will contain exactly 16 ones and 16 zeros. If $i = 2$, 3, 4 or 5, the list will not necessarily contain the same number of ones and zeros. Criterion C-3 states that S-boxes are chosen to minimize this difference between the number of ones and zeros. With respect to this issue, Brickell *et al.* [17] compared each distribution of the number of ones in the output for three kinds of S-boxes while fixing certain single input bits in 2-5 input bits to zero and changing the remaining bits. Three kinds of S-boxes are: DES S-boxes, S-boxes satisfying only C-6, and S-boxes satisfying C-2, C-4, and C-6. The results are shown in Table 3.2.

Table 3.2: Difference in the number of ones and zeros in the output.

difference	S-boxes of DES	S-boxes s.t. C-6	S-boxes s.t. C-2,4,6
0	35.2	19.2	40.9
1	50.8	34.2	43.0
2	12.6	24.2	12.9
3	1.6	13.6	2.7
4	0.0	6.0	0.5
5	0.0	2.0	0.0
6	0.0	0.6	0.0
7	0.0	0.2	0.0
8	0.0	0.0	0.0

The skirt of distribution of DES S-boxes is much narrower than that of S-boxes satisfying only C-6. Therefore, S-boxes are obviously designed so that this distribution becomes narrower. Also, the distribution shape of S-boxes satisfying C-2, 4, and 6 is similar to that of DES S-boxes, and the result suggests that criterion C-3 is derived from criteria C-2, 4, and 6.

3.4.3 Criterion C-6

Criterion C-6 shows that each S-box is a union of four bijective functions on Z_2^4. Some studies focused on such structure that each S-box in DES is a union of four bijective functions. We will call such mappings *composite bijective functions*. The experimental work of Dawson and Tavares [39] indicated that composite bijective functions have better properties than single bijective functions in Z_4^4. Also, O'Connor [102] complemented this experimental work by showing that for large m, composite bijective functions would yield difference distribution tables that are optimized against at least two properties to facilitate differential cryptanalysis.

3.4.4 Criteria C-7 and C-8

Coppersmith [31] described that most of the criteria are for the purpose of strengthening S-boxes against differential cryptanalysis. In particular, C-7 means that the maximal entry in a difference distribution table of any DES S-boxes is 16, and C-8 means that there is nonzero input XOR with less than three active S-boxes which results in the same output.

Kim [68] constructed DES-like S-boxes based on Boolean functions satisfying the strict avalanche criteria (referred to later). He listed five criteria for the constructions, including "resistance to differential cryptanalysis." Furthermore, eight concrete examples of these S-boxes, the s^2DES S-boxes, are listed. The cryptosystem s^2DES is obtained by replacing all eight DES S-boxes with the eight s^2DES S-boxes, keeping everything else as in DES. Kim's s^2DES S-boxes do not have DES criteria C-4, 5, 7, and 8. The lack of C-5 and 8 enables us to build a two-round iterative characteristic where the inputs differ in only two neighboring S-boxes, *i.e.* a two-round iterative characteristic $\phi^* = 0000580\ 00000000_x$ with probability $(8/64) \times (10/64) \simeq 1/51$ exists (see Knudsen [72]). The probability is much larger than the 1/243 of DES.

We conclude that DES designers initially focused on the strength of DES against differential cryptanalysis.

3.5 Measuring Strength and Desirable Criteria of S-box

From the theoretical standpoint, namely information theory and combinatorial theory, many researchers have studied measures of strength and the desirable design criteria of S-boxes, and examined the relationship between their criteria and acknowledged criteria of the DES S-boxes. Since any S-box may be implemented as a circuit, the S-box

can be modeled as Boolean functions. With this functional representation it is then possible to evaluate S-box strength based on the properties of Boolean functions which describe the S-box. It is generally accepted that Boolean functions should combine the following properties: *completeness*, *balance, regularity, correlation immunity*, *strict avalanche criteria*, *nonlinearity*, *resistance to differential and linear cryptanalyses*. In this section, we will survey the evaluation studies of DES S-box strength, conducted from the viewpoint of definition and practicability of each measure as well as design criterion.

3.5.1 Completeness

The criterion of *completeness* was proposed by Kam and Davida [65]. They defined completeness as:

Definition 3.15 (Completeness) *For every possible input value, every output bit depends on all input bits.*

Suppose a cryptosystem does not satisfy completeness and, for some value of the key, some output bit c_i depends only on a few input bits. By observing a significant number of plaintext-ciphertext pairs, the cryptanalyst may be able to detect the relationship among c_i and the corresponding small subset of input bits. The cryptanalyst subsequently uses this information to find the correct key value.

However, as noted by Forré [46], this is a very weak criterion as it does not specify how strong the dependency should be. Dawson and Tavares [39] extended the idea, to define *information completeness*, by requiring that each output bit has to depend on all the information included in each input bit instead of depending on only part of the information included in each input bit. This suggests that any function of the inputs, whose output set has less information content than the original set of inputs, cannot produce the same set of outputs for the S-box. In Fig. 3.3, if the set $\mathbf{x}' = \{x'_1, \cdots, x'_m\}$ made by mapping $G(\mathbf{x}) = \mathbf{x}'$ produces the same outputs (from the S-box) as the set $\mathbf{x}' = \{x'_1, \cdots, x'_m\}$ and has less information content than $\mathbf{x}$, then the S-box is considerable to be incomplete in terms of information.

Definition 3.16 (Information completeness) *A function $F \in \mathcal{F}_n^m$ is said to be information complete if and only if there exists no function G such that:*

$$H(G(\mathbf{x})) < H(\mathbf{x}') \tag{3.1}$$

where $G(\mathbf{x}) = \mathbf{x}'$, and $F(\mathbf{x}') = F(\mathbf{x})$ for all values of $\mathbf{x}$.

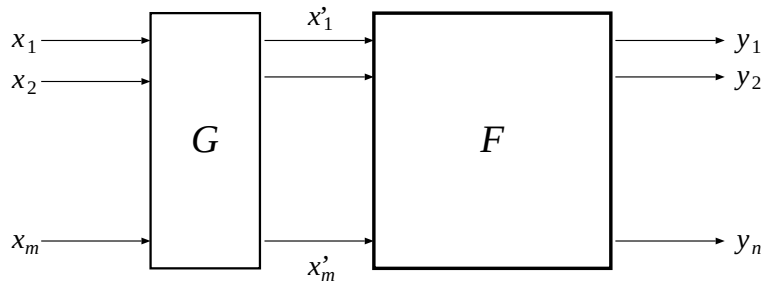

Figure 3.3: Illustration of information completeness for an $m \times n$ bit S-box.

It is clear that information completeness includes the notion of completeness because if any output bit of the S-box does not depend on input bit k, the following function exists:

$$G_{inc}(x_1, \cdots, x_m) = (x_1, \cdots, x_{k-1}, 0, x_{k+1}, \cdots, x_m). \tag{3.2}$$

Since $H(G_{inc}(\mathbf{x})) < H(\mathbf{x})$ and $F(G_{inc}(\mathbf{x})) = F(\mathbf{x})$ for all values of $\mathbf{x}$, the S-box will fail the information completeness test.

This criterion is important because if an S-box is not information complete, the outputs may be reduced to a function of fewer inputs (produced by a smaller S-box), which will reduce system strength because smaller S-boxes have less adequate properties than larger ones.

3.5.2 Balance, Regularity, and Correlation Immunity

Balance

For cryptographic application, it is often very important that the truth table has as many "zero"s as "one"s. In such case, uncertainty regarding the value of $f \in \mathcal{F}_n$ or entropy $H(f)$ is maximal.

Definition 3.17 (Balance) *A Boolean function $f \in \mathcal{F}_n$ is said to be balanced if and only if $hwt(f) = 2^{n-1}$.*

Regularity

Regularity can be regarded as an expansion of the concept of balance to a vector-valued function. It is defined as:

Definition 3.18 (Regularity) *A function $F \in \mathcal{F}_n^m$ is regular if and only if $F(\mathbf{x})$ obtains each value in Z_2^m for 2^{n-m} values of $\mathbf{x}$.*

Regularity of a function $F \in \mathcal{F}_n^m$ can be characterized by the balance of nonzero linear combinations of its component functions. In other words, the following theorem holds (a proof can be found in Dillon [41])

Theorem 3.6 *Let a function $F(=[f_1, f_2, \cdots, f_m]) \in \mathcal{F}_n^m$ be regular, where f_i is a function in $\mathcal{F}_n$. Then any nonzero linear combination of its component functions has Hamming weight 2^{n-1}, i.e.,*

$$hwt(\sum_{i=1}^{n} a_i f_i) = 2^{n-1}$$

for any $a_i \in Z_2$, $(a_1, a_2, \cdots, a_n) \neq (0, 0, \cdots, 0)$.

This condition will allow us to say that, in principle, every f_i is required to be balanced. It can be easily shown that DES S-boxes are all regular.

Correlation Immunity

A concept related to balance, suggested by Siegenthaler [127], is correlation immunity.

Definition 3.19 (Correlation immunity) *A function $f \in \mathcal{F}_n$ is said to be m-th order correlation immune, $CI(m)$ $(1 \leq m \leq n)$, if and only if $\hat{f}(\mathbf{w})$ is statistically independent of any subset of m input variables.*

Xiao and Massey [141] showed that this condition is equivalent to the condition that $f(\mathbf{x})$ is statistically independent of any linear combination of at most m input variables. A function satisfies $CI(m)$ if and only if m input variables give no additional information about the output, or if and only if mutual information among any m input variables $i_1, i_2, \cdots, i_m$ and the output is equal to zero:

$$H(f(\mathbf{x})) - H(f(\mathbf{x}) \mid x_{i_1}, x_{i_2}, \cdots, x_{i_m}) = 0. \tag{3.3}$$

Dawson and Tavares [39] used the concept of mutual information to compare four functions: component bijective functions of DES S-boxes, randomly generated bijective functions, bijective functions which Webster and Tavares [138] created in order to fulfill the influential strict avalanche criteria (referred to later), and their new bijective functions which took account of mutual information (see Table 3.3). The mutual information of bijective functions used in DES was substantially lower than that of random bijective functions and also Webster's bijective functions, and almost equal to Dawson's bijective functions.

Table 3.3: Mutual information between m inputs and output.

m	DES	random	Webster	Dawson
1	0.017	0.037	0.052	0.011
2	0.097	0.156	0.282	0.079
3	0.391	1.875	0.500	0.375

Dawson and Tavares [39] also used the concept of mutual information to verify other relationships such as that between changes in input and output value. Based on the results, they concluded:

The investigations revealed that we could not find S-boxes with substantially better information theoretic properties than DES S-boxes. This indicates that DES S-boxes may be some of the best possible based on a combination of our information theoretic properties (Dawson and Tavares [39] p.361 ll.25-28).

3.5.3 Strict Avalanche Criteria

Feistel [45] created one important criterion to design cryptographic functions.

Definition 3.20 (Avalanche effect) *A function $F \in \mathcal{F}_n^m$ exhibits the avalanche effect if and only if*

$$\sum_{\mathbf{x} \in Z_2^n} hwt(F(\mathbf{x}) \oplus F(\mathbf{x} \oplus \mathbf{c})) = m\, 2^{n-1} \tag{3.4}$$

for all $\mathbf{c} \in Z_2^n$ such that $hwt(\mathbf{c}) = 1$.

This means that an average of one half of the output bits will change whenever a single input bit is complemented.

Webster and Tavares [138] introduced the *Strict Avalanche Criterion* (SAC) in order to combine the notions of the completeness and avalanche effect

Definition 3.21 (SAC) *A function $F \in \mathcal{F}_n^m$ satisfies SAC, if and only if for all $\mathbf{c} \in Z_2^n$ such that $hwt(\mathbf{c}) = 1$ the following equations hold:*

$$\sum_{\mathbf{x} \in Z_2^n} F(\mathbf{x}) \oplus F(\mathbf{x} \oplus \mathbf{c}) = (2^{n-1}, \cdots, 2^{n-1}). \tag{3.5}$$

Preneel *et al.* [110] extended the SAC to a concept, later called as *higher order SAC.* [1]

[1] The definition of higher order SAC we used was first given by Preneel *et al.* [110] (called the propagation criterion of order k) and later independently by Adams *et al.* [3] (where it was called higher order SAC).

Definition 3.22 (Higher order SAC) *A function $F \in \mathcal{F}_n^m$ satisfies the k-th order SAC, $f \in SAC(k)$, if and only if for all $i (1 \leq i \leq n)$ and $\mathbf{c} \in Z_2^n$ such that $1 \leq hwt(\mathbf{c}) \leq k$ there hold the following equation:*

$$\sum_{\mathbf{x} \in Z_2^n} F(\mathbf{x}) \oplus F(\mathbf{x} \oplus \mathbf{c}) = (2^{n-1}, \cdots, 2^{n-1}). \tag{3.6}$$

In order to construct a strong S-box, it was once a dominant idea among researchers that at least SAC, not necessarily higher order SAC, should be satisfied. Such an idea was justified because, a randomly generated S-box satisfying SAC would clear C-1, C-3, and C-5 DES design criteria with a high probability.

An exceptional view can be found in Dawson and Tavares [39]. They indicated that avalanche properties can be divided into three criteria:

Definition 3.23 (Probabilistic avalanche criteria) *An S-box satisfies the probabilistic avalanche criterion if each output of the S-box changes with 50% probability when the input is changed.*

Definition 3.24 (Directed avalanche criteria) *An S-box satisfies the directed avalanche criterion if each output of the S-box changes with 50% probability when certain patterns of change are made to the input.*

Examples of directed avalanche criteria are SAC and higher order SAC.

Definition 3.25 (Minimal avalanche criteria) *An S-box satisfies the minimal avalanche criterion if a minimum number of output bits change when certain patterns of change are made to the input.*

DES design criterion C-2, which requires that at least two output bits change when one input bit is changed, is a good example of a minimal avalanche criterion.

Dawson and Tavares [39] claimed that, among avalanche properties, only the probabilistic avalanche criteria could be regarded as the fundamental property of a strong S-box. However, they also argued the effectiveness in adopting other avalanche criteria as design criteria when composing a large substitution, which is necessary for SP network based cryptosystems with small S-boxes. [2]

The dependence matrix of a DES-like S-box is used to check if DES-like S-boxes satisfy SAC. In the dependence matrix, $\mathbf{P} = (p_{i,j})$ of the S-box, where the element $p_{i,j}$

[2]Permutation guarantees the output of individual S-boxes to be distributed to different S-boxes in the next round. This distribution has the effect of forcing certain patterns of change in the input (those where one or two bits change) to occur in the early rounds. Due to this effect, it is justified to use directed or minimal avalanche criteria to ensure that adequate avalanche will occur for those patterns of change.

is the probability that output variable y_j of the S-box changes when the input variable x_i is complemented. The average values, *i.e.* $(p_{i,1} + p_{i,2} + p_{i,3} + p_{i,4})/4$ of $p_{i,j}$ of S-boxes in DES and s^iDES are compared in Table 3.4.

Table 3.4: Average $(p_{i,j})$ of S-boxes.

box	DES	s^2DES	s^3DES	s^5DES
S1	0.620	0.495	0.609	0.633
S2	0.633	0.510	0.609	0.625
S3	0.661	0.505	0.617	0.620
S4	0.615	0.521	0.617	0.617
S5	0.633	0.516	0.617	0.641
S6	0.651	0.516	0.620	0.628
S7	0.656	0.516	0.638	0.630
S8	0.625	0.508	0.625	0.625

Table 3.4 clearly shows that DES S-boxes do not satisfy SAC. However, this does not mean that DES S-boxes are weak. For example, s^2DES, which Kim [68] derived by substituting DES S-boxes with S-boxes satisfying different criteria such as SAC, is proved to be weaker than DES from the viewpoint of differential cryptanalysis (see Knudsen [72]). Subsequently, Kim *et al.* [69],[70] constructed s^3DES as a cryptosystem which took countermeasures against differential cryptanalysis, and s^5DES as a cryptosystem which took countermeasures against differential and linear cryptanalyses. However, as obvious from Table 3.4, both cryptosystems have given up SAC.

3.5.4 Nonlinearity

There are some measures under the term nonlinearity, and it is not easy task to clarify them in a meaningful way.

Meier and Staffelbach [89] suggested that, in view of cryptographic design, it is important that properties still hold after applying simple or weak transformations to the function. They classified nonlinearity criteria into two in terms of the largest set of functions leaving a criterion invariant under all affine transformations: one is the maximum distance to affine functions and the other is the maximum distance to linear structures.

Preneel [111] considered modifications to the algebraic normal form of the function. An important case is a small change to the truth table. A design criterion is called *robust* if small changes to the truth table do not lead to large changes in the criterion.

In this section, we classify such measures into three categories: *algebraic nonlinearity, distance to affine functions*, and *distance to linear structures.*

Algebraic Nonlinearity

It seems natural to choose algebraic order for a measure of nonlinearity.

Definition 3.26 (Algebraic nonlinearity) *The algebraic nonlinearity $ord(f)$ of a Boolean function f is defined as the maximum of the order of its product terms that have a nonzero coefficient in the algebraic normal form.*

However, this measure is not robust, and a small modification to the truth table would result in a big difference. For example, let $g \in \mathcal{F}_n$ be an affine function. Then the function $f(\mathbf{x}) = g(\mathbf{x}) \oplus x_1 x_2 \cdots x_n$ satisfies $ord(f) = n$, although its truth table differs from that of an affine function.

In addition, it has been shown that a maximal order function is not balanced.

Theorem 3.7 *Let f be a Boolean function in $\mathcal{F}_n\,(n > 1)$. If $ord(f) = n$, $\hat{f}(0) \neq 0$*

O'Connor and Klapper [105] proved that the average algebraic nonlinearity of a Boolean function in $\mathcal{F}_n$ is $n + o(1)$, which implies that a randomly selected function might have high algebraic nonlinearity.

Distance to Affine Functions

Rueppel [118] suggested that the nonlinearity of a Boolean function as cryptographic function can be measured by the Hamming distance to the set of affine functions. The concept of nonlinearity measured in terms of distance to affine functions, however, seems to have been known to coding theorists well before cryptographers (see MacWilliams and Sloane [83]).

Definition 3.27 (Distance to affine functions) *The nonlinearity N_f of a function $f \in \mathcal{F}_n$ is defined by*

$$N_f = d(f, \mathcal{A}) = \min_{g \in \mathcal{A}} d(f, g) \tag{3.7}$$

where A is the set of all affine functions.

Pieprzyk and Finkelstein [109] extended this further to define the two measures of vector valued functions.

Definition 3.28 (Distance to affine functions I) *The nonlinearity N_F of a function $F \in \mathcal{F}_n^m$ is defined by*

$$N_F = \sum_{i=1}^{m} N_{p_i(F)} \tag{3.8}$$

where $F = [f_1, \cdots, f_m]$, $p_i(F) = f_i$.

Definition 3.29 (Distance to affine functions II) *The nonlinearity MN_F of function $F \in \mathcal{F}_n^m$ is defined by*

$$MN_F = \min_{i \in \{1,\cdots,m\}} N_{p_i(F)}. \tag{3.9}$$

In addition, they argued that the following definition should be applied because when F is a bijective function, the nonlinearity of its inverse function will also affect cryptosystem strength.

Definition 3.30 (Distance to affine functions III) *The nonlinearity NB_F of a bijective function $F \in \mathcal{F}_n^m$ is defined by*

$$NB_F = \sum_{i=1}^{m} (N_{p_i(F)} + N_{p_i(F^{-1})}. \tag{3.10}$$

One issue which should be addressed at this point is the degree of nonlinearity; how much nonlinearity is considered "enough," and are there any limits in achieving it? By translating the research results of coding theorists such as Cohen *et al.* [29] into the terminology of nonlinearity N_f, we have the following theorem (see Seberry *et al.* [124]).

Theorem 3.8 *For any function $f \in \mathcal{F}_n$, the nonlinearity of f, N_f satisfies $N_f \leq 2^{n-1} - 2^{n/2-1}$.*

In combinatorial theory, functions with this nonlinearity are known as *bent functions* defined by Rothaus [116]. Bent functions have ideal nonlinearity, although they are not balanced. Seberry *et al.* [124] showed that, if $f \in \mathcal{F}_n$ is limited to a balanced function, then the following theorem regarding the upper bound on nonlinearity holds.

Theorem 3.9 *Let f be a balanced function in $\mathcal{F}_n$ $(n \geq 3)$. Then the nonlinearity of f, N_f is given by*

$$N_f \leq \begin{cases} 2^{n-1} - 2^{n/2-1} & \text{for } n = 3, 5, 7, \cdots \\ \lfloor 2^{n-1} - 2^{n/2-1} - 2 \rfloor & \text{for } n = 4, 6, 8, \cdots \end{cases}$$

where $\lfloor x \rfloor$ denotes the maximum even integer less than or equal to x.

Pieprzyk and Finkelstein [109] applied nonlinearity NB_F to component bijective functions of DES S-boxes and to the randomly selected bijective and balanced functions, and compared them (see Fig. 3.4 and Fig. 3.5).

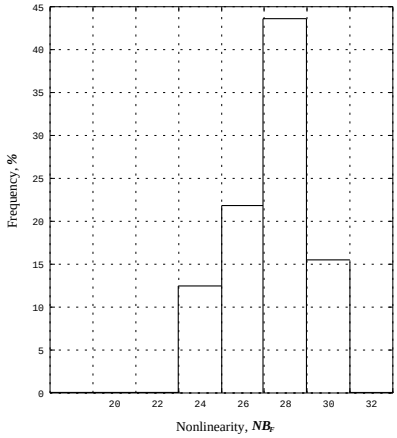

Figure 3.4: Distribution of nonlinearity (NB_F) of DES S-boxes.

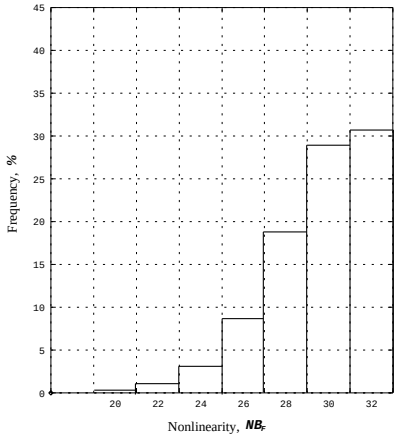

Figure 3.5: Distribution of nonlinearity (NB_F) of random bijective functions.

As obvious from Fig. 3.4 and Fig. 3.5, the distribution of the nonlinearity of bijective functions used in DES is more biased toward the lower end than that of randomly selected functions. Particularly, when comparing the ratio of bijective functions with maximum nonlinearity for both the original and the inverse forms, the former is 0% while the latter is 33%. Note, however, that all bijective functions are applied in the original form for both the enciphering and deciphering algorithm, and the inverse forms are never used. When we focus only on original forms, the difference between the above ratios will become smaller: 44% for the former and 57% for the latter (see Table 3.5). It should be noted that the nonlinearity of a vector function, suggested by Pieprzyk and Finkelstein [109], did not consider the nonlinearity of all linear combinations of output functions,

which limits the usefulness of this result as a measure to evaluate cryptographic strength. In other words, it deems difficult to derive any practical cryptanalysis just from the fact that the nonlinearity of DES S-boxes is relatively low from the viewpoint of N_F.

Table 3.5: Number of maximal nonlinearity (N_F) of S-boxes.

S-box	original	inverse
S1	4	0
S2	1	3
S3	0	1
S4	4	0
S5	1	0
S6	2	1
S7	1	1
S8	1	1
average	1.75	0.88

Distance to Linear Structures

At CRYPTO '84, Reeds and Manferdelli [114] devised an attack they called *cryptosystem factorization*. The idea is that there might be separate affine functions for the plaintext, ciphertext, and key which would, in the mapped domains, reduce the dimensionality of the key space (that is, in the mapped domain, certain key bits are degenerated). If this was the case, time complexity searching for the key space would be reduced. Reeds and Manferdelli [114] showed that, for DES, there is no such factorization of the round functions.

Chaum and Evertes [26] developed this idea. They defined *linear structures* and devised an effective attack for DES restricted to fewer than eight rounds. A function $f \in \mathcal{F}_n$ is said to have a linear structure $\mathbf{b}(\neq \mathbf{0}) \in Z_2^n$ if and only if $f(\mathbf{x}) \oplus f(\mathbf{x} \oplus \mathbf{b})$ is constant. If f has a linear structure, then there is a linear transformation that maps f onto a function that is linear in some of its input bits (see O'Connor and Klapper [104]).

Theorem 3.10 *Let* $\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_k}$ *be a set of linearly independent linear structures for the function* $f \in \mathcal{F}_n$, $1 \leq k \leq n$. *A nonsingular* $n \times n$ *matrix* A *over* Z_2 *exists such that if* $g(\mathbf{x}) = f(\mathbf{x}A)$, *then* $g(\mathbf{x})$ *is given as*

$$g(\mathbf{x}) = \sum_{i+1}^{k} x_i m_i \oplus g'(x_{k+1}, x_{k+2}, \cdots, x_n)$$

where $m_i = f(\mathbf{b_i}) \oplus f(\mathbf{0}) \in Z_2$, $1 \leq i \leq k$.

Let $\mathcal{LS}$ be the set of functions which have linear structures. Then we can define *distance to linear structures* as a measure of nonlinearity.

Definition 3.31 (Distance to linear structures) *The nonlinearity DLS_f of a function $f \in \mathcal{F}_n$ is defined by*

$$DLS_f = d(f, \mathcal{LS}) = \min_{g \in \mathcal{LS}} d(f, g) \tag{3.11}$$

As was pointed out by Nyberg [99], the linear structures of function f form linear subspace Z_2^n. The dimension of the subspace was called the *linearity dimension* of f.

With respect to the relationship among linear structure, algebraic nonlinearity, and completeness, Lai [79] showed the following two theorems.

Theorem 3.11 *For a function $f \in \mathcal{F}_n^m$, if $ord(f) = n$ then the function f has no linear structure.*

Theorem 3.12 *Let $f \in \mathcal{F}_n^n$ $(n \geq 3)$ be a bijective function such that, for any $n \times n$ nonsingular matrices A and M, the function $g(\mathbf{x}) = Mf(\mathbf{x}A)$ achieves completeness, then the function f has no linear structure.*

We have already discussed various nonlinearity measures and Theorem 3.13 summarizes the relationships among them (Seberry [123]).

