COMBINED ATTACKS ON BLOCK CIPHERS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF APPLIED MATHEMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

NEŞE ÖZTOP

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CRYPTOGRAPHY

AUGUST 2009

Approval of the thesis:

COMBINED ATTACKS ON BLOCK CIPHERS

submitted by **NEŞE ÖZTOP** in partial fulfillment of the requirements for the degree of **Master of Science in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Ersan AKYILDIZ _____
Director, Graduate School of **Applied Mathematics**

Prof. Dr. Ferruh ÖZBUDAK _____
Head of Department, **Cryptography**

Assoc. Prof. Dr. Ali DOĞANAKSOY _____
Supervisor, **Department of Mathematics, METU**

Examining Committee Members:

Prof. Dr. Ferruh ÖZBUDAK _____
Department of Mathematics, METU

Assoc. Prof. Dr. Ali DOĞANAKSOY _____
Department of Mathematics, METU

Assist. Prof. Dr. Zülfükar SAYGI _____
Department of Mathematics, TOBB ETU

Dr. Muhiddin Uğuz _____
Department of Mathematics, METU

Dr. Nurdan Saran _____
Department of Computer Engineering, Çankaya University

Date: _____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: NEŞE ÖZTOP

Signature :

ABSTRACT

COMBINED ATTACKS ON BLOCK CIPHERS

ÖZTOP, Neşe
M.S., Department of Cryptography
Supervisor : Assoc. Prof. Dr. Ali DOĞANAKSOY

August 2009, 102 pages

Cryptanalytic methods are very important tools in terms of evaluating the security of block ciphers in a more accurate and reliable way. Differential and linear attacks have been the most effective cryptanalysis methods since the early 1990s. However, as the technology developed and more secure ciphers are designed, these fundamental methods started to be not so efficient. In order to analyze the ciphers, new methods should be introduced. One approach is inventing new techniques that are different from the existing ones. Another approach is extending or combining known cryptanalytic methods to analyze the cipher in a different way. This thesis is a survey of the attacks that are generated by combination of existing techniques and their applications on specific block ciphers. Mentioned attacks are namely differential-linear, differential-bilinear, higher order differential-linear, differential-nonlinear, square-nonlinear, impossible differential and boomerang type attacks.

Keywords: Block Ciphers, Combined Attacks, Differential-Linear Cryptanalysis, Impossible Differential Cryptanalysis, Boomerang Attacks

ÖZ

BLOK ŞİFRELERE YAPILAN BİRLEŞİK ATAKLAR

ÖZTOP, Neşe
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Doç. Dr. Ali DOĞANAKSOY

Ağustos 2009, 102 sayfa

Kriptanalitik metodlar blok şifrelerin güvenliğini daha doğru ve güvenilir bir şekilde değerlendirmek açısından çok önemli araçlardır. Diferansiyel ve lineer ataklar 1990'lı yılların başından beri en etkili kriptanaliz metodları olmuştur. Fakat, teknoloji geliştikçe ve daha güvenli şifreler tasarlandıkça, bu temel metodlar eskisi kadar etkili olmamaya başlamıştır. Şifreleri analiz edebilmek için yeni metodlar ileri sürülmelidir. Bir yaklaşım, varolan tekniklerden farklı teknikler ortaya koymaktır. Diğer bir yaklaşım ise bilinen metodların geliştirilmesi ya da birleştirilmesidir. Mevcut kriptanaliz tekniklerinin birleştirilmesiyle ortaya çıkan atak çeşitleri ve bunların belirli şifrelere olan uygulamaları incelenerek bu tezde sunulmuştur. Bu ataklar; diferansiyel-lineer, diferansiyel-bilineer, yüksek dereceli diferansiyel-lineer, diferansiyel-nonlineer, kare-nonlineer, olanaksız diferansiyel ve bumerang tarzı ataklardır.

Anahtar Kelimeler: Blok Şifreler, Birleşik Ataklar, Diferansiyel-Lineer Kriptanaliz, Olanaksız Diferansiyel Kriptanaliz, Bumerang Ataklar

To my family

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my supervisor Assoc.Prof.Dr. Ali DOĞANAKSOY for his invaluable guidance and inspiration. He provided encouragement, motivation and sound advice throughout this thesis period.

Very special thanks go to Onur KOÇAK for helping me get through the difficult times, for his endless support and caring he provided.

I am grateful to my collegues Aslı DARBUKA, Dilek ÇELİK and İ.Firuze KARAMAN for their close friendship, colloboration and entertainment. Every moment that we spent together was very valuable.

It is a great pleasure to thank also Fatih SULAK, Kerem VARICI, Meltem SÖNMEZ TURAN and Onur ÖZEN for assisting me in many different ways.

I would like to thank everyone with whom I have worked at Graduate School of Natural and Applied Sciences for their understanding and support during this period.

Lastly, and most importantly, I wish to thank my family whose continuous encouragement and endless love are with me throughout of my life. They supported every decision of mine and had always faith in me.

The generous financial support of the Scientific and Technological Research Council of Turkey (TUBITAK) is gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The development of information and communication technologies has led to a dramatic increase in the amount of data transmitted and in the number of people using these technologies. This progress provided people many advantages and convenience manifoldly but also brought some vulnerabilities and threats on the data transformation which causes leakage of protecting personal data and privacy. Such problems created the need for secure communications and trustworthy information infrastructure. As a science of finding solutions to these problems, cryptology has acquired significant importance in the twentieth century.