Theorem 3.13 *Let $f, g \in \mathcal{F}_n$, A be a nonsingular $n \times n$ matrix over Z_2, and let $g(\mathbf{x}) = f(\mathbf{x}A)$. Then f and g have the same algebraic nonlinearity, nonlinearity (N_f), and linearity dimension.*

3.5.5 Measure of Differential Cryptanalysis

Differential Uniformity

Differential uniformity was proposed by Nyberg [99] as an easy measure to evaluate immunity against differential cryptanalysis at the S-box level.

Definition 3.32 (Differential uniformity) *A function $F \in \mathcal{F}_n^m$ is said to be differentially Δ_F-uniform, and accordingly, Δ_F is termed differential uniformity of F:*

$$\Delta_F = \max_{\alpha(\neq \mathbf{0}) \in Z_2^n, \beta \in Z_2^m} \#\{\mathbf{x} \mid F(\mathbf{x}) \oplus F(\mathbf{x} \oplus \alpha) = \beta\}.$$

It is generally suggested that the lower Δ_F is, the more F resists differential cryptanalysis. It is easy to see:

$$\Delta_F \geq \begin{cases} 2 & (n \leq m) \\ 2^{n-m} & (n > m) \end{cases}$$

However, this bound is not the best one; for example, if $\Delta_F = 2^{n-m}$, n must be even and $n \geq m$ (Nyberg [99]). Therefore, at this stage, we can only state that the differential uniformity for the functions in $\mathcal{F}_6^4$ such as the DES S-boxes is lower bounded by $4 (= 2^{6-4})$.

It should be noted that low differential uniformity is only a necessary, not a sufficient condition to guarantee immunity against differential cryptanalysis. Dawson and Tavares [39] proposed that the selection of S-boxes which have a flat differential distribution table would result in a cipher which is immune to differential cryptanalysis. On the contrary, Brown and Kwan [23] indicated that they are extremely weak to differential cryptanalysis. This was done by constructing a *2-round iterative characteristic* (see Section 4.2.1), with input XOR which changes bits to one S-box only. This is obvious since the flat difference distribution table implies that output XOR (which equals zero for a specified input) will occur with the probability of 2^{-n}. When being iterated over r rounds, this will have a probability of $2^{-n(r/2-1)}$, because you will get the last round for free. Consider a 16-round DES-like cryptosystem, but with S-boxes having a flat differential distribution table with $m = 6$, $n = 4$, and $r = 16$. This can be attacked by a 15-round characteristic, which alters inputs to a single S-box only. Then, probability for breaking this cipher is 2^{-28}, implying that about 2^{28} plaintext-ciphertext pairs are necessary to break the cipher, far easier than by exhaustive search.

Kim *et al.* [70] compared differential uniformity of DES S-boxes with those of s^2DES, s^3DES, and s^5DES, as shown in Table 3.6. The order of strength against differential cryptanalysis is shown as $s^2\text{DES} < \text{DES} < s^3\text{DES}$ in view of best characteristic probability (Sorimachi *et al.* [128]). But the differential uniformity of s^3DES is worse than in DES and s^2DES.

R-Robustness

Another measurement that takes into account the number of nonzero entries in the first column of a difference distribution table is called *robustness* as introduced by Seberry *et al.* [120].

Definition 3.33 (R-Robustness) *$F \in \mathcal{F}_n^m$ is said to be R_F-robust against differential cryptanalysis, where R_F is defined by*

$$R_F = \left(1 - \frac{N}{2^n}\right) \cdot \left(1 - \frac{\Delta_F}{2^n}\right) \tag{3.12}$$

Table 3.6: Δ_F of S-boxes.

S-box	DES	s^2DES	s^3DES	s^5DES
S1	16	14	20	18
S2	16	14	18	20
S3	16	14	18	20
S4	16	16	20	20
S5	16	16	20	20
S6	16	16	20	18
S7	16	16	20	18
S8	16	16	20	18
max	16	16	20	20
average	16.0	15.3	19.5	19.0

where N is the number of nonzero entries in the first column of the difference distribution table of F.

Kim *et al.* [70] compared the R-robustness of DES S-boxes to those of s^2DES, s^3DES, and s^5DES (see Table 3.7). Their results such as the R-robustness of DES is superior to that of s^2DES, and worse than that of s^3DES were consistent with the results of the previous strength evaluation derived from best characteristic probability.

Table 3.7: R-robustness of S-boxes.

box	DES	s^2DES	s^3DES	s^5DES
S1	0.32	0.33	0.30	0.36
S2	0.36	0.27	0.34	0.32
S3	0.32	0.28	0.37	0.30
S4	0.47	0.27	0.31	0.35
S5	0.39	0.27	0.36	0.31
S6	0.36	0.28	0.41	0.36
S7	0.34	0.32	0.32	0.36
S8	0.33	0.33	0.31	0.31
min	0.32	0.27	0.30	0.30
average	0.36	0.29	0.34	0.33

In order to make DES resistant to differential cryptanalysis, one has to bring the success probability of 2-round iterative characteristics below a certain level. Since such characteristics correspond to a non-zero input difference leading to a zero output difference (first column of the difference distribution table), R-robustness complemented by

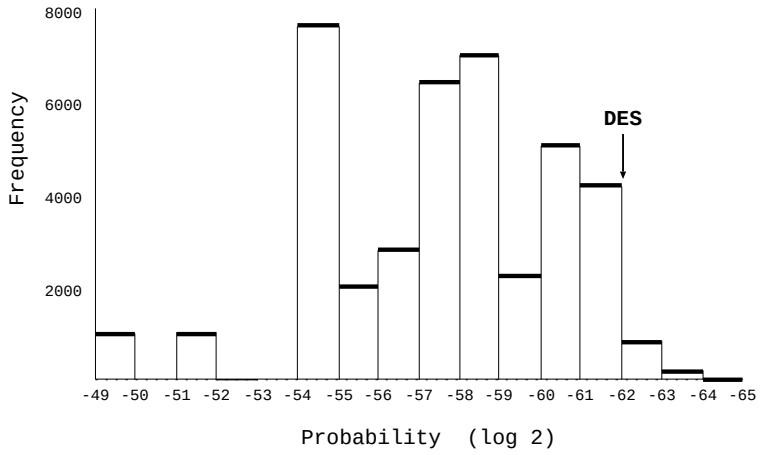

Figure 3.6: Distribution of 16-round characteristic probability by 2-round best iterative characteristics.

the characteristics seems to serve as a superior measure. However, in the case of DES, because of bit expansion E, some output bits are overlapping with other S-boxes, hence there are errors in measures which treat all input XORs equivalently. Moreover, if one focuses too much on 2-round iterative characteristics, the best characteristics become 3-round ones. Therefore, reducing these complex issues to just a multiplication of the number of nonzero entries in the first column and the maximum in the other columns seems to be an oversimplification. To this end, we hold a view that the results of Kim *et al.* [70], shown in Table 3.7, does not necessarily verify the validity of the R-robustness as a measure to evaluate cryptographic strength against differential cryptanalysis. It should be noted that one cannot thoroughly discuss resistance to differential cryptanalysis by only using the structure of S-boxes. Resistance to differential cryptanalysis can be significantly affected by some factors which lie beyond the fundamental structure of the S-box, such as the structure of the P-box. In fact, Matsui [86] showed that S-box order has a significant effect on the strength against differential cryptanalysis. As obvious from Fig. 3.6, DES S-boxes are arranged so as to be sufficiently resistant to differential cryptanalysis. Also, Brown *et al.* [23] indicated that the 3-round iterative characteristic is sensitive to the P-box form.

3.5.6 Measure of Linear Cryptanalysis

The measure Λ_F is used as a simple measure to evaluate strength against linear cryptanalysis at the S-box level.

Definition 3.34 (Linear uniformity) *Λ_F is defined by the following equation:*

$$\Lambda_F = \max_{\alpha \in Z_2^n, \beta(\neq \mathbf{0}) \in Z_2^m} | \#\{\mathbf{x} \mid Parity(\alpha \cdot \mathbf{x}) = Parity(\beta \cdot F(\mathbf{x}))\} - 2^{n-1} |. \tag{3.13}$$

It is suggested that the lower Λ_F is, the more F resists linear cryptanalysis. An upper bound of Λ_F was shown by Chabaud and Vaudenay [25] as follows.

$$\Lambda_F \geq \frac{1}{2} \left(3 \times 2^n - 2 - 2\frac{(2^n - 1)(2^{n-1} - 1)}{2^m - 1} \right)^{1/2}. \tag{3.14}$$

This bound is not the best either; they proved that it can be reached if and only if n is odd and $n = m$. In such case, the bound turns out to be $2^{\frac{n-1}{2}}$. The strict lower bound of Λ_F for a general case such as DES S-box which includes function $\mathcal{F}_6^4$ is unknown to date.

Table 3.8 compares Λ_F of each S-box of DES with those of s^2DES, s^3DES, and s^5DES. From the viewpoint of best linear characteristic probability, the order of strength against linear cryptanalysis is known as $s^3\text{DES} < \text{DES} < s^2\text{DES}$ (Sorimachi *et al.* [128]), and the maximum and average value of each Λ_F of four DES-like ciphers are in fact arranged according to this order. Also, s^5DES is designed so that the maximum value of Λ_F is restricted to under 16. In the table, maximal entry of S5 used in DES, 20, was first observed by Shamir [125].

Table 3.8: Λ_F of S-boxes.

box	DES	s^2DES	s^3DES	s^5DES
S1	18	14	16	16
S2	16	14	16	16
S3	16	14	16	16
S4	16	14	24	16
S5	20	18	24	16
S6	14	14	20	16
S7	18	16	20	16
S8	16	14	16	16
max	20	18	24	16
average	16.8	14.8	19.0	16.0

Needless to say, we cannot discuss strength against linear cryptanalysis based only on this measure. As in the case of differential cryptanalysis, the order of S-boxes is known to have a large effect on the strength of linear cryptanalysis (Matsui [86]). From Fig. 3.7, we can see that DES S-boxes are not arranged in order to be resistant to linear cryptanalysis.

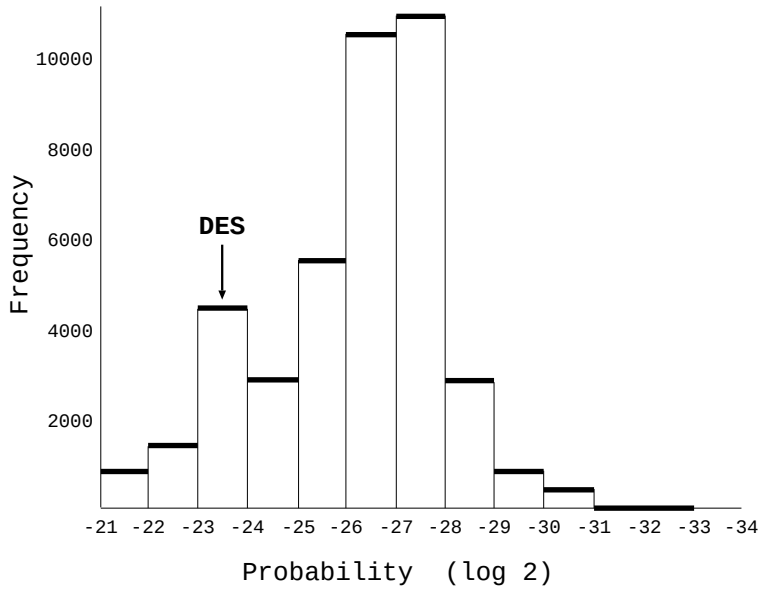

Figure 3.7: Distribution of the 16-round best linear characteristic probability.

3.6 Diffusion Components

In this section, we verify the structure of diffusion components such as P-box and key schedule based on the rate of achieving completeness, which is thought to be an important design criterion of diffusion components.

Kam and Davida [65] were the first to show that complete SP-networks can be constructed from small complete S-boxes. Their algorithm selects special permutations at each round of the product cipher. Subsequently, Ayoub [7] showed that similarly constructed product ciphers, employing only random substitutions, would almost guarantee the completeness of the product cipher. Ayoub [7] derived a combinatorial expression for the probability that a product cipher is complete, and demonstrated empirically that a randomly constructed product cipher would achieve completeness after a small number of rounds.

However, it is desirable that completeness is achieved as quickly as possible. Hence, we examine whether P-box and key schedule are well designed or not from the standpoint of the tempo of achieving completeness.

3.6.1 P-box

P-box diffuses the outputs from each S-box to the inputs of a number of S-boxes at the next stage. Design criteria of P-box have not been revealed, although from Table 3.9, Brown [19] derived the following set of empirical design rules:

Table 3.9: Empirical rules for designing P-box.

S-box	inputs from S-boxes						excluded
	a	b	c	d	e	f	S-box
1	7	4	2	5	6	8	3
2	6	8	3	7	5	1	4
3	5	1	4	6	7	2	8
4	7	2	5	8	3	1	6
5	3	1	2	6	4	8	7
6	4	8	7	1	3	5	2
7	3	5	4	8	2	6	1
8	2	6	3	1	7	4	5

RP-1. Each of the S-box input bits ab cd ef come from the outputs of different S-boxes.

RP-2. None of the input bits ab cd ef to a given S-box S(i) come from the output of the same S-box S(i).

RP-3. An output from S(i-1) goes to one of the ef input bits of S(i), and hence via E an output from S(i-2) goes to one of the ab input bits.

RP-4. An output from S(i+1) goes to one of the cd input bits of S(i).

RP-5. For each S-box output, two bits go to ab or ef input bits, the other two go to cd input bits as noted in [33].

Brown [19] compared DES P-box with P-boxes generated by empirical rules from the view of Meyer-Matyas' analysis [91] of ciphertext dependence on plaintext bits. Following Meyer-Matyas' approach, Brown's analysis can be briefly described as follows. To provide a measure of dependency, a 64×64 matrix $G_{a,b}$ is formed. Each element $G_{a,b}(i,j)$ specifies a dependency of output bit $X(j)$ on input bit $X(i)$, between rounds a and b. The number of marked elements in $G_{0,r}$ indicates the degree to which complete dependence was achieved by round r (details of the derivation of this matrix, and the means by which entries are propagated, can be found in Meyer-Matyas [91]). The analysis for the P-box

used in DES, the 178 empirically generated P-boxes, and the worst P-box (see Brown [19]), gave results as in Table 3.10. The P-box used in DES has a profile that falls fairly close to the median of the 178 P-boxes generated by empirical rules. In conjunction with the substantially inferior profile of the worst P-box, this provides a strong indication that the design rules identified are comprehensive, at least with this aspect of the P-box design.

Table 3.10: Dependency of ciphertext bits on plaintext bits.

round	DES P-box	empirical P-box	worst P-box
1	6.25	6.25	6.25
2	32.06	32.03-32.10	21.09
3	73.49	73.44-73.58	43.75
4	96.90	96.87-96.95	68.75
5	100.00	100.00	89.06
6	100.00	100.00	98.44
7	100.00	100.00	100.00

3.6.2 Key Schedule

The key schedule in the DES algorithm is responsible for forming the sixteen 48-bit subkeys K_i used in the rounds of the enciphering procedure. This function is important not only from the viewpoint of rate of achieving completeness but also the standpoint that if the same key is used on successive rounds (weak keys and semi-weak keys), it could weaken the resulting algorithm (see Meyer-Matyas [91]).

Concerning the latter, it is well known that only 4 *weak keys* and 12 *semi-weak keys* exist in the case of DES (see Davies [33]). Moore and Simmons [95] further investigated the properties of these keys and summarized the results as follows. For weak keys, enciphering equals deciphering: this means that $E(K, E(K, P)) = P$ for any plaintext P. Twelve semi-weak keys can be regarded as six pairs which satisfy $E(K_2, E(K_1, P)) = P$ for any P. Knudsen [73] identified yet other weak keys (*quasi weak keys*) which form a weak enciphering function class. Quasi weak keys are keys for which there exists a simple relation between the corresponding enciphering functions. Weak keys create few problems for enciphering with DES, but can be a problem for hash functions based on DES.

Brown [21] presented some empirical design rules for the key schedule. The rules presented for permutation $PC2$ are:

RP2-1. Bits permuted to the same S-box input are not closer than 3 bits apart.

RP2-2. Bits permuted to an S-box input must have a span from lowest to highest input bit number of at least 22 of the 28 bits in each key half.

RP2-3. Bits permuted to the selector bits a, f on a given S-box must not be adjacent in the sorted list of inputs.

RP2-4. Bits not selected by $PC2$ must be at least three places apart.

The design of the key schedule KS is obviously related to the design of $PC2$ by rules 1 and 4 given above. Brown [21] noted that key schedule KS ensures the following three properties:

RK-1. Each bit is used as input to each S-box.

RK-2. No bit is used as input to the same S-box on successive rounds.

RK-3. The total number of bits rotated is 56.

To verify the rate of achieving completeness, Meyer-Matyas [91] analyzed the dependence of ciphertext on key bits by using a similar analysis to that mentioned in the previous subsection. Table 3.11 shows that the dependency of every ciphertext bit on key bits rapidly increases and is dependent on all key bits after five rounds.

Table 3.11: Dependence of ciphertext on key bits.

round	dependence
1	5.36
2	44.87
3	87.72
4	98.21
5	100.00

Ciphertext bit dependency on key bits is dependent on the choices of P-box and $PC2$ as well as key schedule KS. To quantify this dependency, Brown and Seberry [21] employed a similar analysis as Meyer-Matyas' [91] mentioned above, *i.e.* a 64×56 matrix F_r is formed, and vector U is formed after $PC1$ is applied, *i.e.* $U = PC1(K)$. The number of marked elements in G_r will be examined to provide a profile of degree of dependence achieved by r round. Brown and Seberry [21] used some empirical rules mentioned above to generate a set of permutations $PC2$ (a total 7,315 permutations

Table 3.12: Dependency of ciphertext bits on plaintext bits.

round	DES $PC2$	empirical $PC2$	worst $PC2$
1	5.36	5.36	5.36
2	39.06	38.50-39.17	42.19
3	82.25	80.25-82.37	81.47
4	98.44	96.65-98.66	91.29
5	100.00	99.55-100.00	96.21
6	100.00	100.00	99.55
7	100.00	100.00	100.00

were found) and, given DES P-box and key schedule, compared DES $PC2$ with them (see Table 3.12).

Table 3.12 shows that the rate of achieving completeness of DES $PC2$ is very close to the maximum of 7,315 permutations generated. This fact leads us to think that DES $PC2$ is almost optimized in view of rate of achieving completeness.

3.6.3 Round Function

From the studies described in sections 3.6.1 and 3.6.2, we are sure that the structure of DES P-box and key schedule are excellent from the viewpoint of rate of achieving completeness, given the structure of the round function used in DES, the next step is to evaluate DES round function structure bearing in mind the rate of achieving completeness. The DES round function has a structure called *Feistel-type transformation* and the DES F-function a structure called the SP-network. We first reviewed studies which dealt with the rate of achieving completeness of Feistel-type transformation structures, followed by studies on the SP-network.

Feistel-type transformation

Due to the principle of a Feistel-type transformation which operates on the two halves L and R of an input block, at least three rounds are necessary for completeness. The internal state of a Feistel-type cipher is developed in Table 3.13.

A three-round Feistel cipher is complete if f_2 is complete, and if each of the output bits of f_3 depends on at least one of its input bits and each of the input bits of f_1 affects at least one of its output bits.

Table 3.13: Dependency of ciphertext bits on plaintext bits in the Feistel-type cipher.

round	left half	right half
0	L	R
1	R	$L \oplus f_1(R)$
2	$L \oplus f_1(R)$	$R \oplus f_2(L \oplus f_1(R))$
3	$R \oplus f_2(L \oplus f_1(R))$	$L \oplus f_1(R) \oplus f_3(R \oplus f_2(L \oplus f_1(R)))$

SP-network

Fumy [48] compared the rate of achieving completeness of DES with that of FEAL, which is also a Feistel-type cipher but whose F-function is not an SP-network. FEAL's F-function consists of modular arithmetic operations (addition modulo 2^8). He showed that FEAL is complete after four rounds whereas DES requires five to become a complete cipher (see Fig. 3.8).

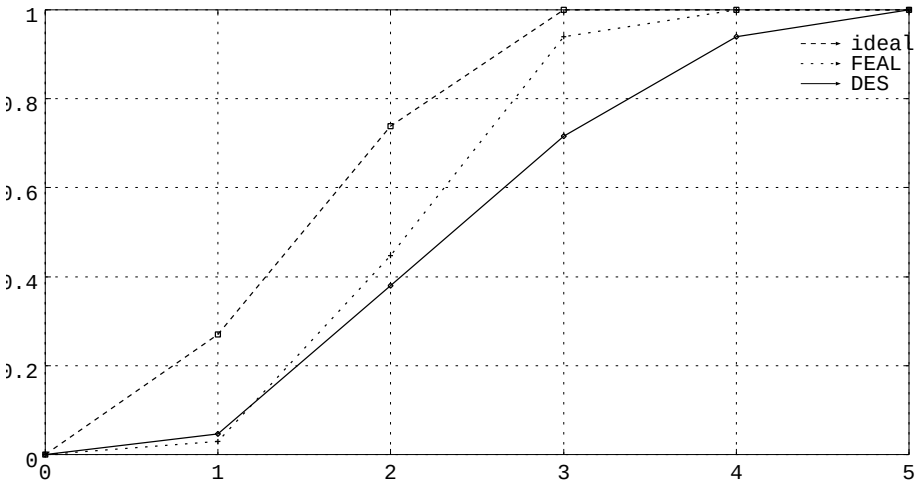

Figure 3.8: Rate of achieving completeness for DES and FEAL.

This result suggests that the SP-network is not necessarily the best F-function structure from just the viewpoint of rate of achieving completeness. However, it is difficult to construct the F-function satisfying desirable design criteria such as R-robustness and linear resistance when it consists of modular arithmetic operations. Actually, when one compares DES with FEAL-16 which have the same number of rounds, DES is much stronger against differential cryptanalysis, and slightly stronger against the linear cryptanalysis from the viewpoint of characteristic probability and linear characteristic prob-

ability, respectively. Such observation leads us to think that, given the structure of SP-network, DES's diffusion components are well designed from the viewpoint of rate of achieving completeness.

3.7 Statistical Properties of DES

If a cryptosystem has any statistical weakness within a smaller number of pairs of ciphertext and plaintext or key, it is considered to be badly designed. This section surveys studies on DES structure conducted from the viewpoint of the correlation between ciphertext bits and plaintext bits or key bits.

3.7.1 Partial Input-Output Dependence Test

If a small change in the key or plaintext were to produce a corresponding small change in the ciphertext with significant probability, this might be used to reduce the size of the key space to be searched. In this subsection, we review studies on the test if one of the following conditions hold: (i) some sets of output bits are dependent on some sets of input bits for fixed keys, and (ii) some sets of output bits are dependent on some sets of key bits for a fixed plaintext. To this end, χ^2 and Kolmogorov-Smirnov tests are available. According to Konheim [76], these tests were carried out both by IBM Research (as part of internal evaluation of DES) and by the NSA. Subsequently, Kim [67] compared DES with other DES-like cryptosystems, *i.e.* FEAL-8, MULTI2-8, s^2DES using these tests. We introduce their studies below.

Let a plaintext $\mathbf{x} = (x_1 x_2 \cdots x_r)$ be enciphered by any DES-like cryptosystem Enc with a key $\mathbf{k} = (k_1 k_2 \cdots k_r)$ into ciphertext $\mathbf{y} = (y_1 y_2 \cdots y_r)$ as

$$\mathbf{y} = Enc(\mathbf{k}, \mathbf{x}). \tag{3.15}$$

The χ^2 test

The procedure of this test is as follows: First, with one of two inputs (plaintext or key) fixed, we decide the subset (N_{in}) of input space and the subset (N_{out}) of output space. Next, we specify the null hypothesis H0: *there is no dependence between N_{in} and N_{out}*. After we compute the acceptance region at the given level of confidence, we count the number of pair occurrences, $Y_{i,j}$, between the two subsets with independent and identically distributed input at the predetermined number of sampling N_s. Then we compute the χ^2 statistic with degree of freedom, $m = 2^{|N_{in}|+|N_{out}|} - 1$ as

$$\chi^2 = \sum_{i=1}^{a} \sum_{j=1}^{b} \frac{(Y_{i,j} - N_s p)^2}{N_s p} \tag{3.16}$$

where $p = 2^{|N_{in}|+|N_{out}|}$, $a = 2^{|N_{in}|}$, and $b = 2^{|N_{out}|}$. We check whether this value exists inside the acceptance region at 1% and 99% confidence levels. If $m > 32$, the approximated χ^2 statistic is given by the formula (Abramowitz [1])

$$\chi^2 \simeq m\left\{1 - \frac{2}{9m} + x_q\sqrt{\frac{2}{9m}}\right\}^{\frac{1}{3}} \tag{3.17}$$

where, if the level of confidence is 0.01 to 0.99, x_q is -2.33 or 2.33, respectively (Knuth [75]).

Kim [70] checked the approximated χ^2 statistic with some DES-like cryptosystems: dependence between a subset $\{1,2,3\}$ of the input block and a subset $\{1,2,3\}$ of the output block with 500 samples. The average χ^2 statistics of different tests are compared in Table 3.14.

Table 3.14: Average of χ^2 statistics.

	DES	FEAL-8	MULTI2-8	s^2DES
fixed key	62.01	61.10	65.63	60.19
fixed plaintext	69.40	57.97	61.57	64.99

$N_{in} = \{1,2,3\}$, $N_{out} = \{1,2,3\}$, $N_s = 500$, and 20 trials

Moreover, Kim [70] expanded the size of the subset from 3 bits to both 4 bits and 1 byte and performed the χ^2 test on all pairs of input/output subsets exhaustively. He counted the number of times χ^2 statistics fell into the acceptance region at the 0.01 and 0.99 level of confidence. Table 3.15 shows the percentage of χ^2 statistics in the acceptance region. We can say that the null hypothesis can be partly acceptable for four DES-like cryptosystems, although, FEAL-8 and MULTI2-8 are much more vulnerable to differential and linear cryptanalyses than DES.

Table 3.15: % of χ^2 statistics in acceptance region.

size of subset	N_s	condition	DES	FEAL-8	MULTI2-8	s^2DES
4 bits	2000	A	97.77	98.92	98.24	98.31
		B	98.82	98.02	98.22	97.23
8 bits	327800	A	98.12	98.75	98.25	98.22
		B	98.02	98.71	96.09	98.74

A: fixed key; B: fixed plaintext

Kolmogorov-Smirnov test

In practice, the acceptance or rejection of the null hypothesis is based on the results of several independent χ^2 tests. The evaluation of multiple χ^2 tests is often made by using the Kolmogorov-Smirnov test.

Let $X_1, X_2, \cdots, X_n$ be random independent and identically distributed variables, and we assume that H0 behaves like continuous distribution:

$$F(\mathbf{x}) = Prob\{X \geq \mathbf{x}\}. \tag{3.18}$$

The discrete distribution F_n of multiple χ^2-statistics $X_1, X_2, \cdots, X_n$ is computed as:

$$F_n(x|X) = F_n(x|X_1, X_2, \cdots, X_n). \tag{3.19}$$

By the law of large numbers, we can confirm that equations (3.18) and (3.19) hold. We begin this test by sorting the sampled test statistics $X_1, X_2, \cdots, X_n$ in ascending order. Kolmogorov-Smirnov statistics $K_{n,+}$ and $K_{n,-}$ are defined by

$$K_{n,+} = \sqrt{n} \max_i((i+1)/n - F(X_i|X_1, X_2, \cdots, X_n)), \tag{3.20}$$

$$K_{n,-} = \sqrt{n} \max_i(F(X_i|X_1, X_2, \cdots, X_n) - (i/n)). \tag{3.21}$$

Intuitively, $K_{n,+}$ and $K_{n,-}$ will measure the positive and negative deviation of sample distribution function F_n from actual distribution F at points $X_1, X_2, \cdots, X_n$. We compute F_n as

$$F_n = erf\left[\frac{\sqrt{2m}}{X_i - m}\right] \tag{3.22}$$

where erf denotes the error function. When $n = 20$, the acceptance region of $K_{n,-}$ and $K_{n,+}$ is 0.038 and 1.468, respectively, at the 0.01 and 0.99 confidence level.

By using multiple χ^2 statistics, we performed the Kolmogorov-Smirnov test to verify the dependence between input and output block subsets. The range of $K_{n,-}$ and $K_{n,+}$ is summarized in Table 3.16 (Kim [67]). Finally, we could accept the null hypothesis that there is no dependence between the checked input and output block subsets.

However, considering the fact that FEAL-8 and MULTI2-8 are much more vulnerable to differential and linear cryptanalyses than DES from the viewpoint of characteristic and linear characteristic probability, we can conclude that the χ^2 and Kolmogorov-Smirnov tests to verify dependence between input and output block subsets, are not very useful for evaluating cryptosystem strength.

3.8 Conclusion

In this chapter, we examined the basic and internal structure of DES. The basic structure is deemed superior because of the following three properties: (i) DES is not closed and

Table 3.16: Range of $K_{n,+}$ and $K_{n,-}$.

size of subset	N_s	DES	FEAL-8	Multi2-8	s^2DES
4 bits	$K_{n,-}$	0.007-1.385	0.313-1.216	0.025-1.155	0.022-1.392
	$K_{n,+}$	0.017-1.415	0.142-1.354	0.022-1.417	0.047-1.444
8 bits	$K_{n,-}$	0.002-1.483	0.043-1.238	0.032-1.345	0.035-1.432
	$K_{n,+}$	0.025-1.446	0.029-1.372	0.029-1.315	0.012-1.462

generates a sufficiently large subgroup; (ii) OFB (output feedback) mode with $m = 64$ performs quite similar to random bijective functions; and (iii) resistance against differential and linear cryptanalyses can be achieved by increasing the number of rounds. Assessment of the internal structure, especially DES S-box, is rather complicated: some measures imply that it does not satisfy SAC (strict avalanche criteria) and that the distance to affine functions is relatively low; and other measures more related to actual cryptanalysis suggest that, except for the relatively poor performance against linear cryptanalysis, it contains favorable properties such as regularity, absence of linear structure, and high R-robustness. Since the size of DES S-boxes does not allow the simultaneous satisfaction of all these measures, we can deem DES S-boxes as being a well-balanced good structure. Also, examining the diffusion function of the P-box and key schedule from the viewpoint of rate of achieving completeness led us to the conclusion that, given the structure of the SP-network, these diffusion components are well designed.

Chapter 4

Evaluation of DES using Short-cut Methods

This chapter employs various *short-cut methods* to examine the cryptographic strength of DES. These methods try to minimize the time to find the correct key by exploiting the analytical and statistical characteristics of the enciphering algorithm. We focus on the evaluation of DES with the two most effective attacks, *i.e.* differential and linear cryptanalyses.

4.1 Various Attacks

4.1.1 Chosen Plaintext Attack using a Complementary Property

The chosen plaintext attack using complementary property reduces the time complexity of cryptanalysis of DES to about a half that via exhaustive search.

In 1976, Hellman *et al.* [53] indicated that DES exhibits the following property with respect to the complement,

$$C = E(K, P) \qquad \Longrightarrow \qquad \bar{C} = E(\bar{K}, \bar{P}). \tag{4.1}$$

where E, P, C, and K are the DES enciphering function, plaintext, ciphertext and key, respectively. This complementary property is easily deduced from the following property of the F-function.

$$\bar{f}(X_{i-1}, K_i) \;\; = \;\; f(\bar{X}_{i-1}, \bar{K}_i) \tag{4.2}$$

where X_i is 32-bit input data to the ith round F-function and $\bar{A}$ is the complement of A.

Let a cryptanalyst choose a complementary pair of plaintexts P_1 and $P_2 = \bar{P}_1$. Given their ciphertexts $C_1 = E(K, P_1)$ and $C_2 = E(K, P_2)$ under the same key K, the cryptanalyst searches for key K by trying all possible keys K'. For each key, he enciphers P_1 into C' by $C' = E(K', P_1)$. If $C' = C_1$, it is very likely that $K = K'$. In addition, the cryptanalyst can predict the ciphertext of P_1 under key $\bar{K}'$ to be $\bar{C}_2$ without an additional enciphering. If $C' = \bar{C}_2$, it is very likely that $K = \bar{K}'$, due to the complementary property $\bar{C}_2 = E(\bar{K}, P_1)$. Otherwise, neither K' nor $\bar{K}'$ can be the correct key K. Therefore, this attack can reduce the time complexity of cryptanalysis to about half that via exhaustive search.

4.1.2 Formal Coding Approach

The idea of the *formal coding approach* is as follows: if we have simple algebraic expressions of S-box outputs in terms of their inputs, we can obtain a complete algebraic expression of DES by progressive substitutions. This approach was initially suggested by Hellman *et al.* [53]. This expression could be viewed, in a known plaintext attack, as a set of Boolean equations to be solved for the keys. Schaumüller-Bichl [119] attempted to represent the S-boxes as modulo 2 sums of products of binary variables. This type of optimization is recognized as a difficult algebraic problem (see Davio [36]) and it has not been solved for functions of more than four variables. The nature of the heuristics introduced in the research of quasi-minimal expansions can substantially reduce the number of required terms. Schaumüller-Bichl finally concluded that their approach is impractical since it requires an enormous amount of computer memory.

Various attempts have been made to extend the Schaumüller-Bichl's research. The direct application of the pseudo-canonical expansion algorithm improved the representations of some of the 32 outputs of the S-boxes. An improved expansion method was devised by Hulsbosch; this method consisted of fixing some terms of the expansion of a particular output function and applying the pseudo-canonical expansion algorithm to the residual function. This technique improved the representation of 19 of the 32 S-box output functions (see Davio *et al.* [37]). The number of terms in the second function in S-box S7 improved by some 25%. On the average, reductions of 9% with respect to terms and 4.1% factors, were obtained. In spite of these improvements, the resulting expressions were still too complex to effect the complete set of substitutions implied by the formal coding.

4.1.3 Meet-in-the-Middle Attack

In 1985, Chaum and Evertse [26] presented a known plaintext attack, called the *meet-in-the-middle attack*. The attack on a DES-like cryptosystem composed of r-rounds can be

described as follows: Suppose a ciphertext C_0 and corresponding plaintext P_0. For each guessed key K_1 the cryptanalyst enciphers P_0 with the first s-rounds of DES yielding I'_0, and deciphers C_0 with the last $(r-s)$-rounds yielding I''_0. If $I'_0 = I''_0$, K_1 is the correct key. Compared with exhaustive search, guesses for the key are considerably fewer, when i and j are such that both the jth bits of I'_0 and I''_0 are independent of the ith key bit. Meet-in-the-middle attack can reduce time complexity, which is the number of DES enciphering operation (see Table 4.1), of searching DES with a small number of rounds. Chaum and Evertse also showed that a slightly modified version of DES with seven rounds could be solved with a time complexity of 2^{55}. However, they proved that it was impossible to apply the meet-in-the-middle attack to DES with eight or more rounds.

Table 4.1: Result of a meet-in-the-middle attack.

number of rounds	time complexity
4	2^{37}
5	2^{47}
6	2^{54}
7	–

4.1.4 Davies' Known Plaintext Attack

Davies' known plaintext attack exploits the correlation between the outputs of adjacent S-boxes. According to Biham and Shamir [10], Davies described a known plaintext attack on DES in 1987; recently, Davies and Murphy [35] released details. Given sufficient known plaintexts, this attack could yield 16 linear relationships among key bits, and reduce the subsequent key search to 2^{40}. Correlation between the outputs of adjacent S-boxes could reveal a linear relationship among the four bits of key used to modify these S-box input bits. This is because their inputs were derived from, among other things, a pair of identical bits produced by the bit expansion operation. The two 32-bit halves of the DES result (ignoring IP) would receive these outputs independently, so each pair of adjacent S-boxes could be exploited twice, yielding 16 bits of key information. The cryptanalysis requires a large number of known plaintexts P and corresponding $P \oplus C$. Since the S-box pairs vary according to the correlation they produce, the pair (S7,S8), for example, needs about $2^{56.6}$ samples while pair (S2,S3) needs about $2^{69.3}$ samples. With some $2^{85.6}$ samples, all but the pair (S3,S4) should give a total of 14 bits of key information. To exploit every pair, the cryptanalyst needs about 2^{86} samples. Although the S-boxes do not seem to have been designed to minimize the correlation, they are somewhat better than a random choice in this respect.

Subsequently, Biham and Biryukov [14] improved the Davies' attack and showed that it finds 6 effective bits from 2^{50} known plaintexts with a success rate of 51% with the rest of the 50 key bits being found by exhaustive search. This result, however, is not as good as the results we get using differential and linear cryptanalyses, which are shown in sections 4.2 and 4.3, respectively. Therefore, for the purpose of this report to confirm the cryptographic strength of DES, we will not go into further details of the Davies attack.

4.2 Differential Cryptanalysis

Differential cryptanalysis was discovered by Biham and Shamir [8], and was reported at CRYPTO '90. This is a chosen plaintext attack using some (generally fairly large numbers of) specific plaintexts and corresponding ciphertexts. The following notations are used in this section.

symbol	definition
P	64-bit data after the initial permutation; called *plaintext*
C	64-bit data before the final permutation; called *ciphertext*
P_L, C_R	left 32-bit data of P, C
P_R, C_L	right 32-bit data of P, C
C_i	64-bit output data of the ith round function
$C_{L,i}$	left 32-bit output data of the ith round function
$C_{R,i}$	right 32-bit output data of the ith round function
X_i	32-bit input data to the ith round F-function
Y_i	32-bit output data of the ith round F-function
K	correct key
K_i	the ith round 48-bit subkey
$f_i(X_i, K_i)$	ith round F-function
ΔA	XOR of the two values
B_x	hexadecimal expression of the value

The value of some bits of correct key is determined from some ciphertext pairs corresponding to pairs of chosen plaintexts with a specific XOR ΔP (they call it *characteristic*). The work factor of the cryptanalysis depends critically on the highest probability $Prob\{\Delta C_{r-1} = \beta \,|\, \Delta P = \alpha\}$ (we call such a pair (α, β) *differential*). In a DES-like cryptosystems, the ith round subkey K_i is entered into the F-function as XOR with the ith round input data X_i (more precisely, ith round input data which was increased to 48-bit by bit expansion). Therefore, the input XOR of the F-function is independent of the round subkey. Consequently, if we assume that subkeys are independent and uniformly random, output XOR of F-function would have a probability distribution which depends only on input XOR of the F-function. Subkeys are, in fact, neither independent

nor uniformly random. Although, as mentioned in Section 3.3, since the hypothesis of stochastic equivalence holds with high probability, we view it as not problematic to assume that $Prob\{\Delta C_{r-1} = \beta \,|\, \Delta P = \alpha\}$ is constant regardless of the value of the keys. The basic procedure of a differential cryptanalysis on an r-round iterated cipher can be summarized as follows:

Step 1 Find an $(r-1)$-round differential (α, β) such that $Prob\{\Delta C_{r-1} = \beta | \Delta P = \alpha\}$ has maximal or near maximal probability.

Step 2 Choose a plaintext P uniformly at random and compute P^* such that their XOR $\Delta P = \alpha$.

Step 3 After intercepting ciphertexts, $C = E(P, K)$ and $C^* = E(P^*, K)$, substitute these ciphertext pairs into the following expression, and derive some bits of a subkey K_r which make the expression hold with the highest probability:

$$\Delta C_{L,r-1} \oplus \Delta f(C_R, K_r) = \Delta C_L.$$

Step 4 After the subkey K_r is derived, the number of rounds of the cipher is thought to be reduced from r to $r-1$. Since $Prob\{\Delta C_{r-2} \,|\, \Delta P = \alpha\} \geq Prob\{\Delta C_{r-1} = \beta \,|\, \Delta P = \alpha\}$, proceed hereafter in a similar way.

4.2.1 Best Characteristic and Its Probability

As shown in Section 3.3.4, if we assume that the subkeys are independent and uniformly random, it follows that the probability of an r-round differential (β_0, β_r) is

$$Prob\{\Delta C_r = \beta_r \,|\, \Delta P = \beta_0\} = \sum_{\beta_1} \sum_{\beta_2} \cdots \sum_{\beta_{r-1}} \prod_{i=1}^{r} Prob\{\Delta C_i = \beta_i \,|\, \Delta C_{i-1} = \beta_{i-1}\},$$

where $\Delta C_0 = \Delta P$.

However, maximal value (we term it *best differential probability*) of the above formula is very difficult to derive, and only the following theorem has been proved for a DES-like cryptosystem (Nyberg and Knudsen [100]).

Theorem 4.1 *Assume that in a DES-like cipher with F-function $f : Z_2^m \to Z_2^n$ $(m \geq n)$, subkeys are independent and uniformly random. Then the best differential probability of an r-round, $r \geq 4$, is less than or equal to $2 \cdot DCP_{max}^2$,*

where $DCP_{max} = \max_{\alpha \neq 0, \beta} Prob\{\Delta Y = \beta \mid \Delta X = \alpha\}$.

Now, probability DCP_{alg} (*best characteristic probability*), defined by the formula below, can be regarded as an approximation to best differential probability:

$$DCP_{alg} = \max_{\Delta P \neq 0, \Delta C_1, \cdots, \Delta C_r} \prod_{i=1}^{r} Prob\{\Delta C_i = \beta_i \mid \Delta C_{i-1} = \beta_{i-1}\},$$

where $\Delta C_0 = \Delta P$.

In this section, we introduce studies on the best characteristic probability of DES.

When both inputs to the F-function are equal, both outputs are also equal (*zero-round*). In other words, an input XOR equal to zero leads to an output XOR which equal to zero with probability 1. Knudsen [72] proposed an approach which tries to maximize the number of zero-rounds in order to find the best characteristics.

Two consecutive zero-rounds of a characteristic of a DES-like cryptosystem lead to equal inputs and outputs in all rounds. The maximum occurrence of zero rounds therefore is in every second round. When zero rounds occur every n-rounds, it was termed an *n-round iterative characteristic* by Knudsen [72].

- 2-round iterative characteristic

$$\begin{array}{cccc} & & (\phi, 0) & \\ 1\text{R} & 0 & \longleftarrow & 0 \\ 2\text{R} & 0 & \longleftarrow & \phi \\ & & (0, \phi) & \end{array}$$

- 3-round iterative characteristic

$$\begin{array}{cccc} & & (\gamma, 0) & \\ 1\text{R} & 0 & \longleftarrow & 0 \\ 2\text{R} & \phi & \longleftarrow & \gamma \\ 3\text{R} & \gamma & \longleftarrow & \phi \\ & & (0, \phi) & \end{array}$$

- 4-round iterative characteristic

$$\begin{array}{cccc} & & (\gamma, 0) & \\ 1\text{R} & 0 & \longleftarrow & 0 \\ 2\text{R} & \phi & \longleftarrow & \gamma \\ 3\text{R} & \gamma \oplus \psi & \longleftarrow & \phi \\ 4\text{R} & \phi & \longleftarrow & \psi \\ & & (0, \psi) & \end{array}$$

By comparing the difference distribution table for the eight S-boxes, we can easily find the best possibilities. The 2-round best iterative characteristics are $\phi^* = 19600000\ 00000000_x$ and $1B600000\ 00000000_x$ with probability $(14/64) \times (8/64) \times (10/64) \simeq 1/234$, as Biham and Shamir [10] showed. From the fact that 3-round and 4-round iterative characteristics contain only one zero-round, it is easy to prove that 3- and 4-round iterative characteristics, which are better than 2-round best iterative characteristics, do not exist in the case of DES. Knudsen could not prove for n-round or more characteristics, $n > 4$, which have probability higher than 2-round best iterative characteristics concatenated with themselves $n/2$ times, but suggested that it would be proved in the future.

Subsequently, Matsui [86] proposed a realistic algorithm which derived the best characteristic for DES-like cryptosystems. This algorithm assumes that the best characteristics of a cryptosystem up to the $(r-1)$-round are known and derives r-round best characteristic. The algorithm proved that the 2-round best iterative characteristics concatenated with themselves $r/2$ times are the best r-round characteristic. According to this algorithm, the probability of the best characteristic and the number of rounds are as in Table 4.2.

Table 4.2: Best characteristic probability and number of rounds.

rounds	best characteristic	probability
11	$1B600000\ 00000000$	1.57×2^{-40}
12	$00084010\ 1B600000$	1.71×2^{-47}
13	$1B600000\ 00000000$	1.71×2^{-48}
14	$00084010\ 1B600000$	1.87×2^{-55}
15	$1B600000\ 00000000$	1.87×2^{-56}

4.2.2 Attack Method

A high probability characteristic will make it possible to look for a particular number of bits in the subkeys of the last round (all the bits that enter some particular S-boxes). We must note that it is not necessary to count all possible subkey bits. The advantage in counting all possible subkey bits is that it will be a good identification of the correct key value. However, counting the number of occurrences of all the possible values of a large number of bits usually needs a huge memory. We can count a small number of S-boxes, and use all the other S-boxes only to identify and discard the wrong pairs where input XORs in such S-boxes cannot lead to expected output XORs.

The required number of plaintext-ciphertext pairs depends on the probability of the characteristic, the number of key bits that we count, and the degree of identification of

wrong pairs that can be discarded before counting. The following definition introduced by Biham and Shamir [10] gives us a tool to evaluate the feasibility of a counting scheme based on a characteristic.

Definition 4.1 (Signal-To-Noise Ratio) *The ratio between the number of correct pairs and the average count in a counting scheme is called the signal-to-noise ratio of the counting scheme and is denoted by S/N.*

If we are looking for k key bits, then we count the number of occurrences of 2^k possible key values in 2^k counters. The counters contain an average count of $m \cdot \alpha \cdot \beta / 2^k$ counts, where m is the number of pairs, α the average count per counted pair, and β the ratio of counted pairs to total. The correct key value is counted about $m \cdot p$ times using the correct pairs, where p is the characteristic's probability plus the random counts estimated for all possible keys. The signal-to-noise ratio of a counting scheme is therefore

$$S/N = \frac{m \cdot p}{m \cdot \alpha \cdot \beta / 2^k} = \frac{2^k \cdot p}{\alpha \cdot \beta}.$$

Usually we relate the required number of pairs by a counting scheme to the required number of correct pairs. The required number of right pairs is mainly a function of the signal-to-noise ratio. When S/N is high enough, only a few occurrences of correct pairs are needed to identify the correct values of the subkey bits.

Biham and Shamir [10] showed three types of attacks. We will next introduce two basic attacks of differential cryptanalysis: *2R-attacks* and *1R-attacks* (see Biham and Shamir [10] for details).

2R-Attacks

In 2R-attacks, counting can be made for all bits of the subkey of the last round. Possibility checks can be effected for all S-boxes in the previous rounds. S-boxes with zero input XOR should also have an output XOR of zero, and the success rate of this check is 1/16. For other S-boxes the success rate is about 0.8.

In DES reduced to 13 rounds, the 48 bits of K_{13} can be found by using 2^{43} pairs with an 11-round characteristic. We know that

$$\begin{array}{lrcl} & & (\phi, 0) & \\ 11\text{R} & 0 & \longleftarrow & 0 \\ 12\text{R} & \Delta O_{12} = \Delta I_{13} \oplus \Delta I_{11} = \Delta C_R & \longleftarrow & \phi \\ 13\text{R} & \Delta O_{13} = \Delta I_{12} \oplus \Delta C_L = \Delta C_L \oplus \phi & \longleftarrow & \Delta I_{13} = \Delta C_R \\ & & (C_L, C_R) & \end{array}$$

We can check that $\Delta I_{12} \rightarrow \Delta O_{12}$ and $\Delta I_{13} \rightarrow \Delta O_{13}$, and count the possible occurrences of the key bits. At $\Delta I_{12} \rightarrow \Delta O_{12}$, the input and output XORs of five S-boxes must be zero (which happens for wrong pairs with probability 1/16), while the input and output XORs of three other S-boxes must be expected values (which happens for wrong pairs with a probability of 0.8). Then the counting on 30 bits of K_{13} has $S/N = 2^{30} \cdot 2^{-40} / \{4^5 \cdot 0.8^3 \cdot 0.8^3 \cdot (1/16)^5\} \simeq 4$. Therefore 13-round DES can be broken by an array of size 2^{30} and by using 2^{43} pairs. A 15-round DES can be broken using the 13-round characteristic with an array of size 2^{42} and $S/N = 2^{42} \cdot 2^{-48} / \{4^7 \cdot 0.8 \cdot 0.8^3 \cdot (1/16)^5\} \simeq 2.5$ and by using 2^{51} pairs. This time complexity is still smaller than that of the exhaustive search, but the space requirement is unrealistic.

1R-Attacks

In 1R-attacks, counting can be done on all bits of the subkey of the last round entering S-boxes with nonzero input XORs. Verification of the values of ΔC_R itself and possibility checks on all the other S-boxes in the last round can be effected. For those S-boxes with a zero input XOR, the output XOR should be zero too and the check success rate is 1/16. Since the input XOR is constant, there are several subkey values which we cannot distinguish. However, the number of such values is small and each of them can be checked later in the next part of the algorithm.

A 14-round DES can be broken by using the 13-round characteristic where

$$
\begin{array}{lcccc}
 & & & (0,\phi) & \\
12\mathrm{R} & & 0 & \longleftarrow & \phi \\
13\mathrm{R} & & 0 & \longleftarrow & 0 \\
14\mathrm{R} & \Delta O_{14} = \Delta C_L \oplus \Delta O_{13} = \Delta C_L & & \longleftarrow & \Delta I_{14} = \phi = \Delta C_R \\
 & & & (\Delta C_L, \Delta C_R) &
\end{array}
$$

We can easily identify the right pairs. Pairs which satisfy $\Delta C_R = \phi$ and the 20 bits in C_L going out of S4,$\cdots$, S8 are zero. This also holds for 2^{-52} of the wrong pairs. For the other three S-boxes we count the possible values of their 18 key bits with $S/N = 2^{18} \cdot 2^{-48} / (4^3 \cdot 2^{-52}) = 2^{16}$. Thus we need 2^{50} pairs.

For a full 16-round DES we get $S/N = 2^{18} \cdot 2^{-56} / (4^3 \cdot 2^{-52}) = 2^8$ by using the 15-round characteristic. However, this needs 2^{57} pairs which is larger than the time complexity of the exhaustive search.

At this time, differential cryptanalysis was not effective against DES, although later Biham and Shamir [11] proved its effectiveness by using an attack called the *filtering method*.

Attacks using the Filtering Method

We can generate without loss of probability pairs of plaintexts, whose XORed inputs $(\phi, 0)$ fits into the 13-round characteristic of rounds 2 to 14 using this filtering method. This attack is divided into data collection and data analysis phases.

Step 1 **Data Collection Phase**

Let $(\phi, 0)$ and $v_0, \cdots, v_{4095}$ be the 2-round best iterative characteristic and the 2^{12} 32-bit constants, which consist of all the possible values at the 12 bit positions XORed with the 12 output bits of S1, S2, and S3 after the first round. We define a structure which consists of 2^{13} plaintexts as:

$$\begin{array}{ll} P_i = P \oplus (v_i, 0) & P_i^* = (P \oplus (v_i, 0)) \oplus (0, \phi) \\ C_i = DES(K, P_i) & C_i^* = DES(K, P_i^*) \end{array}$$

where $0 \leq i < 2^{12}$.

The plaintext pairs we are interested in are all pairs P_i, P_j^* with $0 \leq i, j < 2^{12}$. There are 2^{24} such plaintext pairs, and their XORs are always in the form (v_k, ϕ), where each v_k occurs exactly 2^{12} times. Since the actual processing of the left half of P and the left half of P XORed with ψ in the first round under the actual key create an XORed value after the first round (which can be non-zero only at the outputs of S1, S2, and S3), this XORed value is one of the v_k. As a result, for exactly 2^{12} plaintext pairs, the output XOR of the first F-function is canceled out by XORing it with the left half of plaintext XOR, and thus output XOR of the first round is the desired input XOR $(\phi, 0)$ into the iterative characteristic. Therefore, each structure has a probability of about $2^{12} \cdot 2^{-47.2} = 2^{-35.2}$ to contain a right pair. The problem of this approach is that we do not know the actual value of v_k. Trying all the 2^{24} possible pairs would take too much time, but we can use their cross-product structure to isolate the right pairs in just 2^{12} times. For any right pair, output XOR after 16 rounds should be zero at the five outputs of S4,$\cdots$, S8. We can thus obtain two groups of 2^{12} ciphertexts C_i, C_j^* by these 20 bit positions, and detect all the repeated occurrences of values among the 2^{24} ciphertext pairs in about 2^{12} time. Any pair of plaintexts which fails to pass this test would have a non-zero ciphertext XOR at those 20 bit positions, and thus, by definition, cannot be a right pair. Since each one of the 2^{24} possible pairs passes this test with probability 2^{-20}, we expect about $2^4 = 16$ pairs to survive. By testing other S-boxes in the first, 15th, and 16th rounds and eliminating all the pairs whose XOR values have been indicated as impossible in the difference distribution tables, we

can discard about 92.55% of these surviving pairs, leaving only $16 \cdot 0.0745 = 1.19$ pairs.

Step 2 **Data Analysis Phase**

A key value is suggested when it can create the output XOR values of the last round as well as the expected output XOR of the first round and 15th round for particular plaintext pairs and ciphertext pairs. In the first and the 15th round, the input XORs of S4,$\cdots$, S8 are always zero. Due to the key scheduling algorithm, all the 28 bits of the left key register are used as inputs for S-boxes S1, S2, and S3 in the first and 15th rounds, and S1,$\cdots$, S4 in the 16th round. Only 24 bits of the right key register are used in the 16th round. Thus, $28 + 24 = 52$ key bits enter these S-boxes. By comparing the output XOR of the last round, the three S-boxes of the first round, and the three S-boxes of the 15th round to the expected values, we can derive 0.84 key candidates from 2^{52}. Therefore, each structure suggests about $1.19 \cdot 0.84 \cdot 16 = 16$ choices for the whole key. After two additional rounds, we can verify each key by performing about one quarter of DES enciphering, leaving only about 2^{-12} of the choices of the key. This filtering costs about $16 \cdot (1/4) = 4$ equivalent DES enciphering.

We can summarize the performance of the above attack in the following way. Each structure contains a right pair with probability $2^{-35.2}$. The data collection phase enciphers a pool of about 2^{35} structures, which contain about $2^{35} \cdot 2^{13} = 2^{48}$ chosen plaintexts, and about $2^{35} \cdot 1.19 = 2^{35.25}$ pairs (2^{36} ciphertexts) remain as candidate inputs for the data analysis phase. The probability that at least one of them is a right pair is about 58%, and the analysis of any right pair is guaranteed to lead to the correct key. The time complexity of this data analysis phase is about $2^{35} \cdot 4 = 2^{37}$ equivalent DES operations.

In order to further reduce the number of chosen plaintexts, we can use the *quartet method* (Biham and Shamir [10]). Since the basic collection of plaintexts in the above attack is a structure rather than a pair, we create metastructures which contain 2^{14} chosen plaintexts, built from the two best 2-round iterative characteristics, *i.e.* $1B600000\ 00000000$. With this metastructure, we obtain two times as many pairs, and thus reduce the number of chosen plaintexts enciphered in the data collection phase from 2^{48} to 2^{47}.

Knudsen [74] has found an extension of differential cryptanalysis against DES by focusing on a special differential called the *truncated differential*, which predicts only parts of 64 bit value.

Attacks using Truncated Differentials

By utilizing differential cryptanalysis using truncated differentials against DES, Knudsen [74] showed that a 6-round DES could be analyzed using only 46 chosen plaintexts with an expected running time of about 3,500 encryptions. This result is the best performance among known attacks against 6-round DES. It is an open question as to whether truncated differentials can improve attacks on DES for more than 6 rounds.

Table 4.3 summarizes the cryptographic strength of DES, approximately measured by best characteristic probability; the table also shows cryptographic strength of popular common key ciphers (other than DES) against differential cryptanalysis.

Table 4.3: Strength of common key cipher of DES *et al.* against differential cryptanalysis.

cipher	chosen plaintexts	time complexity	reference
DES	2^{47}	2^{37}	Biham and Shamir [11]
FEAL-8	2^{8}	2 min by PC	Biham and Shamir [12]
FEAL-16	2^{29}	n.a.	Biham and Shamir [9]
Multi2-8	$2^{16.3}$*	10 min by WS	Matsui and Yamagishi [85]
LOKI89	2^{57}**	n.a.	Tokita *et al.* [133]
LOKI91	2^{64}**	n.a.	Tokita *et al.* [133]

* Matsui and Yamagishi [85], improved a differential cryptanalysis based on Gilbert and Chassé [51] and conducted an attack using it.
** Inverse of 13-round characteristic probability shown to reference [133] used as an approximation.

4.2.3 Conclusion of Differential Cryptanalysis

Generally speaking, newly proposed cryptanalysis is immature when first presented, and gradually matures as it incorporates improvements. It is about six years since differential cryptanalysis was first reported in 1990, and epoch-making improvements have been few except for the filtering method and attacks using truncated differentials on 6 round DES.

When cryptanalyzing DES by differential cryptanalysis, which uses a filtering method, the required number of chosen plaintexts, approximated by the best characteristic probability, will be about 2^{47}. In order to obtain this many chosen plaintexts, it would take about a year even under the most favorable circumstances: *i.e.* ciphertexts corresponding to chosen plaintexts are continuously transmitted through the current fastest 200Mbps communication path.

Needless to say, we cannot discuss the real cryptographic strength of DES against differential cryptanalysis, because we do not know the precise best differential probability. If combined with the next section's results, it will become obvious that the DES structure is more vulnerable to linear cryptanalysis than differential cryptanalysis. For the purpose of this report, it will be sufficient if we can confirm the cryptographic strength of DES against linear cryptanalysis, and we will not go into further details of strength against differential cryptanalysis.

4.3 Linear Cryptanalysis

Linear cryptanalysis was discovered by Matsui [84] and was presented at EUROCRYPT '93. This cryptanalysis is a known plaintext attack. In this cryptanalysis, some (generally fairly large numbers of) random plaintext-ciphertext pairs are used to determine the value of some bits of the correct key. The following notations are used in this section. Note that the extreme right bit of each is referred as the 0th (lowest).

symbol	definition
P	64-bit data after the initial permutation; called *plaintext*
C	64-bit data before the final permutation; called *ciphertext*
P_L, C_R	left 32-bit data of P, C
P_R, C_L	right 32-bit data of P, C
C_i	64-bit output data of the ith round function
K	correct key
K_i	the ith round 48-bit subkey
$f_i(X_i, K_i)$	ith round F-function
$A[i]$	ith bit of symbol A
$A[i_1, i_2, \cdots, i_a]$	$A[i_1] \oplus A[i_2] \oplus \cdots \oplus A[i_a]$
ΓB	masking value of B
N	number of given plaintext-ciphertext pairs

A condition for applying linear cryptanalysis to a DES-like cipher is the existence of "effective" linear expressions. A linear expression is shown as follows:

$$P[i_1, i_2, \cdots, i_a] \oplus C[j_1, j_2, \cdots, j_b] = K[k_1, k_2, \cdots, k_c], \tag{4.3}$$

with $i_1, i_2, \cdots, i_a, j_1, j_2, \cdots, j_b$ and $k_1, k_2, \cdots, k_c$ fixed bit locations. The effectiveness of such a linear expression is given by $|p - 1/2|$, where p is the probability that the expression holds. By checking the value of the left-hand side of the expression (4.3) for a large number of plaintext-ciphertext pairs, the right-hand side can be guessed by taking the value that occurs most often. This gives a single bit of information about the key.

For a practical known plaintext attack on DES, the best 14-round expression is used; *i.e.*, the first and last round are enciphered using K_1 and deciphered using K_{16}, respectively. Consequently, the following expression is obtained:

$$P[i_1, i_2, \cdots, i_a] \oplus C[j_1, j_2, \cdots, j_b] \oplus f_1(P_L, K_1)[l_1, l_2, \cdots, l_d] \oplus \\ f_{16}(C_H, K_{16})[m_1, m_2, \cdots, m_e] = K[k_1, k_2, \cdots, k_c]. \tag{4.4}$$

In addition to the inferred single bit of information, we can further infer from the formula some bits of information related to subkeys K_1 and K_{16} which affect the value of the left-hand side. This is the basic principle of linear cryptanalysis.

4.3.1 Best Linear Expression and Its Linear Characteristic Probability

Matsui [88] defined each probability of single round linear expression as follows and called it *linear characteristic probability.*

$$LCP(\Gamma C_i \rightarrow \Gamma C_{i-1}) = |2Prob\{C_{i-1} \cdot \Gamma C_{i-1} = C_i \cdot \Gamma C_i\} - 1|^2$$

Cryptographic strength against linear cryptanalysis must be assessed by the following value, which Matsui [88] called *best average linear probability.*

$$ALP_{max} = \max_{\Gamma C_r \neq 0, \Gamma P} \sum_{\Gamma C_{r-1}} \sum_{\Gamma C_{r-2}} \cdots \sum_{\Gamma C_1} \prod_{i=1}^{r} LCP(\Gamma C_i \rightarrow \Gamma C_{i-1})$$

However, it is difficult to derive best average linear probabilities, and only the following theorem has been proved for a DES-like cryptosystem (Nyberg [101]).

Theorem 4.2 *It is assumed that in a DES-like cipher with F-function $f : Z_2^m \rightarrow Z_2^n$ $(m \geq n)$ the subkeys are independent and uniformly random. Then the best average linear probability of an r-round, $r \geq 4$, is less than or equal to $2 \cdot LCP_{max}^2$,*

here $LCP_{max} = \max_{\Gamma C_i \neq 0, \Gamma C_{i-1}} LCP(\Gamma C_i \rightarrow \Gamma C_{i-1})$.

In this section, we consider the maximal products of linear characteristic probability LCP_{alg} (we call it *best linear characteristic probability*) as an approximation of best average linear probability and use this to approximately assess the strength of DES.

$$LCP_{alg} = \max_{\Gamma C_i \neq 0, \Gamma C_{i-1}} \prod_{i=1}^{r} LCP(\Gamma C_i \rightarrow \Gamma C_{i-1}). \tag{4.5}$$

Let us explain the composition of r-round linear expressions which have best linear characteristic probability expression (*r-round best linear expression*) according to Matsui [84]. The r-round best linear expression is constructed by accumulating single-round linear expressions. Examples of single-round linear expressions are as follows.

symbol	single round linear expression	LCP
A	$X[15] \oplus F(X,K)[7,18,24,29] = K[22]$	$(40/64)^2$
B	$X[27,28,30,31] \oplus F(X,K)[15] = K[42,43,45,46]$	$(20/64)^2$
C	$X[29] \oplus F(X,K)[15] = K[44]$	$(4/64)^2$
D	$X[15] \oplus F(X,K)[7,18,24] = K[22]$	$(20/64)^2$
E	$X[12,16] \oplus F(X,K)[7,18,24] = K[19,23]$	$(32/64)^2$

Accumulate single-round linear expressions to satisfy the following formula (restriction by this formula is termed the *concatenation rule*):

$$\Gamma f(X_{i+2}, K_{i+2}) = \Gamma f(X_i, K_i) \oplus \Gamma X_{i+1} \quad (1 \leq i \leq 14).$$

Out of the derived r-round linear expressions, one which has the maximal product of each single-round linear expression's characteristic probability will become the r-round best linear expression. Matsui [86] derived the r-round best linear expression of DES by using the best expression search algorithm (see Table 4.4).

The success probability $P(S)$ when we use the maximum likelihood to solve the best linear expression, is calculated by

$$P(S) = \int_{-\sqrt{N}LCP_{alg}}^{\infty} \frac{1}{2\pi} e^{-x^2/2}\, dx. \tag{4.6}$$

Table 4.5 shows a numerical calculation on a 14-round best linear expression.

4.3.2 Procedure for Key Search

The following 14-round best linear expression, whose $LCP_{alg} = 1.42 \times 2^{-40}$, is used in order to break DES by linear cryptanalysis.

$$\begin{aligned} &P_R[\delta] \oplus C_L[\alpha] \oplus C_L[15] = \\ &\quad K_2[22] \oplus K_3[44] \oplus K_4[22] \oplus K_6[22] \oplus K_7[44] \oplus K_8[22] \oplus K_{10}[22] \\ &\quad \oplus K_{11}[44] \oplus K_{12}[22] \oplus K_{14}[22] \end{aligned} \tag{4.7}$$

where $\delta = 7, 18, 24$, $\alpha = 7, 18, 24, 29$.

By applying expression (4.7) to the 14 F-functions from the 2nd to the 15th round, we have :

$$\begin{aligned} &P_R[\delta] \oplus f_1(P_R, K_1)[\delta] \oplus C_L[15] \oplus C_R[\alpha] \oplus f_{16}(C_R, K_{16})[15] = \\ &\quad K_3[22] \oplus K_4[44] \oplus K_5[22] \oplus K_7[22] \oplus K_8[44] \oplus K_9[22] \oplus K_{11}[22] \\ &\quad \oplus K_{12}[44] \oplus K_{13}[22] \oplus K_{15}[22], \end{aligned} \tag{4.8}$$

We can write the plaintext-ciphertext bits and subkey bit which affect the left-hand side of the formula more precisely.

Table 4.4: Best linear expression and linear characteristic probability of DES.

rounds	best linear expression	LCP_{alg}	characteristic
11	$P_L[\alpha] \oplus P_R[15] \oplus C_L[\alpha] \oplus C_R[15]$ $= K_1[22] \oplus Q_3 \oplus Q_7 \oplus K_{11}[22]$	1.82×2^{-29}	A-ACD-DCA -A
12	$P_L[\alpha] \oplus P_R[15] \oplus C_L[15] \oplus C_R[\alpha, \beta]$ $= K_1[22] \oplus Q_3 \oplus Q_7 \oplus K_{11}[22] \oplus K_{12}[\gamma]$	1.42×2^{-32}	A-ACD-DCA -AB
13	$P_L[15] \oplus P_R[\alpha, \beta] \oplus C_L[15] \oplus C_R[\alpha, \beta]$ $= K_1[\gamma] \oplus K_2[22] \oplus Q_4 \oplus Q_8 \oplus K_{12}[22] \oplus K_{13}[\gamma]$	1.11×2^{-35}	BA-ACD-DC A-AB
14	$P_R[\delta] \oplus C_L[\alpha] \oplus C_R[15]$ $= Q_2 \oplus Q_6 \oplus Q_{10} \oplus K_{14}[22]$	1.42×2^{-40}	-DCA-ACD- DCA-A
15	$P_L[\delta] \oplus P_R[12, 16] \oplus C_L[\alpha] \oplus C_R[15]$ $= K_1[19, 23] \oplus Q_3 \oplus Q_7 \oplus Q_{11} \oplus K_{15}[22]$	1.42×2^{-42}	E-DCA-ACD -DCA-A
16	$P_L[\delta] \oplus P_R[12, 16] \oplus C_L[15] \oplus C_R[\alpha, \beta]$ $= K_1[19, 23] \oplus Q_3 \oplus Q_7 \oplus Q_{11} \oplus K_{15}[22] \oplus K_{16}[\gamma]$	1.11×2^{-45}	E-DCA-ACD -DCA-AB

α : $7, 18, 24, 29$
β : $27, 28, 30, 31$
γ : $42, 43, 45, 46$
δ : $7, 18, 24$
Q_i : $K_i[22] \oplus K_{i+1}[14] \oplus K_{i+2}[22]$
A: $X[15] \oplus F(X, K)[7, 18, 24, 29] = K[22]$
B: $X[27, 28, 30, 31] \oplus F(X, K)[15] = K[42, 43, 45, 46]$
C: $X[29] \oplus F(X, K)[15] = K[44]$
D: $X[15] \oplus F(X, K)[7, 18, 24] = K[22]$
E: $X[12, 16] \oplus F(X, K)[7, 18, 24] = K[19, 23]$

Table 4.5: The success probability and number of known plaintext-ciphertext pairs on a 14-round best linear expression.

N	1.42×2^{40}	1.42×2^{41}	1.42×2^{42}
$P(S)$	84.1%	92.1%	97.7%

- plaintext ciphertext bit (13 bit) $P_R[11]-P_R[16]$, $C_R[0]$, $C_R[27]-C_R[31]$, $P_L[7,18,24]\oplus C_L[15]\oplus C_R[7,18,24,29]$.
- subkey bit (12 bit) $K_1[18]-K_1[23]$, $K_{16}[42]-K_{16}[47]$.

Thus, from the 13 bit of information obtained from known plaintexts and corresponding ciphertexts we can obtain 12 bits of the subkeys and one bit from the right-hand side of the expression (4.8). In addition, the formula obtained by substituting C for P and K_{17-i} for K_i is also an expression formed with the same probability, since we can also get information on subkeys 13 bit from this formula; total information from this formula would be of subkeys 26 bit.

The description algorithm is illustrated in the following (Matsui [87]).

Data Counting Phase

Step 1 Prepare 2^{13} counters TA_{t_A} $(0 \le t_A < 2^{13})$ and initialize them by zeros, where t_A corresponds to each value on 13 effective text bits of equation (4.8).

Step 2 For each plaintext P and the corresponding ciphertext C, compute the value 't_A' of step 1, and count the TA_{t_A}.

Key Counting Phase

Step 1 Prepare 2^{12} counters KA_{k_A} $(0 \le k_A < 2^{12})$ and initialize them by zeros, where k_A corresponds to each value of 12 effective key bits of expression (4.8).

Step 2 For each k_A of Step 1, let KA_{t_A} be the sum of TA_{t_A} so that the left side of equation (4.8), which can be determined by t_A and k_A, is equal to zero.

Step 3 Rearrange KA_{k_A} in order of magnitude of $|KA_{k_A} - N/2|$ and rename them as $\overline{KA}_{l_A}$ $(0 \le l_A \le 2^{12})$, where N denotes the number of plaintexts. Then, for each l_A,

(a) If $|KA_{k_A} - N/2| \le 0$, guess that the right side of expression (4.8) is 0.

(b) If $|KA_{k_A} - N/2| > 0$, guess that the right side of expression (4.8) is 1.

Exhaustive Search Phase

Step 1 Let W_m $(m = 0, 1, \cdots)$ be a series of candidates for 26 key bits arranged in order of reliability.

Step 2 For each W_m, search for the remaining 30 key bits until the correct value is found.

Matsui [87] used the maximum likelihood method mentioned above for solving the expression (4.8). However, it has not been confirmed that this is the most successful method for solving the expression, and there might be a more successful one. Rijmen *et al.* [115] proposed a method which uses overall information contained in the expression (4.8). Although, as far as DES is concerned, this method did not improve success probability either.

Matsui [87] used the experimental result of an 8-round DES to derive the relation between the number of known plaintexts, time complexity, and success probability which could be obtained as a result of cryptanalysis using the above algorithm (see Fig. 4.1).

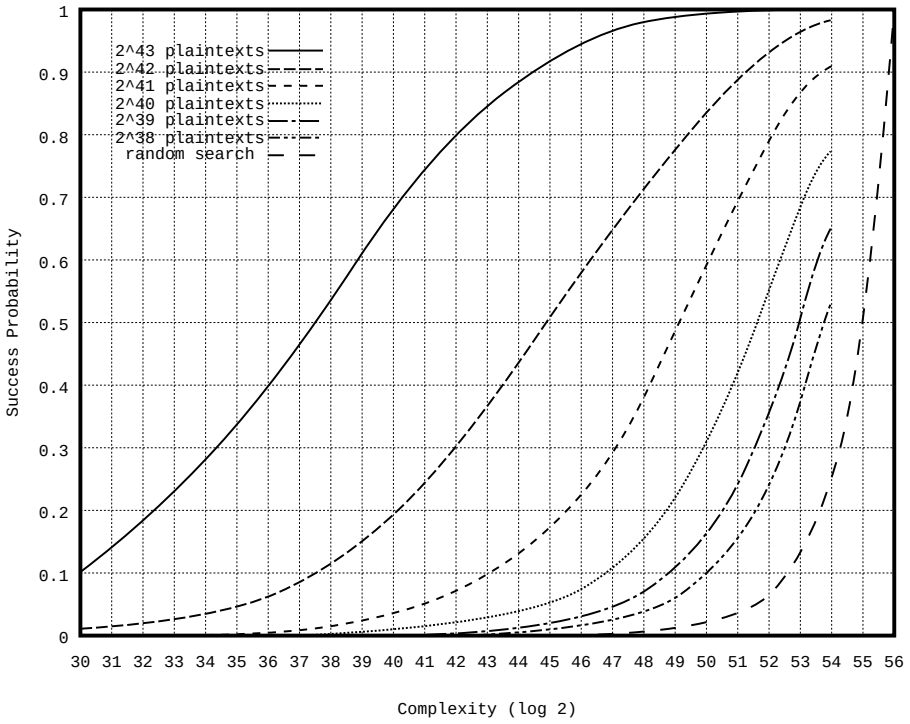

Figure 4.1: Result of linear cryptanalysis of DES.

It has been shown that, if you want to cryptanalyze with 80% success probability, you need to obtain 2^{43} plaintext-ciphertext pair to analyze by the time complexity of 2^{42} equivalent to the DES enciphering operation. When we compare differential and linear cryptanalyses, the difference in the two are: (i) texts required for analysis are chosen plaintexts for differential cryptanalysis and known plaintexts for linear cryptanalysis; and (ii) the approximate number of required plaintext-ciphertext pairs in order to deduce the correct key with high probability is 2^{47} in differential cryptanalysis compared with 2^{43} in linear cryptanalysis. Since the accuracy of these approximations is believed to

be satisfactory, we can assess with confidence that linear cryptanalysis is superior to differential cryptanalysis. In addition, as shown in the next section, many effective improvements in linear cryptanalysis have been proposed.

4.3.3 Improvements in Linear Cryptanalysis

Differential-Linear Cryptanalysis

Langford and Hellman [80] combined differential and linear cryptanalyses to create differential-linear cryptanalysis. They demonstrated that the attack is very effective for 8-round DES. In their cryptanalysis, they applied plaintext pairs with a specific XOR value that propagates, with probability 1, to a certain XOR in the intermediate state after 4 rounds confined to a subset of its bits. Then the 3-round best linear expression is constructed between the output of the 7th round and the input of the 5th round. As a result, it is possible to attack the last round by a 1R-attack of differential cryptanalysis. They effected this attack on 8-round DES and showed that 10 bits of keys could be obtained with a success rate of 80% using only 512 chosen plaintexts. However, approximation in differential-linear cryptanalysis holds when either both pairs of plaintext-ciphertext satisfy linear expression or when they do not. Consequently, the number of required plaintext-ciphertext pairs is in the order of the power of -2 of the bias of linear expression's success rate in the case of linear cryptanalysis, while it is -4 in the case of differential-linear cryptanalysis.

Attacks using Multiple Linear Expressions

In order to reduce the number of known plaintexts required, Kaliski and Robshaw [64] proposed a cryptanalyzing algorithm using multiple linear expressions. They showed that, against DES reduced to 6 rounds, a linear attack using two linear expressions has better success probability in deducing the correct right hand side of the linear expression under given numbers of plaintexts than a linear cryptanalysis using each linear expression. Moreover, trying to measure this attack's effect on full 16-round DES, Kaliski and Robshaw [64] examined the number of 14-round linear expressions involving a single S-box at each round with bias $|\epsilon_i| > 10^{-8} \simeq 1.34 \times 2^{-27}$. As a result, they found 10,006 such linear expressions, and that $\sum_{i=1}^{10,006} \epsilon_i^2 \simeq 1.69 \times 2^{-37}$. Therefore, even if we could use all these linear expressions at the same time, it would only result in a reduction by a factor of about 38 in the number of required known plaintexts compared with using one "best" linear expression ($\epsilon^2 \simeq 1.42 \times 2^{-42}$).

Terashima and Kaneko [132] proposed an attack using quadratic expressions composed of multiple linear expressions. They showed that, against 8-round DES, quadratic

expressions composed of two linear expressions could reduce the number of required plaintexts by about half compared with linear cryptanalysis using the best linear expression. This attack is quite effective when the linear expressions involved have a high correlation, although it is still open to discussion whether there is an effective quadratic expression (composed of linear expressions highly correlated to result in a large reduction in the number of required known plaintexts) against full 16-round DES.

Takeda and Kaneko [131] examined a binary symmetric channel model in which data sent are the masked value of the key and data received the masked value of plaintexts and ciphertexts. They tried to reduce the number of required known plaintexts by expanding channel capacity by using multiple linear expressions. As a result, they found that by using four of the best 128 linear expressions obtained, the number of required plaintexts is, theoretically, reduced by some 11.5% compared with linear cryptanalysis using the best linear expression.

Statistical Cryptanalysis

Recently, Vaudenay [136] showed that differential and linear cryptanalyses could be expanded and improved as a more generalized statistical attack. He further showed that, by using such generalized linear cryptanalysis against DES, the number of known plaintexts could be reduced from some 2^{43} to $2^{42.9}$ (an approximate 5% reduction).

4.3.4 Estimates of DES Strength with respect to Linear Cryptanalysis

We will estimate by setting success probability at 50%, and assume that the cryptanalyst uses an appropriate machine which conducts parallel processing based on several existing DES chips (in the next chapter, we term such a machine an *Eberle type machine*, and presents an estimation of a brute force method using it).

First, based on the results of the next chapter, we can estimate amount of calculation, analyzing time, and analyzing costs when the Eberle type machine is used (Table 4.6).

When we assume that known plaintext-ciphertext pairs can be obtained at the rate of 200Mbps, the time to obtain given known plaintext-ciphertext pairs and amount of calculation for cryptanalyzing DES by using the pairs with 50% success probability, are shown in Table 4.7.

These two tables illustrate that, in the case of linear cryptanalysis: collection of $2^{40+\alpha}\,(0 < \alpha < 1)$ known plaintexts-ciphertexts pairs in about a week would guarantee 50% success probability with time complexity of about 2^{50}, and at a cost of about \$0.06 million assuming a week's calculation; and collection of $2^{39+\beta}\,(0 < \beta < 1)$ known plaintext-ciphertext pairs in about three days would guarantee 50% success probability

Table 4.6: Computations, time, and cost to cryptanalyze DES by the Eberle type machine.

[million dollars]

time complexity	1 day	3 day	1 week	2 weeks
2^{52}	1.71	0.57	0.24	0.12
2^{51}	0.86	0.29	0.12	0.06
2^{50}	0.43	0.14	0.06	0.03
2^{49}	0.21	0.07	0.03	0.02

Table 4.7: Number of known-plaintexts, time to obtain plaintext, and time complexity of linear cryptanalysis – by Eberle type machine with 50% success probability.

known plaintexts	time complexity	time to obtain plaintexts
2^{41}	2^{49}	7.8 days
2^{40}	$2^{51.5}$	3.9 days
2^{39}	2^{53}	1.9 days

with time complexity of about 2^{52}, and a cost of about \$0.12 million assuming two weeks calculation.

For reference, Table 4.8 presents examples of cryptographic strength against linear cryptanalysis approximated by best linear characteristic probability for DES and other popular common key ciphers. The strength of DES against linear cryptanalysis is not much different from that of FEAL-16, in contrast with DES's superiority against differential cryptanalysis.

4.3.5 Conclusion of Linear Cryptanalysis

We have shown estimates that: if a cryptanalyst can obtain known plaintexts with the speed of 200Mbps, collecting known plaintexts in about a week and conducting linear cryptanalysis will enable the cryptanalyst to form an analysis of 50% success probability in a week at the cost of about \$0.06 million; and similarly, collecting known plaintexts in three days will enable to perform an analysis of 50% success probability in two weeks at the cost of about \$0.12 million. However, these estimates rely heavily on the assumption that the cryptanalyst can obtain known plaintexts at the speed of 200Mbps. Such an assumption is unrealistic under current communication networks among banks, and

Table 4.8: Strength of common key ciphers against linear cryptanalysis.

cipher	known plaintexts	time complexity	reference
DES	2^{43}	2^{42}	Matsui [87]
FEAL-8	2^{16}*	n.a.	Ohta *et al.* [108]
FEAL-16	2^{39}*	n.a.	Ohta *et al.* [108]
Multi2-8	$2^{16.5}$	$2^{48.5}$	Aoki and Kurokawa [6]
LOKI89	2^{83}**	n.a.	Tokita *et al.* [133]
LOKI91	2^{86}**	n.a.	Tokita *et al.* [133]

* Square value of the inverse of success probability of 15-round best linear expression, shown in Ohta *et al.* [108] used as an approximation.
** Square value of the inverse of success probability of 14-round best linear expression, shown in reference [133], used as an approximation.

among banks and companies; and, as long as networks use DES under circumstances which can reject such an assumption as unrealistic, one should not be concerned about linear cryptanalysis in its current form.

We should, however, note that the above estimates were based on approximation by best linear characteristic probability, and thus cannot deny the possibility of having been lenient compared with estimates based on best average linear probability. In addition, considering the fact that important improvements such as attacks using multiple linear expressions and statistical cryptanalysis have been proposed since the debut of linear cryptanalysis, and that it might be too early to say that all improvements have been exhausted, we should still keep our eyes on theoretical developments.

4.4 Conclusion

This chapter has attempted to assess the cryptographic strength of DES against the short-cut methods. Although DES has been subject to various attacks using short-cut methods such as the formal coding approach and meet-in-the-middle-attack, only Davies' attack, differential and linear cryptanalyses are superior (at least in some respects) to brute force methods. The differences in the three analyses being: (i) texts required for analysis are chosen-plaintexts for differential cryptanalysis and known-plaintexts for Davies' attack and linear cryptanalysis; and (ii) the approximate number of required plaintext-ciphertext pairs in order to deduce the correct key with high probability is

2^{50} for Davies' attack and 2^{47} for differential cryptanalysis compared with 2^{43} for linear cryptanalysis. Since the accuracy of these approximations is believed to be satisfactory, we can say with confidence that linear cryptanalysis is the best among the three cryptanalyses. Accordingly, based on the approximations we estimated success probability, time needed to obtain required plaintexts, breaking time, and analyzing cost. As a result, if one assumes a cryptanalyst obtains plaintext-ciphertext pairs at the speed of 200 Mbps, cryptanalysis with 50% success probability with breaking time of three to seven days would be estimated as possible at a cost of \$0.1 - \$0.6 million. However, linear cryptanalysis is not an immediate threat because the assumption for the attack is not realistic under current telecommunication network systems among banks and among banks and firms. This view does not, of course, lessen the importance of monitoring the theoretical developments of linear cryptanalysis: the assessment made above is no more than an approximation, and some improvement in the analysis might be proposed in the near future.

Chapter 5

Evaluation of DES using Brute Force Methods

This chapter examines the cryptographic strength of DES against the brute force methods. In Section 5.1, we first survey the previous researches regarding exhaustive key search against DES. Then, we examine the strength of DES including its prospect in the near future by doing trial computations for the time length and the expenses necessary to break DES using highly practical technology. We also discuss whether the efficacy of authentication through cryptosystem can be preserved when the frequency of key exchange is increased.

In Section 5.2, a recent work (Kusuda-Matsumoto [77]) on the optimization of time-memory trade-off cryptanalysis presented by Hellman [53] is introduced and the safety of DES is evaluated against an attack using this method.

5.1 Exhaustive Search

For an exhaustive search, the condition for an attack to be made possible is as simple as obtaining either a single ciphertext, or a pair of known plaintext and corresponding ciphertext. As for attacking by using a single ciphertext, a cryptanalyst tries all possible keys one after another until the ciphertext is deciphered to a valid plaintext. As for attacking by using a pair of known plaintext and corresponding ciphertext, the known plaintext is enciphered with a trial key and the output is compared with a given ciphertext for equality.

While in theory the correct key can always be found by repeated trials, in practice the attack is thwarted if the time complexity is too large. Especially, when attacking by using only a single ciphertext, the process becomes very difficult since an additional task of interpreting each deciphered text is needed in order to check whether it is in fact the correct plaintext. Therefore, in order to focus on highly realistic exhaustive search,

5.1.1 Studies on Exhaustive Search

In the mid-1970s, when NBS (National Bureau of Standard) proposed DES as a data encryption standard, Diffie and Hellman were concerned about possible weakness in that the key length was too short to withstand an exhaustive search which could be conducted by any organization (such as governmental bodies) which had substantial computer facilities. In response to NBS's counter-argument, Diffie-Hellman [40] reported the results of their detailed study in the magazine *Computer*, which led to intense debate. The views of the two sides are as follows: Diffie-Hellman claimed that "an appropriate machine, consisting of a million LSI chips ($10 per unit) which detect one key per $1\mu (= 10^{-6})$ second, could parallel process all keys in one day. The cost of such a machine would be about $20 million." NBS argued against that: "it is impossible to make such high-speed LSIs for just $10 per unit, since LSIs which can detect one key per 40μ second would cost $100. In addition, 1,000 LSIs would be needed for parallel processing and it would take almost 100 years for the entire search."

In 1976, the Institute of Computer Sciences and Technology sponsored a workshop to assess the feasibility of building a machine that could recover a 56-bit key from a given fragment of plaintext and corresponding ciphertext. The workshop reached the following conclusion.

A machine which finds, on the average, one key per day could probably not be built until 1990 and the probability factor of it being available even then is estimated at only between 10% and 20%. In addition, the cost of such a machine would be several tens of millions of dollars (Meyer and Matyas [91] p.139 ll.30-33).

Looking back on the debate between Diffie-Hellman and NBS, it seems that NBS's claim was closer to the truth. Also, the forecast by the workshop, which mostly supported NBS's claim, seems to have been on the right track because at CRYPTO '92, held two years later than the forecast year, two technological innovations were reported which enabled building a machine that could conduct exhaustive search of DES in a day (on average) at the cost of several millions of dollars.

Wayner [137] proposed a method to search DES key space with a content-addressable search engine. Wayner assumed a machine with 2^{11} special chips each of which equipped by vast amount of processors. He computed that 450 of these machines are needed to explore a half of the key space of DES in a day. He also assessed that the unit price of the chip will be $30 and that of the control hardware $10,000. According to this assumption, the cost of 450 machines will be approximately $32 million, unit cost of a machine being approximately $71,400.

Eberle [42] suggested searching the DES key space with a 1 Gbps gallium-arsenide (GaAs) DES chip developed by Digital Equipment Corporation (DEC). Its unit price was $300 when ordered in lots of a thousand units. Assuming that the manufacturing cost of key search machine is twice the cost of encryption chips, a $16 million machine would take one day to search through half of the key space.

The results of the two estimates are mere forecasts presented at the conference mentioned above, but in the rump session of CRYPTO '93, Wiener [140] showed how to build a special key search machine for $1 million (an additional $0.5 million for development costs[1]) that could break DES in an average of 3.5 hours. Such high cost-performance depends on the use of key search chips which realizes each round encryption by total pipeline processing. By this system, 16 encryptions take place simultaneously. In other words, although it takes 16 clock ticks to complete an encryption, one encryption is completed at the end of the pipeline on each clock tick. In addition, since there is very little need for input and output, it can be comfortably clocked at 50 MHz and would cost $10.5 to manufacture. [2] A key search machine is built out of the chips by building a hierarchy of controllers. Such machine's frame which contains 5,760 chips, would cost $0.1 million. A key search machine would consist of a number of frames. Machine cost and expected search time are shown in Table 5.1.

Table 5.1: Wiener's machine cost versus key search time.

search time [hour]		0.35	3.5	35
machine cost	A	0.5	0.5	0.5
[million dollar]	B	10.0	1.0	0.1

A:fixed cost, B:variable cost

Among the methods presented by Wayner, Eberle, and Wiener, Wiener's system is the best in terms of the cost-performance calculated by themselves. However, Wiener's machine has not yet been actually built, and therefore the accuracy of his trial computation has not been verified. Hence, in the next subsection, the strength of DES against exhaustive key search is assessed using the method presented by Eberle which just requires parallel processing of multiple chips which are already available.

[1] According to Wiener, it would take 30 man months for the development, $0.3 million for personnel expenses, and an additional $0.2 million for support in layout, chip fabrication, and board layout, bringing total development costs to about $0.5 million.

[2] As to the validity of this estimate, Wiener stated: *The precise assessment of speed and cost is made possible by the fact that the chip has been designed down to the gate level in a mature CMOS process.*

In addition, Blaze *et al.* [15] presented two types of new technology on exhaustive search against DES. [3] One type takes up Application-Specific Integrated Circuits (ASICs) and the other uses Field Programmable Gate Arrays (FPGAs). In the former type, the cost for finding a key in a day on average is approximately $21,000 (only chip costs included) according to their estimates. However, the development cost of the machine is stated to be enormous (no specific figure given). As for the latter, since FPGAs are widely available mounted on cards, can be installed in standard PCs. Hence, if many PCs are at hand, a key search machine can be materialized by only procuring the FPGAs. In this case, the cost of finding the key in a day on average is shown as $8.3 million, which is almost the same as our estimate based on the Ebelre type machine presented in the next subsection. [4]

Very recently, Boneh *et al.* [16] presented a method of exhaustive search against DES using a molecular computer. [5] According to their trial computation, it takes two months in average to find the key for DES from a pair of known plaintext and ciphertext using a molecular computer presented by them. However, their computation requires more scrutiny for its adequacy since it is based on an unrealistic assumption such as all the molecular biology experiments work perfectly without any errors. Moreover, Boneh *et al.* [16] lacks sufficient information for assessing its cost-performance since it doesn't refer to the method of building the molecular computer and how much it costs. Anyway, since the research for the materialization of molecular computers have just started, the developments in this field should be kept under careful observation.

5.1.2 Estimates of DES Strength by the Exhaustive Search

The strength of DES against exhaustive key search is assessed assuming the key search machine presented by Eberle. For simplicity, the following premises are introduced.

[3] It should be noted that the approach of Blaze *et al.* [15] is quite different from ours. While our approach is to assess the strength of DES with the fixed key length of 56 bits, Blaze *et al.* [15] focus on showing the approximate minimal key length required for the key of DES to repel various types of attackers under different budget constraints.

[4] It is also thought that software approach which is the way to run the cracking program on PCs or WSs. The approach had been infeasible hitherto because it requires greatly many PCs or WSs. But, the rapid spread of Internet has been making it feasible gradually. Very recently, in RSA Secret-Key Challenge, a group which was led by Verser, a programmer in U.S.A., used ten thousands of computers over the Internet and succeeded in finding a DES key tying given plaintext-ciphertext pairs by spending four months, which amounts about the one fourth of the 56-bit key space (RSA Data Security Inc. [117]).

[5] Molecular computer is a device which utilizes the fact that when DNA strands and comlementary strands come close to each other, they will merge and form a double helix structure. After Adleman [4] had solved the traveling salesman problem using this feature of DNA, Lipton [81] further developed this theory and showed an encoding technique which translates the pairs of DNA into serial numbers consisting of 0 and 1. Boneh *et al.* [16] showed the basic role of programming DES algorithm on molecular computers using this technique.

- Variable cost including the manufacturing cost of the system is proportional to the cost of producing encryption chips.
- The annual rate of decline in variable cost is constant.

It should be noted that there is some fixed cost such as development cost besides variable cost. Such fixed cost is not mentioned due to difficulties in assessment. However, since this system does not require the development of special encryption chip like Wiener's system, the development cost can be expected to be substantially lower than Wiener's as long as the magnitude of parallel processing is not extremely large.

Assuming that the cost for unit processing speed of the key search machine (cost in dollars per Gbps) is c_2 \$/Gbps at the point of τ, the processing rate of the key search machine for the cryptanalysis of DES under exhaustive search method, $V_{ES}(T, P)$ Gbps, and its cost C_{ES} dollars can be derived as a function of breaking time T hours and success probability P.

$$V_{ES}(T, P) = 2^{56} \times 64 \times P \div (3.6 \times 10^3 T) \div 2^{30} \simeq 1.14 \times 2^{20} \frac{P}{T}, \tag{5.1}$$

$$C_{ES}(T, P; \tau) = c_2(\tau)\, V_{ES}(T, P) \simeq 1.14 \times 2^{20}\, c_2(\tau)\, \frac{P}{T}. \tag{5.2}$$

Key Search Machine

Under the above assumption, the cost-performance of the key search machine is presented by $c_2(\tau)$. In order to derive the actual value of $c_2(\tau)$, we referred to the S machine assumed in Kusuda-Matsumoto [77]. S machine is based on the method introduced by Eberle and therefore, we will call S machine "Eberle type machine" hereafter. The assumption employed in deriving the cost for S machine is the same as ours mentioned before. They estimated $c_2(\tau)$ as follows.

First, they took in account that Wiener's machine [140] costs 1.6 times as much as chips and assumed that the manufacturing cost is twice the cost of encryption chips.

In 1992, the encryption rate of the chip was 1 Gbps and its unit price \$300 when ordered in lots of a thousand (Eberle [42]). The operation of $f^{(s)}$ includes key generation. Considering the fact that three years have passed since Eberle presented his research, assumed that a chip with encryption rate (including key generation) of 1 Gbps and unit price of \$300 when ordered in lots of a hundred was available at January 1995. Assuming that the cost of Eberle type machine is twice the cost of chips, they decided $c_2(0) = 600$.

In order to estimate the future value of $c_2(0)$, Kusuda-Matsumoto [77] replaced $c_2(\tau)$ with cost per processing rate of the encryption chip, the cost of which accounts for half of the total cost of the key search machine. Since such data were difficult to obtain, they

took the middle value between the decrease rate (37.2% in [18], 37.0% in [50]) used by other researchers and one (33.9%) which they calculated from the available data. Details are as follows.

Garon and Outerbridge [50] forecast that the processing rate of the same priced (nominal basis) chip would increase eight times every five years. Since the rate of inflation was forecast at 5% annually, the real price per processing rate decreases 37.2% every year. Brickell *et al.* [18] forecast that the breaking cost by exhaustive search would decrease by half every 18 months. This shows that the price per processing rate will decrease 37.0% every year.

Kusuda-Matsumoto [77] estimated the processing rate price by comparing the latest real processing rate price of the DES with one in the late 1970s when it was first launched. For data in the late 1970s, they set the price per Gbps at \$64,000 because, in the argument between Diffie-Hellman and NBS in 1977, NBS commented that it cost \$100 to make chips (1.6 Mbps) which could check one key per 40 μsec. For the latest data, Kusuda-Matsumoto [77] adopted \$300/Gbps, which is the processing rate price of a chip developed by DEC in 1992. Considering the CPI increase (60.6 $\rightarrow$ 140.3) from 1977 to 1992, they estimated that the real processing rate price is decreasing 33.9% every year.

Hence, they formulated $c_2(\tau)$ as follows:

$$c_2(\tau) \;=\; 0.650^{\tau} c_2(0). \tag{5.3}$$

In this case, the manufacturing cost C_{ES} (variable cost only) of the Eberle type machine is formulated as follows:

$$C_{ES}(T, P, \tau) \;\;=\;\; 7.17 \times 10^2 \times 0.650^{\tau} \frac{P}{T} \quad [\$\mathrm{M}]. \tag{5.4}$$

Basic Attack

The relationship among the search time, the required processing rate of the key search machine, and the manufacturing cost (variable cost only) is represented in Table 5.2 derived from equations (5.1) and (5.4). Table 5.2 indicates that currently (at January 1996), successful attack in a short period such as one day is difficult in the sense that it would cost as much as \$10 million. However, if the search time is extended to above one month, variable cost will be below \$1 million which indicates that it can no longer be stated as totally unrealistic. If we look into the future, in 2001 which is five years from now, a successful attack in one day also may not be stated as unrealistic since variable cost may decline to approximately \$1 million.

Let us examine the above results from the viewpoint of using DES in inter-bank ciphered data communication system. First, in inter-bank communication system, au-

Table 5.2: Results of exhaustive search applied to DES using the Eberle-type machine.

	year	1 hour	4 hours	1 day	1 week	1 month	3 months
V'_{ES}[Tbps]*	—	583.7	145.9	24.3	3.47	0.81	0.27
C_{ES} [$M]**	1996	233.00	58.25	9.71	1.39	0.32	0.11
	1997	151.45	37.86	6.31	0.90	0.21	0.07
	1998	98.44	24.61	4.10	0.59	0.14	0.05
	1999	64.00	16.00	2.67	0.38	0.09	0.03
	2000	41.59	10.40	1.73	0.25	0.06	0.02
	2001	27.03	6.76	1.13	0.16	0.04	0.01

* $V'_{ES} = 2^{-10} V_{ES}$.
** Fixed cost such as development cost is not included.

thentication is far more important than confidentiality as a requisite for cryptosystem since tampering and masquerading are larger threats than eavesdropping in general. As shown in Chapter 2, if the time needed for cryptanalysis exceeds the interval of key exchange, the efficacy of authentication will still be preserved while that of confidentiality may be lost. Hence, from the viewpoint that only the authentication facility must be maintained, there maybe an idea that the threat of exhaustive key search would be avoided for some time if the frequency of key exchange was increased to about once in a day.

This idea is accurate in an environment where cryptanalysts can obtain known plaintexts only rarely. On the contrary, in an environment where cryptanalysts can obtain known plaintexts relatively easily, merely increasing the frequency of key exchange is not enough for preserving effective authentication, since consecutive attack on newly exchanged keys (it will be called "attack in waves" hereafter) is possible.

Attack in Waves

In order to verify whether efficacy of authentication will be preserved by users' increasing frequency of key exchange in an environment where cryptanalysts can relatively easily obtain a known plaintext and hence the attack in waves is possible, we will make a trial computation assuming that the cryptanalyst has a key search machine which can explore a half of the key space of DES in T_0 days. In order to simplify the calculation, the following two premises are introduced.

- The line under attack is used 24 hours everyday.
- Cryptanalysts can obtain a known plaintext right after each key exchange.

The second premise is introduced assuming cases such as a fixed plaintext is transferred right after each key exchange or chosen plaintext attack is possible.

Under the above premises, the probability $P_{EX}(n, T_0)$ for a cryptanalyst to successfully find a true key before key exchange at least once in an attack in waves lasting T_0 days against a line where the interval of key exchange is n days will be given by the following formula.

$$\begin{aligned} P_{EX}(n, T_0) &= 1 - \left(1 - \frac{0.5n}{T_0}\right)^{\frac{T_0}{n}} \\ &> \lim_{n \to +0} 1 - \left(1 - \frac{0.5n}{T_0}\right)^{\frac{T_0}{n}} \\ &= 1 - e^{-0.5} \\ &\simeq 0.393 \end{aligned} \tag{5.5}$$

The above formula shows that the probability of successful attack before key exchange cannot be decreased under 39.3% no matter how frequently the key exchange is made. We have already shown that it can no longer be stated that searching half of the key space in about a month is totally impossible. Hence, in a ciphered data communication system where cryptanalysts can obtain a known plaintext relatively easily, assuming that they can continue their attack in waves for about a month, an alert for exhaustive key search by an attack in waves is needed even if the frequency of key exchange is increased to once in each day.

5.2 Time-Memory Trade-off Cryptanalysis

The problem with exhaustive search is that the amount of computation needed after the interception of ciphertext is enormous. Therefore, as we showed in the last section, in an exhaustive search, enormous expenses are needed in order to successfully search the key in a short time after obtaining a ciphertext. There is a method called *table lookup attack* in which a cryptanalyst effects computation before he intercepts a ciphertext. In the table lookup attack, the cryptanalyst enciphers one specific plaintext under each key in advance, and stores the results in a table. After intercepting the corresponding ciphertext, he looks it up in the table and immediately finds the correct key. Obviously, in this attack, time complexity of the intercepted ciphertext is very small, but space complexity becomes huge. As a result, the breaking cost by table lookup attack is much more than by exhaustive search.

In 1980, Hellman [53] improved the table lookup attack into *time-memory trade-off cryptanalysis*. This method dramatically reduces the table size, and enables the

cryptanalyst to search many keys by an operation equivalent to one encryption. Hellman [53] showed that both time complexity of intercepted ciphertexts required by exhaustive search and space complexity required by the table lookup attack could be dramatically reduced.

The major features of an exhaustive search, a table lookup attack, and a time-memory trade-off cryptanalysis can be summarized as in Table 5.3.

Table 5.3: The major differences among exhaustive search, table look-up attack, and time-memory trade-off cryptanalysis.

	exhaustive search	table lookup attack	time-memory trade-off cryptanalysis
premises	–one ciphertext –one known plaintext	–one predictable known plaintext	–one predictable known plaintext
precomputation	unnecessary	necessary	necessary
space complexity*	$O(1)$	$O(2^N)$	$O(2^{\frac{2}{3}N})$
time complexity**	$O(2^N)$	$O(1)$	$O(2^{\frac{2}{3}N})$

N: key length

* The unit is one block.

** The unit is the amount of computation required for one encryption.

As shown in Table 5.3, there is a more strict precondition for a cryptanalyst in a time-memory trade-off cryptanalysis compared with exhaustive search which is that a plaintext must be predictable a long time before the actual attack is carried out. The cases in which this condition is met are, i) a long time fixed plaintext is transferred either after each key exchange or at the beginning of a session, and, ii) the case in which a chosen plaintext attack is possible.

It should be noted that when making an attack against a single key, the total amount of computation required for time-memory trade-off cryptanalysis exceeds that of exhaustive search, because that for precomputation of the former is approximately the same as that for total computation of the latter. However, for the cases in which the same look-up table can be used in attacking different keys, precomputation is not needed for the second attack and after. As for the ciphered data communication systems in which a common predictable known plaintext can be intercepted from multiple communication lines, an attack against multiple keys, which will be explained later, is possible and in these cases, we will show that the amount of precomputation will be substantially reduced compared with exhaustive search.

The open question regarding time-memory trade-off cryptanalysis was the relation

among breaking cost, time, and success probability, but reference [53] did not clarify it. Recently, Kusuda and Matsumoto [77] clarified the relations with respect to time-memory trade-off cryptanalysis. They formulated the relation among breaking cost, time, and success probability and then optimized the formula.

In this section, we introduce the outline of time-memory trade-off cryptanalysis and some recent work on its optimization. Then we discuss how the strength of DES can be evaluated by the optimized time-memory trade-off cryptanalysis.

5.2.1 Outline of Time-Memory Trade-off Cryptanalysis

Time-memory trade-off cryptanalysis presented by Hellman [53] consists of two phases, the precomputation phase and the key search phase. The following descriptions of both are taken from Kusuda-Matsumoto [77] (see tables 5.4 and 5.5 for notations).

Table 5.4: The target.

name	symbol	definition
message block length	M	a positive integer
key length	N	a positive integer
enciphering function	Enc	a function: $Z_2^N \times Z_2^M \rightarrow Z_2^M$
correct key	K_0	$K_0 \in Z_2^N$
chosen plaintext	X_0	$X_0 \in Z_2^M$
intercepted ciphertext	Y_0	$Y_0 = Enc(K_0, X_0)$

Precomputation Phase

Time-memory trade-off cryptanalysis employs one chosen plaintext and plural *look-up tables*. The chosen plaintext mentioned here does not have to be special but should be predictable. In using l look-up tables, l *modified enciphering functions* $f : Z_2^N \rightarrow Z_2^N$ such that $f^{(s)}(K) \stackrel{\text{def}}{=} R^{(s)}(Enc(K, X_0))$ where $R^{(s)}$ are *adjustment functions*: $Z_2^M \rightarrow Z_2^N$ which are linear functions of rank N (see Fig. 5.1).

The cryptanalyst chooses $m \times l$ different keys, $K_{10}^{(s)}, K_{20}^{(s)}, \cdots, K_{m0}^{(s)}$, to make up *initial vectors* $I^{(s)}$ from the key space and computes

$$K_{ij}^{(s)} = f^{(s)}(K_{i,j-1}^{(s)}) \qquad (i = 1, \cdots, m;\ j = 1, \cdots, t;\ s = 1, \cdots, l) \tag{5.6}$$

as depicted in Fig. 5.2. As a result, *searched matrices* $A^{(s)}$ and *searching vectors* $B^{(s)}$ are generated. The cryptanalyst discards keys in $A^{(s)}$ columns from the 1st to $t-1$st, which are within the solid frame in Fig. 5.2, and makes *look-up tables* $D^{(s)}$ and stores.

Table 5.5: Notation for time-memory trade-off cryptanalysis.

name	symbol	definition
number of tables	l	a positive integer
number of rows of search matrix	m	a positive integer
number of columns of search matrix	t	a positive integer
adjustment function	$R^{(s)}$	a linear function of rank N: $Z_2^M \rightarrow Z_2^N$ $(s = 1, \cdots, l)$
modified enciphering function	$f^{(s)}$	a function: $Z_2^N \rightarrow Z_2^N$ defined by $f^{(s)}(K) = R^{(s)}(Enc\,(K, X_0))$ $(s = 1, \cdots, l)$
initial vector	$I^{(s)}$	$I^{(s)} = \begin{bmatrix} K_{10}^{(s)} \\ \vdots \\ K_{m0}^{(s)} \end{bmatrix}$ where $K_{i0}^{(s)} \in Z_2^N$ and $(i,s) \neq (i',s') \Rightarrow K_{i0}^{(s)} \neq K_{i'0}^{(s')}$ $(i = 1, \cdots, m;\ s = 1, \cdots, l)$
search matrix	$A^{(s)}$	$A^{(s)} = \begin{bmatrix} K_{10}^{(s)} & \cdots & K_{1,t-1}^{(s)} \\ \vdots & \ddots & \vdots \\ K_{m0}^{(s)} & \cdots & K_{m,t-1}^{(s)} \end{bmatrix}$ where $K_{ij}^{(s)} = f^{(s)}(K_{i,j-1}^{(s)})$ $(i = 1, \cdots, m;\ j = 1, \cdots, t-1;\ s = 1, \cdots, l)$
searching vector	$B^{(s)}$	$B^{(s)} = \begin{bmatrix} K_{1t}^{(s)} \\ \vdots \\ K_{mt}^{(s)} \end{bmatrix}$ where $K_{it}^{(s)} = f^{(s)}(K_{i,t-1}^{(s)})$ $(i = 1, \cdots, m;\ s = 1, \cdots, l)$
look-up table	$D^{(s)}$	$D^{(s)} = \begin{bmatrix} K_{\sigma(1)0}^{(s)} & K_{\sigma(1)t}^{(s)} \\ \vdots & \vdots \\ K_{\sigma(m)0}^{(s)} & K_{\sigma(m)t}^{(s)} \end{bmatrix}$ where σ is a substitution on $\{1, \cdots, m\}$ such that $\sigma(p) \preceq \sigma(p') \Rightarrow K_{\sigma(p)t}^{(s)} \preceq K_{\sigma(p')t}^{(s)}$ where $\preceq$ is a total order defined over Z_2^N $(s = 1, \cdots, l)$
modified key	$K_r^{(s)}$	$K_1^{(s)} = R^{(s)}(Y_0) \in Z_2^N$ $K_r^{(s)} = f^{(s)}(K_{r-1}^{(s)}) \in Z_2^N$ $(r = 2, \cdots, t;\ s = 1, \cdots, l)$

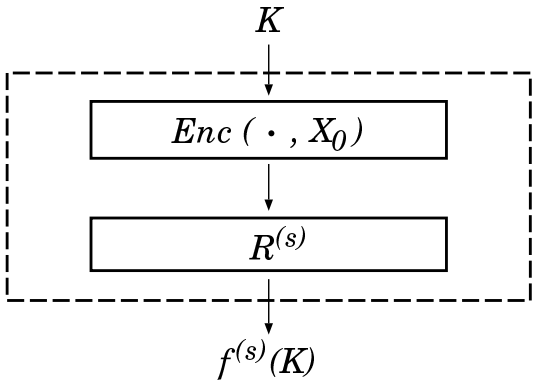

Figure 5.1: Constitution of modified enciphering functions $f^{(s)}$.

$$\begin{array}{ccccccccc}
K_{10}^{(s)} & \overset{f^{(s)}}{\rightarrow} & K_{11}^{(s)} & \overset{f^{(s)}}{\rightarrow} & \cdots & \overset{f^{(s)}}{\rightarrow} & K_{1,t-1}^{(s)} & \overset{f^{(s)}}{\rightarrow} & K_{1t}^{(s)} \\
K_{20}^{(s)} & \rightarrow & K_{21}^{(s)} & \rightarrow & \cdots & \rightarrow & K_{2,t-1}^{(s)} & \rightarrow & K_{2t}^{(s)} \\
\vdots & & \vdots & & \ddots & & \vdots & & \vdots \\
K_{m0}^{(s)} & \rightarrow & K_{m1}^{(s)} & \rightarrow & \cdots & \rightarrow & K_{m,t-1}^{(s)} & \rightarrow & K_{mt}^{(s)}
\end{array}$$

Figure 5.2: The procedure for making look-up tables with modified enciphering function $f^{(s)}$.

Key Search Phase

After the cryptanalyst intercepts $Y_0 = Enc(K_0, X_0)$, he searches for a *correct key* K_0 using look-up table $D^{(s)}$, via the following procedure.

Step 1 The cryptanalyst applies an adjustment function $R^{(s)}$ to obtain *modified key* $K_1^{(s)} = R^{(s)}(Y_0) = f^{(s)}(K_0)$, and checks if $K_1^{(s)}$ exists in the 2 nd column of look-up table $D^{(s)}$. If it does not, the correct key K_0 does not exist in the $t-1$ st column of the searched matrix $A^{(s)}$. If $K_1^{(s)} = K_{it}^{(s)}$, *i.e.* $f^{(s)}(K_0) = f^{(s)}(K_{i,t-1}^{(s)})$ for a certain $i \in \{1, \cdots, m\}$, this means either of the following:

(a) $K_0 = K_{i,t-1}^{(s)}$,

(b) $K_{it}^{(s)}$ has more than one inverse image and $K_0 \neq K_{i,t-1}^{(s)}$.

The latter event is called a *false alarm.* If $K_1^{(s)} = K_{it}^{(s)}$, the cryptanalyst computes $K_{i,t-1}^{(s)}$ from $K_{i0}^{(s)}$ which exists in the 1 st column of $D^{(s)}$ and checks if it is the correct key.

Step 2 If the correct key does not exist in the $t-1$ st column of searched matrix $A^{(s)}$, the cryptanalyst computes $K_2^{(s)} = f^{(s)}(K_1^{(s)})$ and checks if $K_2^{(s)}$ exists in searching vector $B^{(s)}$. If $K_2^{(s)} = K_{i't}^{(s)}$ for a certain $i' \in \{1, \cdots, m\}$, he checks if $K_{i',t-2}^{(s)}$ is the correct key.

Step 3 If the correct key does not exist in the $t-2$ nd column of $A^{(s)}$, then the cryptanalyst computes $K_r^{(s)} = f^{(s)}(K_{r-1}^{(s)})\,(r = 3, \cdots, t)$ in order and searches remaining columns of $A^{(s)}$ in a similar manner.

This is the outline of time-memory trade-off cryptanalysis. Once a look-up table is made, it can be used, even though the correct key has changed, as long as ciphertext corresponding to chosen plaintext X_0 can be intercepted.

5.2.2 Optimization of Cryptanalysis

This section introduces an optimization of time-memory trade-off cryptanalysis, developed by Kusuda-Matsumoto [77].

As obvious from the previous section, when a cryptanalyst makes $l\,(\simeq \frac{2^N}{mt})$ look-up tables, excluding the influence of false alarms, $f^{(s)}$ operations required in the precomputation phase are about $1/m$ those of exhaustive search, and required storage capacity about $1/t$ that for table lookup. From this viewpoint, larger matrices are preferable. However, as they become larger, more false alarms occur and $f^{(s)}$ operations increase with more keys randomly generated overlapping one another, and the number of non-overlapping keys to all keys in searched matrices decreases. Therefore, the relation among l, m, t, breaking time, and success probability becomes complicated.

Let P_L and O_F be the lower bound of probability of success and increase in $f^{(s)}$ operations with the occurrence of false alarms, respectively. It was assumed that modified enciphering functions have the following property to make analytic treatment easy.

Assumption 5.1 *Images under $f^{(s)}$ are generated uniformly and randomly.*

Two expressions involving P_L and O_F have been deduced from this assumption (for a proof, see Kusuda-Matsumoto [77]),

$$P_L = 1 - \exp\left(-\frac{l\,m\,t}{2^N} \cdot \frac{1}{u} \int_0^u \frac{1 - e^{-x}}{x}\, dx\right), \tag{5.7}$$

$$0 \leq E(O_F) \leq \frac{u}{6} \cdot l\, t, \tag{5.8}$$

where

$$u = \frac{m\, t^2}{2^N}. \tag{5.9}$$

Kano *et al.* [66] conducted an experiment with DES reduced to a 20-bit key in order to verify the validity of the above two expressions and showed that both are valid. Also, experiment results showed that the lower bound of expression (5.7) and upper bound (5.8) are both good approximations to true ones. It is highly possible, that the validity of the two expressions can be verified for DES. [6]

Cryptanalyzing Machine

Kusuda-Matsumoto [77] proposed the following cryptanalyzing machine.

A cryptanalyzing machine consists of plural units. As depicted in Fig. 5.3, a cryptanalyzing unit comprises of three parts:

- *Table storage* containing HDDs (hard disk drives) and array controllers
- *Key search machine* containing encryption chips, microcontroller, and a RAM
- *Controller*

Under the controller, look-up tables are loaded from HDDs in the table storage to the RAM in the key search machine, then each chip searches look-up tables in parallel. In this search, if $K_{r-1}^{(s)}$ exist in $B^{(s)}$ are checked while modified keys $K_r^{(s)}$ $(r = 1, \cdots, m;\ s = 1, \cdots, l)$ are generated from $K_{r-1}^{(s)}$ by $f^{(s)}$. Hence, the processing rate of the machine can be regarded as that of the operation of modified enciphering function $f^{(s)}$.

Optimal Cost Function

To simplify the estimates, Kusuda-Matsumoto [77] assumed that the costs of table storage and key search machine would satisfy the following properties.

Assumption 5.2 *Table storage cost and key search machine cost should be in proportion to the storage capacity and processing rate of $f^{(s)}$, respectively.*

[6] It seems that the only case where it would not hold is where the average cycle of modified enciphering functions is not long enough compared with columns of searched matrices; in this case, coverage of searched matrices will decline because the searched matrices meet a cycle within the same row with high probability. As regards this point, Hellman and Reyneri [55] reported an experimental result where the average cycle of the modified enciphering functions of DES is about 2^{28}. This number is considered to be long enough compared with the number of columns of assumed searched matrices ($2^{22} \leq t \leq 2^{24}$) if optimized time-memory trade-off cryptanalysis is to be used.

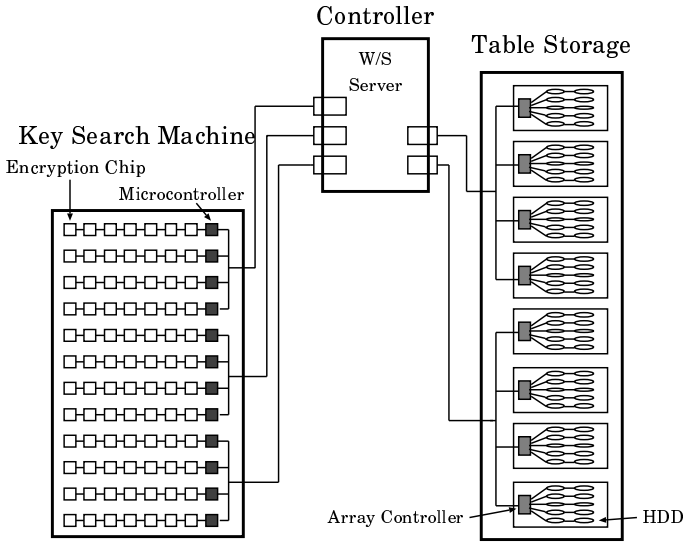

Figure 5.3: A unit of cryptanalyzing machine.

Then, neglecting controller cost, the cost of a cryptanalyzing machine, $C(t,u)$ [dollars] is represented as:

$$C(t,u) = c_1 U(t,u) + c_2 V(t,u) \tag{5.10}$$

where c_1 dollars/Gbit, $U(t,u)$ Gbit, c_2 dollars/Gbps, and $V(t,u)$ Gbps, are the table storage cost in dollars per Gbit, the *required storage capacity* of the table storage, the key search machine cost in dollars per Gbps, and the *required processing rate* of the key search machine, respectively.

Based on these assumptions, they derived the following theorem which showed the cost of analysis (*general cost function*) as a function of breaking time and success probability (for a proof, see Kusuda-Matsumoto [77]).

Theorem 5.1 (General Cost Function) *If the assumptions 5.1 and 5.2 are satisfied, the general cost function (excluding controller cost) $C(t,u)$ [dollars] is bounded by using breaking time T_s [seconds] and lower bound of success probability P_L as follows:*

$$C_L(t,u) \quad \leq \quad C(t,u) \quad \leq \quad C_U(t,u), \tag{5.11}$$

where

$$C_L(t,u) = 2^{-30}\Big(\ln(1-P_L)^{-1}\Big)\left\{ 2^{N+1} c_1 N t^{-1} \frac{1}{g(u)} + c_2 M T_s^{-1} t^2 \frac{1}{u \cdot g(u)} \right\}, \tag{5.12}$$

$$C_U(t,u) = 2^{-30}\Big(\ln(1-P_L)^{-1}\Big)\left\{ 2^{N+1} c_1 N t^{-1} \frac{1}{g(u)} + c_2 M T_s^{-1} t^2 \frac{u+6}{6u \cdot g(u)} \right\}. \tag{5.13}$$

A general cost function, minimized under given breaking time and success probability, is called *optimal cost function*, and they proved the following theorem (for a proof, see Kusuda-Matsumoto [77]).

Theorem 5.2 (Optimal Cost Function) *When breaking time T [hours] and lower bound of success probability P_L are given, optimal cost function $C^*(T, P_L)$ [dollars] is bounded by*

$$C_L^*(T, P_L) \ \leq \ C^*(T, P_L) \ \leq \ C_U^*(T, P_L), \tag{5.14}$$

where

$$C_L^*(T, P_L) \ = \ 0.94 \times 2^{\frac{2}{3}N-32} c_1^{\frac{2}{3}} c_2^{\frac{1}{3}} N^{\frac{2}{3}} M^{\frac{1}{3}} \frac{\ln(1-P_L)^{-1}}{T^{\frac{1}{3}}}, \tag{5.15}$$

$$C_U^*(T, P_L) \ = \ 1.02 \times 2^{\frac{2}{3}N-32} c_1^{\frac{2}{3}} \, c_2^{\frac{1}{3}} \, N^{\frac{2}{3}} \, M^{\frac{1}{3}} \, \frac{ln(1-P_L)^{-1}}{T^{\frac{1}{3}}}. \tag{5.16}$$

The difference between C_U^* and C_L^* is only 8%. To simplify the estimates, they regarded C_U^* as the optimal cost function. [7] Then, required storage capacity, $U_{GB}^*(T, P_L)$ Gbyte, and required processing rate, $V^*(T, P_L)$ Gbps are shown by

$$\begin{aligned} U_{GB}^*(T, P_L) \ &\simeq \ 2^{N-29} N \Big(\ln(1-P_L)^{-1}\Big) t^{*-1} \frac{1}{g(u^*)} \div 8 \\ &= \ 1.36 \times 2^{\frac{2}{3}N-36} \left(\frac{c_2}{c_1}\right)^{\frac{1}{3}} N^{\frac{2}{3}} M^{\frac{1}{3}} \frac{\ln(1-P_L)^{-1}}{T^{\frac{1}{3}}}, \end{aligned} \tag{5.17}$$

$$\begin{aligned} V^*(T, P_L) \ &\simeq \ 2^{-30} M \Big(\ln(1-P_L)^{-1}\Big) T_s^{*-1} t^{*2} \frac{u^*+6}{6u^* \cdot g(u^*)} \\ &= \ 1.36 \times 2^{\frac{2}{3}N-34} \left(\frac{c_1}{c_2}\right)^{\frac{2}{3}} N^{\frac{2}{3}} M^{\frac{1}{3}} \frac{\ln(1-P_L)^{-1}}{T^{\frac{1}{3}}}. \end{aligned} \tag{5.18}$$

The required storage capacity presented in blocks $U_{block}^*(T, P_L)$, and the expected amount of computation during the key search phase presented by the number of calculation of the modified enciphering functions (it will be called the "expected amount of ex post computation" hereafter) $E(O^*(T, P_L))$ are shown as follows:

$$\begin{aligned} U_{block}^*(T, P_L) \ &= \ 2l^*m^* \\ &\simeq \ 1.36 \times 2^{\frac{2}{3}N-3} \left(\frac{c_2 M}{c_1 NT}\right)^{\frac{1}{3}} \ln(1-P_L)^{-1}, \end{aligned} \tag{5.19}$$

[7] The results of the tests conducted by Kano *et al.* [66] mentioned above suggested that expression (5.7) on success probability and the upper bound of expression (5.8) on the influence of false alarms are both good approximations to true ones. If the upper bound of optimal cost function C_U^* corresponds to that of expression (5.8), it is thought to be valid to regard C_U^* as the optimal cost function.

$$
\begin{aligned}
E(O^*(T, P_L)) &= l^* t^* \left(1 + \frac{1}{6} u^*\right) \\
&\simeq 1.59 \times 2^{\frac{2}{3}N+8} \left(\frac{c_1 NT}{c_2 M}\right)^{\frac{2}{3}} \ln(1 - P_L)^{-1}. \qquad (5.20)
\end{aligned}
$$

The time complexity of precomputation is equivalent to that of exhaustive search, but cryptanalysts can share it if they attempt to break the same cipher, and save cost and time. In the case which a single cryptanalyzing machine is used for the precomputation, the required time length $T^*_{pre}(T)$ months, and the required amount of computation $O^*_{pre}(P_L)$ are shown as follows:

$$
\begin{aligned}
T^*_{pre}(T) &= M l^* m^* t^* \div (2^{30} V^*) \div (3600 \times 24 \times 30) \\
&\simeq 1.61 \times 2^{\frac{1}{3}N-18} \left(\frac{c_2 M}{c_1 N}\right)^{\frac{2}{3}} T^{\frac{1}{3}}, \qquad (5.21) \\
O^*_{pre}(P_L) &= l^* m^* t^* \\
&\simeq 1.36 \times 2^N \ln(1 - P_L)^{-1}. \qquad (5.22)
\end{aligned}
$$

5.2.3 Estimates of DES Strength by optimized time-memory trade-off cryptanalysis

In this section, a trial computation is performed in order to evaluate the strength of DES using optimized time-memory trade-off cryptanalysis under the assumption that the Eberle type machine described in Section 5.1.2 is used as a key search machine.

Table Storage

From the assumption 5.2, the cost-performance of the Table Storage will be described as $c_1(\tau)$. To decide the value of $c_1(\tau)$, Kusuda-Matsumoto [77] showed the structure of table storage to sales traders, as depicted in Fig. 5.3, in which 100 HDDs (9 Gbyte) are connected to ten array controllers, and requested the trader to give an estimate for storage. The trader estimated ¥58 million (HDDs: ¥45 million, array controllers, ¥9 million, cabinets, setting, and adjusting, ¥4 million). With an exchange rate of ¥100/\$, table storage cost is \$81.9 per Gbit. Hence, they decided $c_1(0) = 82$.

First, Kusuda-Matsumoto [77] used DISK/TREND REPORT (1992-1994 ed.) which contain the prices of storage capacity per storage capacity of extended HDDs made in USA (dollars/Mbyte, noncaptive worldwide shipment basis). They selected the lowest prices on an annual basis, z_τ (1990 to 1996; 1994 to 1996 data are projected). Then, by deflating their data with the US consumer price index, CPI_τ (data for 1995 and 1996

calculated from the projected rate of inflation by CBO), they calculated the real price per storage capacity, y_τ (1990 = 100). See Table 5.6.

Table 5.6: Change in cost-performance of HDD.

Year	90	91	92	93	94	95	96
z_τ	2.44	1.48	1.12	.700	.434	.247	.148
CPI_τ	130.7	136.2	140.3	144.5	148.3	152.9	158.1
y_τ	100.0	58.21	42.76	25.95	15.68	8.650	5.014

By regression analysis on the model

$$\ln y_\tau \;=\; \alpha + \beta\tau + \mu_\tau,$$

we have

$$\bar{R}^2 = 0.9933, \quad \hat{\beta} = -0.4927, \quad t_{\hat{\beta}} = -22.37.$$

Since the adjusted coefficient of determination, $\bar{R}^2$, is nearly 1.0, and the absolute value of the t-value, $t_{\hat{\beta}}$, is sufficiently large, this model is considered to be valid. The value $\exp(\hat{\beta}) = 0.6110$ shows that y_τ decreases 38.9% every year. Therefore, they set up $c_1(\tau)$ as follows:

$$c_1(\tau) \;=\; 82 \times 0.611^\tau. \tag{5.23}$$

Basic Attack

The following equations are obtained by substituting $M = 64$, $N = 56$, (5.23), (5.3), and $c_2(0) = 600$ into (5.16), (5.17), (5.18), (5.19), (5.20), (5.21), and (5.22):

$$C^*(T, P_L, \tau) \;=\; 3.82 \times 10^{-1} \times 0.624^\tau \frac{\ln(1-P_L)^{-1}}{T^{\frac{1}{3}}} \quad [\$\mathrm{M}], \tag{5.24}$$

$$U^*_{GB}(T, P_L, \tau) \;\simeq\; 3.89 \times 10^{2} \times 1.021^\tau \frac{\ln(1-P_L)^{-1}}{T^{\frac{1}{3}}} \quad [\mathrm{Gbit}], \tag{5.25}$$

$$V^*(T, P_L, \tau) \;\simeq\; 2.12 \times 10^{2} \times 0.960^\tau \frac{\ln(1-P_L)^{-1}}{T^{\frac{1}{3}}} \quad [\mathrm{Gbps}], \tag{5.26}$$

$$U^*_{block}(T, P_L) \;\simeq\; 1.78 \times 2^{35} \frac{\ln(1-P_L)^{-1}}{T^{\frac{1}{3}}}, \tag{5.27}$$

$$E(O^*(T, P_L)) \;\simeq\; 1.94 \times 2^{43} \times 0.960^\tau T^{\frac{2}{3}} \ln(1-P_L)^{-1}, \tag{5.28}$$

$$T^*_{pre}(T, \tau) \;\simeq\; 1.06 \times 10^{1} \times 1.040^\tau T^{\frac{1}{3}} \quad [\mathrm{month}], \tag{5.29}$$

$$O^*_{pre}(T) \;\simeq\; 1.36 \times 2^{56} \ln(1-P_L)^{-1}. \tag{5.30}$$

The required amount of precomputation, the expected amount of ex post computation, required storage capacity and other related results in order to achieve the success probability of above 50% using the optimized time-memory trade-off cryptanalysis at the point of January 1996 are shown on Table 5.7 derived from equations (5.24)-(5.30).

Table 5.7: Results of optimized time-memory trade-off cryptanalysis applied to DES using the Eberle type machine.

T [hour]	0.125	0.5	2	8
U^*_{block}[block]	$2^{36.3}$	$2^{35.6}$	$2^{35.0}$	$2^{34.3}$
$E(O^*)$	$2^{41.4}$	$2^{42.7}$	$2^{44.0}$	$2^{45.4}$
O^*_{pre}	$2^{56.3}$	$2^{56.3}$	$2^{56.3}$	$2^{56.3}$
U^*_{GB} [GB]	551	347	219	138
V^* [Gbps]	377	237	150	94
T^*_{pre} [month]	5.5	8.7	13.8	22.0
C^* [$M]*	0.33	0.21	0.13	0.08

* The fixed cost, such as development cost is not included.

According to Table 5.7, the expected amount of ex post computation is between $2^{41.4}$ and $2^{45.4}$ which are approximately between 1/12,000 and 1/800 of the amount of computation required for exhaustive search (2^{55}). Also, the required storage capacity is between $2^{36.3}$ and $2^{34.3}$ blocks which are approximately between 1/850,000 and 1/3,400,000 of that required for table look attack (2^{56} blocks). On the other hand, the required amount of precomputation is $2^{56.3}$ which is approximately 2.5 times the amount of computation required for exhaustive search (2^{55}) which indicates that the feasibility of this precomputation is a key point for this method.

Next, examining the required storage capacity and the required processing rate of the cryptanalyzing machine, the former is 347 GB (equivalent to 39 HDDs of 9 GB each), and the latter is 237 Gbps (equivalent to 237 chips of 1 Gbps each) both being the size which can no longer be stated as totally unrealistic. The production cost for the cryptanalyzing machine is estimated to be $0.21 million. In this case, the time needed for precomputation using only this machine is 8.7 months.

These results show that although the time needed for precomputation is rather long, once the look-up table has been prepared, it can no longer be stated that successfully cryptanalyzing DES in a short time is totally unrealistic. Also, for the cryptanalysts who don't care for massive precomputations, there are some ways to decrease them, which will be discussed next.

Limited Key Space Attack

Limited key space attack is, for example, a method which prepares a small look-up table such as 0.1% of the key space, and then try to find the actual key inside the look-up table by consecutively shifting the target key. Since this attack is valid when a cryptanalyst tries to masquerade as any agent in the network, it should be noted as a possible attack in the interbank ciphered data communication system where authentication is important.

The advantage of the limited key space attack for a cryptanalyst is that not only the amount of ex post computation but also that of precomputation can be reduced substantially. The method of reducing the amount of computation when trying to find with certain probability at least one key before key exchange within a certain attacking time, is to increase the number of attacks as much as possible, in other words shorten the time for each attack as much as possible and execute them consecutively. If the interval of key exchange is short in the target line, and also a predictable plaintext is transferred right after each key exchange, cryptanalyst may achieve the goal without difficulty. Another way is to shift the target line in a short interval. This consecutive attack against multiple lines can be carried out easier in a ciphered data communication system where keys are changed universally at the beginning of each day's session and fixed common plaintext is sent right after the key exchange. This is because the attack can be achieved by, first, intercepting all the lines right after a key exchange getting ciphertexts corresponding to the fixed plaintext, and second, storing all of them in the table storage, and third, extracting and attacking the stored ciphertexts one after another.

We made a trial calculation of the amount precomputation and the magnitude of the optimal apparatus required for a limited key space attack of which the length of each attack is 3 minutes and has a success (finding at least one key) probability of over 50% under a given number of attacks (see Table 5.8). In the above trial computation, the size of the equipment is far smaller than the previous ones and therefore it can be expected that the cost per unit storage capacity and the cost per unit processing rate will be substantially larger compared to C_1 and C_2 which were set under the condition that the cryptanalyzing machine is very large. However, since the ratio of these two numbers can be expected to be rather invariable with respect to the size of the equipment, it is assumed to be the same as the Eberle type machine.

Table 5.8 shows that the amount of precomputation is in the range of $2^{49.6}$ to $2^{44.2}$ and the expected amount of ex post total computation is $2^{40.5}$ which are approximately 1/40 to 1/1,800 and 1/23,000 respectively of that for total computation in exhaustive search. The size of the cryptanalyzing machine has become considerably smaller reflecting the decrease of the required amount of computation.

Table 5.8: The result of time-memory trade-off cryptanalysis using limited key space attack of which the length of each attack is 3 minutes. The purpose of the attack is to find at least one key by the probability of more than 50%.

n^*	80	480	3,360
$P(1)^{**}$[%]	0.86	0.14	0.02
T^{***}_{sum}	4 hours	1 day	1 week
$O^*_{pre}(n)^{****}$	$2^{49.6}$	$2^{47.0}$	$2^{44.2}$
$E(O^*(n))^{*****}$	$2^{40.5}$	$2^{40.5}$	$2^{40.5}$
T^*_{pre} [month]	4.0	4.0	4.0
U^*_{block}[block]	$2^{30.4}$	$2^{27.8}$	$2^{25.0}$
$E(O^*)$	$2^{34.2}$	$2^{31.6}$	$2^{28.8}$
U^*_{GB} [GB]	9.38	1.56	0.22
V^* [Gbps]	6.38	1.06	0.15

* number of attacks.
** $P(1) = 100P_L$, where $1 - (1 - P_L)^n = 0.5$.
*** The total time length of the attacks when executed consecutively.
**** The amount of precomputation for limited key space attacks of n times.
***** The amount of ex post computation for limited key space attacks of n times.

For example, under the condition that the cryptanalyst can intercept 480 lines getting ciphertexts of a common expected plaintext, the required storage capacity of an optimal cryptanalyzing machine is 1.56 GB which can be met by 1 HDD of 2 Gbyte in the case which 480 attacks, each of 3 minutes length are made (The total length of the attack will be a day when carried out consecutively). Also, in this case, required processing rate will be 6.38 Gbps which can be met by parallel processing of 32 CMOS chips, the speed of each are 200 Mbps which are relatively easy to be acquired. The time length of precomputation using only this machine is 4.0 month, which is rather long compared to that of each attack, but couldn't be stated to be totally impossible to carry out.

5.3 Conclusion

In this chapter, we examined the cryptographic strength of DES against the brute force method. In addition to exhaustive search, we focused on an optimized method of, rather overlooked, time-memory trade-off cryptanalysis.

We have compared the various techniques in exhaustive search using the data in published articles of trial computation performed by the presenters themselves. The result was that the apparatus presented by Wiener, equipped with special chips which realizes each round encryption by total pipeline processing, is the best in terms of cost-performance. According to Wiener's trial computation, a half of the key space of DES will be searched with a machine whose cost is $1.5 million (including the development cost of $0.5 million). If this is in fact true, it can be stated that cryptanalyzing DES in a short time by exhaustive search is no longer unrealistic. However, such a machine has not yet been actually built, and therefore the accuracy of his trial computation has not yet been verified.

Very recently, a work on exhaustive search against DES using a molecular computer was presented by Boneh *et al.*. The work towards materialization of molecular computers should be observed carefully from now on. However, sufficient information for the assessment of cost-performance is not provided as regards the computer presented by Boneh *et al.*.

Therefore, in this article, we have chosen another approach. We assumed a simple parallel processing machine using GaAs chips presented by Eberle and performed a trial computation of the relation between searching time and cost (variable cost only) for exploring a half of the key space of DES. The result was that it was still very difficult to do the search in one day due to formidable cost. However, we obtained some figures implying that a search in a month can no longer be stated as totally unrealistic. Also, we got some figures suggesting that in the year 2001, which is five years from now, a search in one day can no longer be stated as totally unrealistic though that in a very short time such as an hour can be stated to be very difficult because of the enormous cost required.

In the ciphered data communication system among banks, the common view is that authentication is more important than confidentiality. Hence, there maybe an idea of increasing the frequency of key exchange to about once in an hour as a tentative measure to protect the efficacy of authentication from the attack of exhaustive search. This measure may be effective in a system where a cryptanalyst can obtain known plaintexts only rarely. However, in a system where a cryptanalyst can obtain known plaintexts easily, we showed that increasing the frequency of key exchange is not effective because of the possibility of getting an attack in waves. For a system in which a fixed plaintext is

transferred each time after a key exchange, or a system in which a chosen plaintext attack is possible, an alert for an attack in waves should be needed because a cryptanalyst can execute a known plaintext attack relatively easily.

For those systems in which fixed plintext unchanged for a long time is transferred either right after each key exchange or at the beginning of each session, or those systems in which a chosen plaintext attack is possible, an attack in time-memory trade-off cryptanalysis should be kept in mind. Time-memory trade-off cryptanalysis is a method which decreases the amount of computation after intercepting the ciphertext by storing a look-up table beforehand which was built by exhaustive search against predicted plaintext in an environment where a plaintext which will be transferred can be foreseen far before the actual interception. We have done a trial computation assuming an Eberle type machine as a key search machine, and the result was that as for building a look-up table in order to explore more than a half of the key space of DES, it takes 2.5 times long time compared with the total hours required for exhaustive search. But once the look-up table is prepared, we obtained some data implying that cryptanalysis in 10 minutes or less may no longer be stated as totally unrealistic. In addition, it was shown that in a ciphered data communication system where interception of a common fixed plaintext against multiple lines is relatively easy, a cryptanalyst can attack multiple lines one after another in the aim of masquerading oneself as an indefinite legitimate user. It was shown that in this case, the amount of precomputation will be substantially lesser, and the size of the machine be much smaller, both compared with exhaustive search.

Chapter 6

A Strength Evaluation of Triple DES

The previous chapter's results estimates suggest that the security of DES is in increasing danger from the brute force method, especially optimized time-memory trade-off cryptanalysis. Therefore, this chapter examines the strength of triple DES proposed by Tuchman [134] in 1979 (see Fig. 6.1) since it is thought to be the most likely alternative to DES in the financial field. Sometimes triple DES refers to enciphering with three independent keys. We call this method *three-key triple DES*.

Double DES, *i.e.* $DES(K^{(2)}, DES(K^{(1)}, P)) = C$, is not considered in this paper because there is an attack which breaks double DES on the order of 2^{56} memories and 2^{56} encryptions. This attack is a kind of table look-up attack: a cryptanalyst creates a look-up table from $DES(K^{(1)}, P)$ (encipher known plaintext P using single DES with each key of $K^{(1))}$) and compare them with $DES^{-1}(K^{(2)}, C)$ (decipher the corresponding ciphertext C using single DES with each key of $K^{(2)}$).

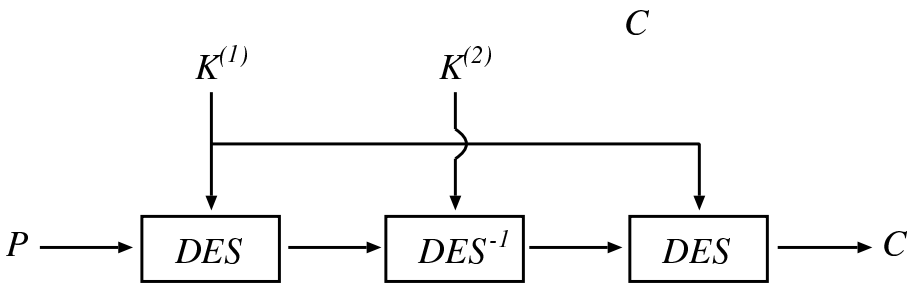

Figure 6.1: Structure of triple DES.

Triple DES, compared with DES (single DES hereafter), has double the key length (2^{56} times as much key space) and three times as many rounds, which are expected to

considerably improve cryptographic strength. In the case of a cascade cipher such as triple DES, the triple enciphering function will become a single enciphering function when a key indexed set of single enciphering functions is algebraically closed under compositional function. And, even if the set is not closed, strength would be affected if the size of the subgroup generated by the set is not sufficiently large. We have already confirmed in Section 3.3.1 that, with respect to DES, the set is not closed and that the size of subgroup generated is sufficiently large.

6.1 Effective Cryptanalytical Attacks against DES

Let us briefly review how the cryptographic strength of triple DES can be assessed against differential and linear cryptanalyses, exhaustive search, and optimized time-memory trade-off cryptanalysis; all of which are regarded as effective attacks against DES.

From the viewpoints of differential and linear cryptanalyses, triple DES can be regarded as a 48-round DES. For differential cryptanalysis, if we assume that the filtering method is used, the number of plaintext-ciphertext pairs required for analysis can be approximated to about 2^{174} by an inverse of characteristic probability which has been obtained by piling up 2-round best iterative characteristics from the 2nd to 46th round. For linear cryptanalysis, the number of required plaintext-ciphertext pairs can be approximated to about 2^{118} by an inverse of the cube of the probability of the 14-round best linear characteristic. In both cases, the numbers are much, much larger than the size of the message space (2^{64}), and deemed as theoretically unavailable.

Let us assess the strength of triple DES against exhaustive search and time-memory trade-off cryptanalysis, based on the estimate in sections 5.1 and 5.2. However, here we assumed a key search machine equipped with special chips which realizes each round's encryptions by total pipeline processing presented by Wiener [140] (we will call this "Wiener type machine" hereafter). We assumed here that in the near future, a trial computation resembling those done in Wiener [140] will become realistic due to technical innovation, although there is much controversy over its feasibility at this point. We estimated the cost per unit processing rate $c_2(0)$ of the machine at January 1995 ($\tau = 0$) as follows.

According to Wiener [140], a chip with encryption rate of 191 Mbps costs only \$10.5 because there is very little need for input and output. The cost per Gbps is \$56.4, about 1/5 the cost of DEC chip. By pipeline processing each encryption round by this chip, the encryption rate is 16 times faster than a machine without pipeline processing. As a result, the machine can be manufactured at a cost of \$5.87 per Gbps excluding fixed

cost such as development cost.

Assume conducting cryptanalysis with 50% success probability by exhaustive search within a week. If we calculate the breaking time of such analysis using a $10 million Wiener type machine from expressions (5.2), (5.3), and $c_2(0) = 5.87$, we get about 74 years (until 2070 counting from January 1996). Similarly, in order to make a look-up table covering 50% of key space using time-memory trade-off cryptanalysis within a year using a $50 million Wiener type machine, we will get about 63 years (2057) from expressions (??), (5.3), and $c_2(0) = 5.87$. At this moment, the cost of cryptanalyzing within an hour using an Eberle type machine would be about $5000 from expressions (5.16), (5.23), (5.3), and $c_2(0) = 600$.

Therefore, as long as we assume the above four cryptanalyses, triple DES appears to be a very secure system until the middle of the 21st century. However, there exist two specific cryptanalyses against two-key triple enciphering cryptosystems such as triple DES. We explain each cryptanalysis and estimate durable years for triple DES in the rest of this chapter.

Denote DES by the function $S(K, P)$ and triple DES by the function E:

$$C = E(K, P) = S(K^{(1)}, S^{-1}(K^{(2)}, S(K^{(1)}, P))).$$

Let A and B be intermediate data in $E(K, P)$:

$$A = S(K^{(1)}, P) \quad \text{and} \quad B = S^{-1}(K^{(2)}, A).$$

6.2 Merkle-Hellman's Chosen Plaintext Attack

In 1981, Merkle and Hellman [90] proposed a chosen plaintext attack against triple DES. This attack is a kind of table look-up attack where a cryptanalyst identifies a target key $K = (K^{(1)}, K^{(2)})$ using a look-up table by finding the plaintext-ciphertext pair such that the intermediate value A is zero. This attack takes the following steps.

Step 1 A cryptanalyst creates a look-up table for all the plaintexts that could give $A = 0$ (see Fig. 6.1):

$$P_i = S_i^{-1}(K_i, 0) \qquad (i = 1, 2, \cdots, 2^{56}). \tag{6.1}$$

Step 2 After the cryptanalyst intercepts $C_i = E(K, P_i)$ $(i = 1, 2, \cdots, 2^{56})$, he calculates intermediate value B_i for each C_i using K_i:

$$B_i = S_i^{-1}(K_i, C_i) \qquad (i = 1, 2, \cdots, 2^{56}),$$

and checks if $B_i \in \{P_j = S^{-1}(K_j, 0) \mid j = 1, 2, \cdots, 2^{56}\}$. If $B_i = P_j = S^{-1}(K_j, 0)$ for a certain (i, j), then (K_i, K_j) is a candidate for the target key. Each candidate

Table 6.1: Look-up table in Merkle-Hellman's attack.

P or B	$K^{(1)}$
$P_1 = S^{-1}(K_1, 0)$	K_1
$P_2 = S^{-1}(K_2, 0)$	K_2
$\vdots$	$\vdots$
$P_{2^{56}} = S^{-1}(K_{2^{56}}, 0)$	$K_{2^{56}}$

key found from the look-up table (see Table 6.1) is tested on a few other plaintext-ciphertext pairs. If all of these additional pairs (P_k, C_k) encipher P_k to C_k under the key (K_i, K_j), then the target key $K = (K_i, K_j)$.

Merkle-Hellman's attack [90] against triple DES compared with exhaustive search, shows that required time complexity decreases substantially, while required space complexity and chosen plaintexts increase considerably; which brings the feasibility of such increased factors to the fore. With Merkle-Hellman's attack [90], the feasibility of required memory and also chosen plaintexts for cryptanalysis with 50% success probability is estimated, based on Section 5.3. In this case, the amount of memory is $120 \times 2^{55} \div 2^{30} \simeq 4.03 \times 10^{10}$ Gbit. To build a machine which has this much memory at the cost of $10 million would take about five years (until 2001 counting from January 1996). In addition, to obtain required chosen plaintexts (in this case 2^{55}) within a week, they would have to be collected at the speed of some 1 Tera bps. If we compare the year when we can obtain plaintexts at the speed of 1 Tera bps, it would differ as shown in Table 6.2, by assuming current high-speed encryption as 200Mbps, and by setting three scenarios with respect to the annual growth of processing rate of encryption chip.

Table 6.2: Result of Merkle-Hellman's attack against triple DES.

scenario	speed of increase	year
best	20%	2043
standard	30%	2029
worst	40%	2022

6.3 Van Oorschot-Wiener's Known Plaintext Attack

Van Oorschot and Wiener [135] extended Merkle-Hellman's chosen plaintext attack to a known plaintext attack. This attack requires $2^{112-\log_2 N}$ operations and 2^{56} blocks of memory using N known plaintext-ciphertext pairs. This attack proceeds as follows.

Step 1 After intercepting ciphertexts $C_i = E(K, P_i)$ $(i = 1, 2, \cdots, 2^{56})$, a cryptanalyst tabulates the (P, C) pairs, sorted by plaintext values (see Panel 1 in Table 6.3).

Table 6.3: Look-up table in van Oorschot-Wiener's attack.

Panel 1

P	C
P_1	C_1
P_2	C_2
$\vdots$	$\vdots$
P_N	C_N

Panel 2 (for fixed A)

B	$K^{(2)}$

Step 2 Randomly selecting and fixing a value X for A, the cryptanalyst checks if $S^{-1}(K_i, X)$ $(i = 1, 2, \cdots, 2^{56})$ exists in the first column of Panel 1.

Step 3 If $P_k = S^{-1}(K_i, X)$ exists, then the cryptanalyst takes the corresponding ciphertext value C_k, computes the intermediate value $B_{i,k} = S^{-1}(K_i, C_k)$ and places it along with key K_i into the second table (see Panel 2 in Table 6.3). He completes Panel 2 in a similar manner.

Step 4 The cryptanalyst checks if $S^{-1}(K_j, X)$ $(j = 1, 2, \cdots, 2^{56})$ exists in the first column of Panel 2. If $B_{i,k} = S^{-1}(K_j, X)$ exist, then the key (K_i, K_j) is a candidate for the target key. It will be tested on a few other plaintext-ciphertext pairs. If all the key candidates derived here turn out not to be the target key, the cryptanalyst will return to Step 2, select a different value for A, and search for the target key following the same procedure.

The expected value of time and space required in this estimation are as follows. The required storage capacity for Panel 1 is $120N$ bit. For each value of X that is tried, the time required to build Panel 2 is in the order of 2^{56}, assuming that Panel 1 is sorted on the plaintext values so that look-ups take constant time. Because only 2^{56} out of 2^{64}

possible plaintexts are searched in Panel 1, the expected number of entries in Panel 2 is $N/2^8$. This space is reusable across different values of A. The time required to work with Panel 2 to find candidate pairs of keys is in the order of 2^{56}.

The probability of selecting a value of A that leads to success is $N/2^{64}$. The expected number of draws required to draw one red ball out of a bin containing N red balls and $M - N$ green balls is $(M + 1)/(N + 1)$ if the balls are not replaced. Therefore, the expected number of draws that must be tried is $(2^{64} + 1)/(N + 1) \simeq 2^{64}/N$. Thus, the expected running time for the attack is in the order of $(2^{56} \times 2^{64})/N = 2^{120-\log_2 N}$ and required storage capacity, $120N$ bit. On this basis, we estimate the case where known plaintexts, obtained in a week, are analyzed by van Oorschot-Wiener's attack, within a week, by a $10 million Wiener type machine. Under the scenarios specified in Table 6.2, the number of known plaintexts which can be obtained in a week $(\tau - 1)$ years later will be:

$$\begin{aligned} N(\tau) &= 200 \times 2^{20} \times (1 + \alpha)^{(\tau-1)} \times (3600 \times 24) \div 64 \\ &\simeq 1.80 \times 2^{40} \times (1 + \alpha)^{(\tau-1)} \end{aligned} \tag{6.2}$$

where $\alpha = 0.2,\ 0.3$, and 0.4.

Therefore, the cryptanalyzing cost $C_{OW}(\tau)$[$M] of $(\tau - 1)$ later years will be

$$\begin{aligned} C_{OW}(\tau) &= 2^{112-\log_2 N(\tau)} \times 5.87 \times 0.65^{\tau} \times 64 \times 0.5 \div (3600 \times 24 \times 7) \div 10^6 \\ &\simeq 1.63 \times 10^{12} \times (1 + \alpha) \times \left(\frac{0.65}{1 + \alpha}\right)^{\tau}. \end{aligned} \tag{6.3}$$

Given expressions (6.2), (6.3), and $C_{OW}(\tau) = 10$, we can estimate the year when the cryptanalysis considered here becomes feasible and the number of known plaintext-ciphertext pairs which the cryptanalyst uses at that time according to each scenario. The second column of Table 6.4 shows the number of plaintext-ciphertext pairs and the third column the year when the cryptanalysis considered here is expected to become feasible.

Table 6.4: Result of van Oorschot-Wiener's attack against triple DES.

scenario	plaintexts	year
best	1.53×2^{51}	2037
standard	1.36×2^{54}	2032
worst	1.39×2^{56}	2028

Let us synthesize our findings obtained in tables 6.2 and 6.4 in Table 6.5. Table 6.5 suggests the following: (1) As far as the processing rate of encryption chip grows not

more than +30% annually, triple DES would serve as a very high secure system until around 2030. (2) Even under the worst scenario of 40% annual growth in encryption rate, triple DES would be secure until around 2020.

Table 6.5: Result of two attacks against triple DES.

scenario	attack	plaintexts	year
best	Van Oorschot-Wiener's	$2^{51.6}$	2037
standard	Merkle-Hellman's	2^{55}	2029
worst	Merkle-Hellman's	2^{55}	2022

Chapter 7

Conclusion

DES (data encryption standard) has been the most widely used cryptosystem, especially in the area of financial applications, since its adoption as a federal U.S. standard in 1977. It has also been the subject of much controversy in terms of how secure it is.

The security of a cryptosystem can be assessed by subjecting it to various cryptanalytic attacks under circumstances considered favorable to the cryptanalyst. In this paper, we have attempted to present an exhaustive review of literature on such cryptanalytic attacks against DES, including the short-cut method (differential and linear cryptanalyses) and brute force method (exhaustive search and time-memory trade-off analysis). We have particularly focused on the importance of time-memory trade-off analysis and presented some new evidence based on our own numerical examples.

Major conclusions are summarized as follows.

In Chapter 3, we examined the basic and internal structure of DES. The basic structure is deemed as superior because of the following three properties: (i) DES is not closed and generates a sufficiently large subgroup; (ii) OFB (output feedback) mode performs quite similar to random bijective functions; and (iii) strength against differential and linear cryptanalyses can be achieved by increasing the number of rounds. Assessment of the internal structure, especially the DES S-boxes, is rather complicated: some measures imply that it does not satisfy SAC (strict avalanche criteria) and the distance to affine functions is relatively low; and other measures more related to actual cryptanalysis suggest that, except for relatively poor performance against linear cryptanalysis, it contains favorable properties such as regularity, absence of linear structure, and high R-robustness. Since the size of DES S-boxes does not allow the simultaneous satisfaction of all these measures, we can deem DES S-boxes as being a well-balanced good structure. Also, examining the diffusion function of a P-box and key schedule from the viewpoint of achieving completeness led us to the conclusion that, given the structure of

an SP-network, these diffusion components are well designed.

Chapter 4 attempted to assess the cryptographic strength of DES against the short-cut methods. DES has been subject to various attacks using short-cut methods such as the formal coding approach and meet-in-the-middle-attack, although as far as the authors know, attacks which are superior (at least in some respects) to brute force methods are limited to improved Davies' attack, differential cryptanalysis, and linear cryptanalysis. The differences in the three cryptanalyses are as follows: (i) texts required for analysis are chosen-plaintexts for differential cryptanalysis and known-plaintexts for improved Davies' attack and linear cryptanalysis; and (ii) the approximate number of plaintext-ciphertext pairs needed to deduce the correct key with high probability is 2^{50} in improved Davies' attack and 2^{47} in differential cryptanalysis compared with 2^{43} in linear cryptanalysis. Since the accuracy of these approximations is believed to be satisfactory, we can say with confidence that linear cryptanalysis is the best among the three. Accordingly, based on the approximations, we estimated success probability, time needed to obtain required plaintexts, breaking time, and analyzing cost. As a result, if one assumes a cryptanalyst obtains plaintext-ciphertext pairs at the speed of 200 Mbps, cryptanalysis of 50% success probability with breaking time of three to seven days would be estimated as possible at a cost of \$0.1 - \$0.6 million. However, linear cryptanalysis would not be the immediate threat since the circumstance assumed above are not feasible given current telecommunication network systems among banks and among banks and firms. This view would, of course, not lessen the importance of monitoring the theoretical developments of linear cryptanalysis: this assessment is no more than an approximation, and some improvement in the analysis might be proposed in the near future.

In Chapter 5, we examined the cryptographic strength of DES against the brute force methods. In addition to exhaustive search, we focused on an optimized method of, rather overlooked, time-memory trade-off cryptanalysis.

We have compared the various techniques in exhaustive search using the data in published articles of trial computation performed by the presenters themselves. The result was that the apparatus presented by Wiener, equipped with special chips which realizes each round encryption by total pipeline processing, is the best in terms of cost-performance. If Wiener's trial computation is in fact true, it can be stated that crypt-analyzing DES in a short time by exhaustive search is no longer unrealistic. However, such a machine has not yet been actually built, and therefore the accuracy of his trial computation has not been verified.

Therefore, in this article, we have chosen another approach. We assumed a simple parallel processing machine using GaAs chips presented by Eberle and performed a trial

computation on the relation between searching time and cost (variable cost only) for exploring a half of the key space of DES. The result was that it was still very difficult to do the search in one day due to formidable cost. However, we obtained some figures implying that a search in a month can no longer be stated as totally unrealistic. Also, we got some figures suggesting that in the year 2001, a search in one day can no longer be stated as totally unrealistic.

In the ciphered data communication system among banks, the common view is that authentication is more important than confidentiality. Hence, there maybe an idea of increasing the frequency of key exchange to about once in an hour as a tentative measure to protect the efficacy of authentication from the attack of exhaustive search. This measure may be effective in a system where a cryptanalyst can obtain known plaintexts only rarely. However, in a system where a cryptanalyst can obtain known plaintexts easily, we showed that increasing the frequency of key exchange is not effective because of the possibility of getting an attack in waves.

For those systems in which fixed plintext unchanged for a long time is transferred either right after each key exchange or at the beginning of each session, or those systems in which a chosen plaintext attack is possible, an attack in time-memory trade-off cryptanalysis should be kept in mind. Time-memory trade-off cryptanalysis is a method which decreases the amount of computation after intercepting the ciphertext by storing a look-up table beforehand which was built by exhaustive search against predicted plaintext in an environment where a plaintext which will be transferred can be foreseen far before the actual interception. We have done a trial computation assuming an Eberle type machine as a key search machine, and the result was that as for building a look-up table in order to explore more than a half of the key space of DES, it takes 2.5 times long time compared with the total hours required for exhaustive search. But once the look-up table is prepared, we obtained some data implying that cryptanalysis in 10 minutes or less may no longer be stated as totally unrealistic. In addition, it was shown that in a ciphered data communication system where interception of a common fixed plaintext against multiple lines is relatively easy, a cryptanalyst can attack multiple lines one after another in the aim of masquerading oneself as an indefinite legitimate user. It was shown that in this case, the amount of precomputation will be substantially lesser, and the size of the machine be much smaller, both compared with exhaustive search.

In Chapter 6, we assessed the cryptographic strength of triple DES, a cryptosystem which the banking industry expects as the most likely alternative to DES. As long as we assume the four kinds of attacks, namely, differential and linear cryptanalyses, exhaustive search, and optimized time-memory trade-off cryptanalysis, triple DES appeared to be a very secure system until the middle of the 21st century. As for Merkle Hellman's attack

and Van Oorschot-Wiener's attack, estimates showed that triple DES would be feasible as a secure system until about 2020 even under assumptions considered most favorable to the cryptanalyst.

Bibliography

[1] M. Abramowitz, *Handbook of mathematical functions with formulas, graphs and mathematical tables*, Applied Mathematical Series, Vol.55, National Bureau of Standards, June 1964.

[2] C.M. Adams and S.E. Tavares, "The structured design of cryptographically good S-boxes,"Journal of Cryptology, Vol.3, No.1, pp.27-41, 1990.

[3] C.M. Adams and S.E. Tavares, "The use of bent sequences to achieve higher-order strict avalanche criterion,"Technical Report TR90-013, Queen's University, 1990.

[4] L. Adleman, "Molecular computation of solutions to combinatorial problems,"Science, Vol.266, pp.1021-1024, November 1994.

[5] ANSI, "ANSI X3.106: Data encryption algorithm – Modes of operation,"1983.

[6] K. Aoki and K. Kurokawa, "A study on linear cryptanalysis of Multi2 (in Japanese),"Proceedings of The 1995 Symposium on Cryptography and Information Security, SCIS95-A4.1, IEICE (Japan), 1995.

[7] F. Ayoub, "Probabilistic completeness of substitution-permutation encryption networks ,"IEE, Vol.E-129, No.5, pp.195-199, September 1995.

[8] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems,"Advances in Cryptology – Proceedings of CRYPTO'90, Lecture Notes in Computer Science, Vol.537, pp.2-21, Springer-Verlag, 1991.

[9] E. Biham and A. Shamir, "Differential cryptanalysis of FEAL and N-hash,"Advances in Cryptology – Proceedings of EUROCRYPT'91, Lecture Notes in Computer Science, Vol.547, pp.1-16, Springer-Verlag, 1991.

[10] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems,"Journal of Cryptology, Vol.4, No.1, pp.3-72, 1991.

[11] E. Biham and A. Shamir, "Differential cryptanalysis of the full 16-round DES,"Advances in Cryptology – Proceedings of CRYPTO'92, Lecture Notes in Computer Science, Vol.740, pp.487-496, Springer-Verlag, 1993.

[12] E. Biham and A. Shamir, *Differential cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1994.

[13] E. Biham, "New types of cyptanalytic attacks using related keys,"Advances in Cryptology – Proceedings of EUROCRYPT'93, Lecture Notes in Computer Science, Vol.765, pp.398-409, Springer-Verlag, 1994.

[14] E. Biham and A. Biryukov, "An improvement of Davies' attack,"Advances in Cryptology – Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science, Vol.950, pp.461-467, Springer-Verlag, 1995.

[15] M. Blaze, W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Wiener, "Minimal key lengths for symmetric ciphers to provide adequate commercial security,"A Report by an Ad Hoc Group of Cryptographers and Computer Scientists, January 1996.

[16] D. Boneh, C. Dunworth, and R.J. Lipton, "Breaking DES using a molecular computer,"Princeton CS Tech-Report, No.489, 1995.

[17] E.F. Brickell, J.H. Moore, and M.R. Purtill, "Structure in the S-boxes of the DES,"Advances in Cryptology – Proceedings of CRYPTO'86, Lecture Notes in Computer Science, Vol.263, pp.3-7, Springer-Verlag, 1987.

[18] E.F. Brickell, D.E. Denning, S.T. Kent, and D.P. Maher, "The Skipjack algorithm – Skipjack review interim report,"July 1993.

[19] L. Brown, "A proposed design for an extended DES,"Proceedings of Fifth International Conference and Exhibition on Computer Security, IFIP, May 1988.

[20] L. Brown and J. Seberry, "On the design of permutation P in DES type cryptosystems,"Advances in Cryptology – Proceedings of EUROCRYPT'89, Lecture Notes in Computer Science, Vol.434, pp.696-705, Springer-Verlag, 1990.

[21] L. Brown and J. Seberry, "Key scheduling in DES type cryptosystems,"Advances in Cryptology – Proceedings of ASIACRYPT'90, Lecture Notes in Computer Science, Vol.453, pp.221-228, Springer-Verlag, 1990.

[22] L. Brown, J. Pieprzyk, and J. Seberry "LOKI – A cryptographic primitive for authentication and secrecy applications,"Advances in Cryptology – Proceedings of ASIACRYPT'90, Lecture Notes in Computer Science, Vol.453, pp.229-236, Springer-Verlag, 1990.

[23] L. Brown, M. Kwan, J. Pieprzyk, and J. Seberry, "Improving resistance to differential cryptanalysis and the redesign of LOKI,"Advances in Cryptology – Proceedings of ASIACRYPT'91, Lecture Notes in Computer Science, Vol.739, pp.36-50, Springer-Verlag, 1993.

[24] K.W. Campbell and M.J. Wiener, "DES is not a group,"Advances in Cryptology – Proceedings of CRYPTO'92, Lecture Notes in Computer Science, Vol.740, pp.512-520, Springer-Verlag, 1993.

[25] F. Chabaud and S. Vaudenay, "Links between differential and linear cryptanalysis,"Advances in Cryptology – Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science, Vol.950, pp.356-365, Springer-Verlag, 1995.

[26] D. Chaum and J.-H. Evertse, "Cryptanalysis of DES with a reduced number of rounds, sequences of linear factors in block ciphers,"Advances in Cryptology – Proceedings of CRYPTO'85, Lecture Notes in Computer Science, Vol.218, pp.192-211, Springer-Verlag, 1986.

[27] S. Chee, S. Lee, and K. Kim, "Semi bent functions,"Advances in Cryptology – Proceedings of ASIACRYPT'91, Lecture Notes in Computer Science, Vol.917, pp.107-118, Springer-Verlag, 1995.

[28] R. Cleve, "Complexity theoretic issues concerning block ciphers related to DES,"Advances in Cryptology – Proceedings of CRYPTO'90, Lecture Notes in Computer Science, Vol.537, pp.530-544, Springer-Verlag, 1991.

[29] G.D. Cohen, M.G. Karpovsky, and H.F.Jr. Mattson, and J.R. Schatz, "Covering radius – Survey and recent results,"IEEE Transactions on Information Theory, Vol.IT-31, No.3, pp.328-343, 1985.

[30] D. Coppersmith and E. Grossman, "Generators for certain alternating groups with applications to cryptography,"SIAM J. Appl. Math., pp.624-627, 1975.

[31] D. Coppersmith, "The Data Encryption Standard (DES) and its strength against attacks,"IBM Journal of Reseach and Development, Vol.8, No.3, pp.243-50, May 1994.

[32] J. Daemen, *Cipher and hash function design*, Doctoral Dissertation, Katholieke Universiteit Leuven, 1995.

[33] D.W. Davies, "Some regular properties of the Data Encryption Standard," Advances in Cryptology – Proceedings of CRYPTO'82, pp.89-96, Plenum Press, 1983.

[34] D.W. Davies and G.I.P. Parkin "The average cycle size of the key stream in output feedback encipherment," Advances in Cryptology – Proceedings of CRYPTO'82, pp.97-98, Plenum Press, 1983.

[35] D.W. Davies and S. Murphy, "Pairs and triplets of DES S-boxes," Journal of Cryptology, Vol.8, No.1, pp.1-25, 1995.

[36] M. Davio, "Ring-sum expansions of boolean functions," Proceedings of Symposium on Computers and Automata, pp.411-418, Polytechnic Institute of Brooklyn, April 1971.

[37] M. Davio, Y. Desmedt, W. Fosseprez, R. Govaerts, J. Hulsbosch, P. Neutjens, P. Piret, J.-J. Quisquater, J. Vandewalle, and P. Woulters, "Analytical characteristics of the DES," Advances in Cryptology – Proceedings of CRYPTO'83, pp.171-202, Plenum Press, 1984.

[38] M. Davio, Y. Desmedt, and J.-J. Quisquater, "Propagation characteristics of the DES," Advances in Cryptology – Proceedings of EUROCRYPT'84, Lecture Notes in Computer Science, Vol.209, pp.62-73, Springer-Verlag, 1984.

[39] M.H. Dawson and S.E. Tavares, "An expanded set of S-box design criteria based on information theory and its relation to differential-like attacks," Advances in Cryptology – Proceedings of EUROCRYPT'91, Lecture Notes in Computer Science, Vol.547, pp.352-367, Springer-Verlag, 1991.

[40] W. Diffie and M.E. Hellman, "Exhaustive Search of the NBS Data Encryption Standard," Computer, Vol.10, No.6, pp.74-84, June 1977.

[41] J.F. Dillon, "A survey of bent functions," *The NSA Technical Journal*, pp.191-215, 1972.

[42] H. Eberle, "High-speed DES implementation for network applications," Advances in Cryptology – Proceedings of CRYPTO'92, Lecture Notes in Computer Science, Vol.740, pp.521-539, Springer-Verlag, 1993.

[43] S. Even and O. Goldreich, "DES-like functions can generate the alternating group," IEEE Transactions on Information Theory, Vol.IT-29, No.6, pp.863-865, 1983.

[44] J.-H. Evertes, "Linear structures in block ciphers,"Advances in Cryptology – Proceedings of EUROCRYPT'87, Lecture Notes in Computer Science, Vol.304, pp.249-266, Springer-Verlag, 1988.

[45] H. Feistel, "Cryptography and computer privacy,"SCIENTIFIC AMERICAN, Vol.228, No.5, pp.15-23, May 1973.

[46] R. Forré, "Methods and Instruments for designing S-boxes,"Journal of Cryptology, Vol.2, No.3, pp.115-130, 1990.

[47] R. Forré, "The strict avalanche criterion: spectral properties of Boolean functions and an extended definition,"Advances in Cryptology – Proceedings of CRYPTO'88, Lecture Notes in Computer Science, Vol.403, pp.450-468, Springer-Verlag, 1990.

[48] W. Fumy, "On the F-function of FEAL,"Advances in Cryptology – Proceedings of CRYPTO'87, Lecture Notes in Computer Science, Vol.293, pp.434-437, Springer-Verlag, 1988.

[49] J. Gait, "A new nonlinear pseudorandom number generator,"IEEE Transactions on Software Engineering, Vol.SE-3, pp.359-363, September 1977.

[50] G. Garon and R. Outerbridge, "DES Watch : an examination of the Data Encryption Standard for financial institution information security in the 1990's,"Cryptologia, Vol.15, No.3, pp.177-193, 1991.

[51] H. Gilbert and G. Chassé, "A statistical attack of the FEAL-8 cryptosystem,"Advances in Cryptology – Proceedings of CRYPTO'90, Lecture Notes in Computer Science, Vol.537, pp.22-33, Springer-Verlag, 1991.

[52] B. Harris, "Probability distribution related to random mapping,"Ann. Math. Statistics, Vol.31, pp.1045-1062, 1959.

[53] M.E. Hellman, R. Merkle, R. Schroeppel, L. Washington, W. Diffie, S. Polig, and P. Schweitzer, "Results of an initial attempt to cryptanalyze the NBS Data Encryption Standard,"SEL 76-042, Stanford University, 1976.

[54] M.E. Hellman, "A cryptanalytic time-memory trade-off,"IEEE Transactions on Information Theory, Vol.IT-26, No.4, pp.401-406, July 1980.

[55] M.E. Hellman and J.M. Reyneri,"Drainage and the DES,"Advances in Cryptology – Proceedings of CRYPTO'82, pp.129-131, Plenum Press, 1983.

[56] ISO/IEC JTC1/SC27, "ISO 9160: Information processing – Data encipherment – Physical layer interoperability requirements,"International Organization for Standardization, 1988.

[57] ISO/IEC JTC1/SC27, "ISO 10116: Information technology – Security techniques – Modes of oeration of an n-bit block cipher algorithm,"International Organization for Standardization, 1991.

[58] ISO/IEC JTC1/SC27, "2nd CD 13888-2: Information technology – Security techniques – Non-repudiation – Part 2: Using symmetric encipherment algorithms,"International Organization for Standardization, 1995.

[59] ISO/TC68, "ISO 8732: Banking – Keymanagement (wholesale), "International Organization for Standardization, 1988.

[60] ISO/TC68, "ISO 10126-2: Banking – Proceedings for message encipherment (wholesale) – Part 2: DEA Algorithm,"International Organization for Standardization, 1991.

[61] ISO/TC97, "ISO 8372: Information Processing – Modes of operation for a 64-bit block cipher algorithm,"International Organization for Standardization, 1987.

[62] R.R. Jueneman, "Analysis of certain aspects of output feedback mode ,"Advances in Cryptology – Proceedings of CRYPTO'82, pp.99-127, Plenum Press, 1983.

[63] B.S. Kaliski, R.L. Rivest, and A.T. Sherman, "Is the Data Encryption Standard a group (results on cycling experiments on DES),"Journal of Cryptology, Vol.1, No.1, pp.3-36, 1988.

[64] B.S. Kaliski and M.J.B. Robshaw, "Linear Cryptanalysis using multiple approximations,"Advances in Cryptology – Proceedings of CRYPTO'94, Lecture Notes in Computer Science, Vol.839, pp.26-39, Springer-Verlag, 1994.

[65] J.B. Kam and G.I. Davida, "Structured design of substitution-permutation encryption networks,"IEEE Transactions on Computers, Vol.c-28, No.10, pp.747-753, October 1979.

[66] T. Kano, K. Kusuda, and T. Matsumoto, "On validity of optimized time-memory trade-off cryptanalysis (in Japanese),"Technical Report of The Institute of Electronics, Information and Communication Engineers, ISEC95-12, 1995.

[67] K. Kim, *A study on the construction and analysis of substitution boxes for symmetric cryptosystems*, Doctoral Dissertation, Yokohama National University, 1990.

[68] K. Kim, "Construction of DES-like S-boxes based on boolean functions satisfying the SAC,"Advances in Cryptology – Proceedings of ASIACRYPT'91, Lecture Notes in Computer Science, Vol.739, pp.59-72, Springer-Verlag, 1993.

[69] K. Kim, S. Lee, P. Sangjun and D. Lee, "Reconstruction of s^2DES S-boxes and their immunity to differential cryptanalysis,"Technical Report of The Institute of Electronics, Information and Communication Engineers, ISEC94-15D, 1994.

[70] K. Kim, S. Lee, P. Sangjun, and D. Lee, "How to strengthen DES against two robust attacks,"Proceedings of 1995 Japan-Korea Joint Workshop on Information Security and Cryptology, pp.173-182, 1995.

[71] L.R. Knudsen, "Cryptanalysis of LOKI91,"Advances in Cryptology – Proceedings of AUSCRYPT'92, Lecture Notes in Computer Science, Vol.718, pp.196-208, Springer-Verlag, 1993.

[72] L.R. Knudsen, "Iterative Characteristics of DES and s^2-DES,"Advances in Cryptology – Proceedings of CRYPTO'92, Lecture Notes in Computer Science, Vol.740, pp.497-511, Springer-Verlag, 1993.

[73] L.R. Knudsen, "New potentially 'weak' keys for DES and LOKI,"Advances in Cryptology – Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science, Vol.950, pp.419-424, Springer-Verlag, 1995.

[74] L.R. Knudsen, "Truncated and higher order differentials, "Proceedings of Fast Software Encryption, Lecture Notes in Computer Science, Vol.1008, pp.196-211, Springer-Verlag, 1995.

[75] D.E. Knuth, *The art of computer programming, Volume II*, Reading, Mass., Addison-Wesley, 1973.

[76] A.G. Konheim, *Cryptography: A Primer*, John Wiley and Sons Inc., 1981.

[77] K. Kusuda and T. Matsumoto "Optimization of time-memory trade-off cryptanalysis and its application to DES, FEAL-32, and Skipjack,"IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, Vol.E79-A, No.1, IEICE Engineering Science Society (Japan), pp.35-48, January 1996.

[78] X. Lai, J.L. Massey, and S. Murphy "Markov ciphers and differential cryptananlysis,"Advances in Cryptology – Proceedings of EUROCRYPT'91, Lecture Notes in Computer Science, Vol.547, pp.17-38, Springer-Verlag, 1991.

[79] X. Lai, "Additive and linear structures of cryptographic functions,"Proceedings of Fast Software Encryption, Lecture Notes in Computer Science, Vol.1008, pp.75-85, Springer-Verlag, 1995.

[80] S.K. Langford and M.E. Hellman "Differential-linear cryptananlysis,"Advances in Cryptology – Proceedings of CRYPTO'94, Lecture Notes in Computer Science, Vol.839, pp.17-25, Springer-Verlag, 1994.

[81] R.J. Lipton, "Using DNA to solve NP-complete problem,"Science, Vol.268, pp.542-545, April 1995.

[82] M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions,"SIAM J. Comput., Vol.17, No.2, pp.373-386, 1988.

[83] F.J. MacWilliams and N.J.A. Sloane, *The theory of error-correcting codes*, North-Holland, 1977.

[84] M. Matsui, "Linear cryptanalysis method for DES cipher,"Advances in Cryptology – Proceedings of EUROCRYPT'93, Lecture Notes in Computer Science, Vol.765, pp.386-397, Springer-Verlag, 1993.

[85] M. Matsui and Yamagishi, "On a statistical attack of secret key cryptosystems (in Japanese),"The Transactions of The Institute of Electronics, Information, and Communication Engineers, Vol.77-a No.3, pp.476-484, March 1994.

[86] M. Matsui, "On correlation between the order of S-boxes and the strength of DES,"Advances in Cryptology – Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science, Vol.950, pp.366-375, Springer-Verlag, 1994.

[87] M. Matsui, "The first experimental cryptanalysis of the Data Encryption Standard,"Advances in Cryptology – Proceedings of CRYPTO'94, Lecture Notes in Computer Science, Vol.839, pp.1-11, Springer-Verlag, 1994.

[88] M. Matsui, "On provable security of block ciphers against differential and linear cryptanalysis,"Proceedings of The 18th Symposium on Information Theory and Its Applications (SITA95), C-2-5, IEICE (Japan), 1995.

[89] W. Meier and O. Staffelbach, "Nonlinearity criteria for cryptographic functions,"Advances in Cryptology – Proceedings of EUROCRYPT'89, Lecture Notes in Computer Science, Vol.434, pp.549-562, Springer-Verlag, 1990.

[90] R.C. Merkle and M.E. Hellman "On the security of multiple encryption," Communications of the ACM, Vol.24, No.7, pp.465-467, July 1981.

[91] C.H. Meyer and S.M. Matyas, *Cryptography: a new dimension in data security,* John Wiley & Sons, 1982.

[92] C. Mitchell, "Enumerating boolean functions of cryptographic significance,"Journal of Cryptology, Vol.2, No.3, pp.155-170, 1990.

[93] S. Miyaguchi, A. Shiraishi, and A. Shimizu, "Fast data encipherment algorithm FEAL-8,"Review of Electrical Communication Laboratories, Vol.36, No.4, pp.321-327, NTT, 1988.

[94] S. Miyaguchi, "The FEAL cipher family,"Advances in Cryptology – Proceedings of CRYPTO'90, Lecture Notes in Computer Science, Vol.537, pp.627-638, Springer-Verlag, 1991.

[95] J.H. Moore and G.J. Simmons "Cycle structure of the DES for keys having palindromic (or antipalindromic) sequences of round keys,"IEEE Transactions on Software Engineering, Vol.SE-13, No.2, pp.262-273, 1987.

[96] M. Morita, K. Ohta, and K. Takagi, "Is DES closed —another evidence by switching closure test,"Technical Report of IEICE, ISEC92-49, Vol.92, No.260, pp.25-32, 1992.

[97] National Bureau of Standards, "Data Encryption Standard,"U.S.Department of Commerce, Federal Information Processing Standards Publication, Vol.46, January 1977.

[98] National Bureau of Standards, "DES modes of operation,"U.S.Department of Commerce, Federal Information Processing Standards Publication, Vol.81, September 1981.

[99] K. Nyberg, "Perfect nonlinear S-boxes,"Advances in Cryptology – Proceedings of EUROCRYPT'91, Lecture Notes in Computer Science, Vol.547, pp.378-386, Springer-Verlag, 1991.

[100] K. Nyberg and L.R. Knudsen, "Provable security against differential cryptanalysis,"Advances in Cryptology – Proceedings of CRYPTO'92, Lecture Notes in Computer Science, Vol.740, pp.566-574, Springer-Verlag, 1993.

[101] K. Nyberg, "Linear approximation of block ciphers,"Advances in Cryptology – Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science, Vol.950, pp.439-444, Springer-Verlag, 1995.

[102] L. O'Connor, "On the distribution of characteristics in composite permutations," Advances in Cryptology – Proceedings of CRYPTO'93, Lecture Notes in Computer Science, Vol.773, pp.403-412, Springer-Verlag, 1994.

[103] L. O'Connor, "An analysis of a class of algorithms for S-box construction," Journal of Cryptology, Vol.7, No.3, pp.133-151, 1994.

[104] L. O'Connor and A. Klapper, "Algebraic nonlinearity and its applications to cryptography," Journal of Cryptology, Vol.7, No.4, pp.213-227, 1994.

[105] L. O'Connor and J.D. Golić, "A unified Markov approach to differential and linear cryptanalysis," Advances in Cryptology – Proceedings of ASIACRYPT'94, Lecture Notes in Computer Science, Vol.917, pp.387-397, Springer-Verlag, 1995.

[106] L. O'Connor, "On the distribution of characteristics in bijective mappings," Journal of Cryptology, Vol.8, No.2, pp.67-86, 1995.

[107] K. Ohta and K. Aoki, "Linear cryptanalysis of the Fast Data Encipherment Algorithm," Advances in Cryptology – Proceedings of CRYPTO'94, Lecture Notes in Computer Science, Vol.839, pp.12-16, Springer-Verlag, 1994.

[108] K. Ohta, S. Moriai, and K. Aoki, "Improving the search algorithm for the best linear expression," Advances in Cryptology – Proceedings of CRYPTO'95, Lecture Notes in Computer Science, Vol.963, pp.157-170, Springer-Verlag, 1995.

[109] J. Pieprzyk and G. Finkelstein "Towards effective nonlinear cryptosystem design," IEE Proceedings, Vol.135, Pt.E, No.6, pp.325-335, November 1988.

[110] B. Preneel, W. Van Leekwijck, and L. Van Linden, "Propagation characteristics of Boolean functions," Advances in Cryptology – Proceedings of EUROCRYPT'90, Lecture Notes in Computer Science, Vol.473, pp.161-173, Springer-Verlag, 1991.

[111] B. Preneel, *Analysis and design of cryptographic functions*, Doctoral Dissertation, Katholieke Universiteit Leuven, 1993.

[112] B. Preneel, M. Nuttin, V. Rijmen, and J. Buelens, "Cryptanalysis of the CFB mode of the DES with a reduced number of rounds," Advances in Cryptology – Proceedings of CRYPTO'93, Lecture Notes in Computer Science, Vol.773, pp.212-223, Springer-Verlag, 1993.

[113] P.W. Purdon and J.H. Williams, "Cycle length in a random function," Ann. Math. Soc., Vol.133, pp.547-551, 1968.

[114] J.A. Reeds and J.L. Manferdelli, "DES has no per round linear factors,"Advances in Cryptology – Proceedings of CRYPTO'84, Lecture Notes in Computer Science, Vol.196, pp.377-389, Springer-Verlag, 1985.

[115] V. Rijmen, B. Preneel, R. Govaerts, and J. Vandewalle, "On using maximum likelihood to optimize recent cryptanalytic techniques,"Presented at the Rump Session of EUROCRYPT'94, 1994.

[116] O.S. Rothaus, "On bent functions,"Journal of Combinatorial Theory, Vol.A-20, pp.300-305, 1976.

[117] RSA Data Security Inc., "Government encryption standard DES takes a fall,"http://www.rsa.com.des/, June 1996.

[118] R.A. Rueppel, *Analysis and design of stream ciphers*, Springer-Verlag, 1986.

[119] I. Schaumüller-Bichl, "The method of formal coding,"In Cryptography: Proceedings of Workshop Cryptography, Lecture Notes in Computer Science, Vol.149, pp.235-255, Springer-Verlag, 1983.

[120] J. Seberry, X.-M. Zhang, and Y. Zheng "Systematic generation of cryptographically robust S-boxes,"Proceedings of the first ACM Conference on Computer and Communications Security, pp.172-182, The Association for Computing Machinery, 1993.

[121] J. Seberry, X.-M. Zhang, and Y. Zheng "On constructions and nonlinearity of correlation immune functions,"Advances in Cryptology – Proceedings of EUROCRYPT'93, Lecture Notes in Computer Science, Vol.765, pp.181-199, Springer-Verlag, 1994.

[122] J. Seberry, X.-M. Zhang, and Y. Zheng "Pitfalls in designing substitution boxes (extended abstract),"Advances in Cryptology – Proceedings of CRYPTO'94, Lecture Notes in Computer Science, Vol.839, pp.383-396, Springer-Verlag, 1994.

[123] J. Seberry, X.-M. Zhang, and Y. Zheng "Relationships among nonlinearity criteria,"Advances in Cryptology – Proceedings of EUROCRYPT'94, Lecture Notes in Computer Science, Vol.950, pp.376-388, Springer-Verlag, 1995.

[124] J. Seberry, X.-M. Zhang, and Y. Zheng "Nonlinearity and propagation characteristics of balanced boolean functions," Information and Computation, Cryptology, Vol.119, No.1, 1995.

[125] A. Shamir, "On the security of DES,"Advances in Cryptology – Proceedings of CRYPTO'85, Lecture Notes in Computer Science, Vol.218, pp.280-281, Springer-Verlag, 1986.

[126] C.E. Shannon, "Communication theory of secrecy systems,"Bell System Technical Journal, Vol.28, pp.656-715, 1949.

[127] T. Siegenthaler, "Correlation immunity of non-linear combining functions for cryptographic application,"IEEE Transactions on Information Theory, Vol.IT-30, No.5, pp.776-780, 1984.

[128] T. Sorimachi, T. Tokita, and M. Matsui, "On a cipher evaluation method based on differential cryptanalysis (in Japanese),"Poceedings of The 1994 Symposium on Cryptography and Information Security, SCIS94-4C, IEICE (Japan), January 1994.

[129] A. Sorkin, "Lucifer, a cryptographic algorithm,"Cryptologia, Vol.8, No.1, pp.22-41, January 1984.

[130] K. Takaragi, K. Sasaki, and F. Nakagawa, "Multi-media encryption algorithm (in Japanese),"89-MDP-40-5, January 1989.

[131] M. Takeda and T. Kaneko, "The best 128 linear approximate equations of DES (in Japanese),"Proceedings of The 1996 Symposium on Cryptography and Information Security, SCIS96-11B, IEICE (Japan), January 1996.

[132] A. Terashima and T. Kaneko, "On the quadratic approximation cryptanalysis of DES (in Japanese),"Proceedings of The 18th Symposium on Information Theory and Its Applications, C-2-6, October 1995.

[133] T. Tokita, T. Sorimachi, and M. Matsui, "Linear cryptanalysis of LOKI and s^2DES,"Advances in Cryptology – Proceedings of ASIACRYPT'94, Lecture Notes in Computer Science, Vol.917, pp.293-303, Springer-Verlag, 1995.

[134] W. Tuchman, "Hellman presents no shortcut method to the DES,"IEEE Spectrum, Vol.16, No.7, pp.40-41, Springer-Verlag, July 1979.

[135] P.C. van Oorschot and M.J. Wiener, "A known plaintext attack on two-key triple encryption,"Advances in Cryptology – Proceedings of EUROCRYPT'90, Lecture Notes in Computer Science, Vol.473, pp.318-325, Springer-Verlag, 1990.

[136] S. Vaudenay, "Statistical cryptanalysis,"Proceedings of the 3rd ACM Conference.

[137] P. Wayner, "Content-addressable search engines and DES-like systems," Advances in Cryptology – Proceedings of CRYPTO'92, Lecture Notes in Computer Science, Vol.740, pp575-586, Springer-Verlag, 1993.

[138] A.F. Webster and S.E. Tavares, "On the design of S-boxes," Advances in Cryptology – Proceedings of CRYPTO'85, Lecture Notes in Computer Science, Vol.218, pp.523-534, Springer-Verlag, 1986.

[139] R. Wernsdorf, "The one-round functions of the DES generate the alternating group," Advances in Cryptology – Proceedings of EUROCRYPT'92, Lecture Notes in Computer Science, Vol.658, pp.99-112, Springer-Verlag, 1992.

[140] M.J. Wiener, "Efficient DES key search," Technical Report TR-244, School of Computer Science, Carleton University, Canada, May 1994, Presented at the Rump Session of CRYPTO'93.

[141] G. Xiao and J.L. Massey, "A spectral characterization of crrelation-immune combining functions," IEEE Transactions on Information Theory, Vol.IT-34, No.3, pp.569-571, 1988.