Cryptology has two main components; *cryptography* and *cryptanalysis*. Cryptography is the science of designing secure algorithms that provide confidentiality, authenticity and integrity. On the other hand, cryptanalysis is the science of analyzing and evaluating security of the cryptographic algorithms.

Confidentiality assures that information is accessible only to authorized users and prevents the data leakage to unauthorized ones. In cryptography, confidentiality is provided via encryption and decryption algorithms. There are two types of encryption algorithms: secret-key (symmetric) encryption and public-key (asymmetric) encryption. In symmetric key cryptography, the same or a related secret key is used for encryption and decryption whereas in asymmetric key cryptography, encryption is done by using a key which is publicly known, called public-key and decryption can be done by a specific user with his/her private key.

One of the most important primitives of symmetric key cryptography is block ciphers. A block cipher is a function that maps a fixed-length data block into another data block of the same length under a secret key. Formal definition of a block cipher is given as follows:

Definition 1.0.1 *An n-bit block cipher is a function* $E : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$*, such that for each* K *in the key space,* $E(P,K)$ *is an invertible mapping (encryption function). The inverse mapping denotes the decryption function.*

Many of the cryptographic primitives such as stream ciphers, hash functions, pseudo random number generators, and message authentication codes use block cipher as a building structure. Most block ciphers are constructed by repeating a round function F a certain number of times R. These block ciphers are called *iterated block ciphers*. In general, block ciphers are either of the form *Feistel Networks* or *Substitution-Permutation Networks (SPNs)*. In a Feistel Network block cipher, first the input block is split into two halves, L_0 and R_0. Then, for each round $i \in \{1, \ldots, R\}$, the round function F is applied to the right part R_{i-1} with the round subkey K_i and the result is XORed with the left part L_{i-1} and the two halves are swapped as given in the following equations

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

Data Encryption Standard (DES) [3] can be given as a traditional example to Feistel Networks.

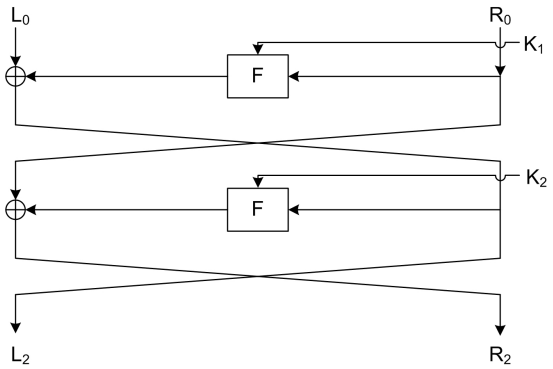

Figure 1.1: Feistel Network

Substitution Permutation Networks consist of two invertible layers: substitution and permutation. Substitution layer provides confusion in the cipher because of its nonlinear structure. On the other hand, permutation layer is a linear transformation and supplies diffusion over the cipher. Advanced Encryption Standard (AES) [31] is the most well-known block cipher of type SPN.

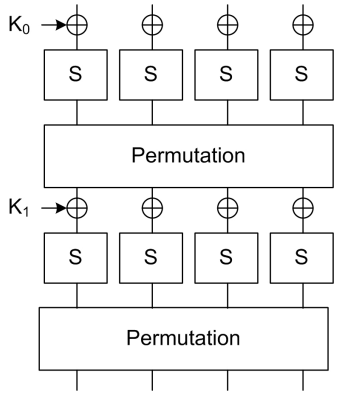

Figure 1.2: SPN

SPNs are advantageous in the sense that they are easy to analyze and implement due to their simple structure. On the other hand, the main advantage of Feistel Networks is that the round function F need not to be bijective which provides more freedom in design.

1.1 Cryptanalysis of Block Ciphers

Cryptanalysis of block ciphers is very crucial in evaluating the security and designing more secure algorithms. For this reason, in order to clarify cryptanalysis methods described in next chapters, fundamental concepts about cryptanalysis and basic techniques are mentioned in this section.

1.1.1 Attack Scenarios

Kerckhoffs' Principle [2]: *A cryptosystem should be secure even if everything about the system, except the secret key, is public knowledge.*

This principle assumes that the cryptanalyst knows all the details about the encryption algorithm except for the secret key. This means that security of the cryptosystem should depend only on the key.

Under Kerckhoffs' assumption, four widely discussed attack scenarios are given below:

- **Ciphertext-Only Attacks:** In this attack scenario, the cryptanalyst is assumed to have

knowledge on ciphertexts and too little knowledge on plaintexts. This type of attack is very difficult to apply and any cryptosystem that is vulnerable to this attack is considered to be totally insecure.

- **Known-Plaintext Attacks:** In known-plaintext attacks, the cryptanalyst is assumed to have access to ciphertexts and corresponding plaintexts.
- **Chosen Plaintext Attacks:** In this attack model, it is assumed that the attacker has the encryption box and is able to choose arbitrary plaintexts, input them to the encryption box and obtain the corresponding ciphertexts.
- **Adaptively Chosen Plaintext-Ciphertext Attacks:** In this case, the attacker is assumed to choose plaintexts, get the corresponding ciphertexts then choose other ciphertexts based on the information obtained from the previous choices.

1.1.2 Elementary Attack Techniques

One of the best measures of security for ciphers is the complexity. Complexity shows the cost of the attack in terms of some resources, for instance data, memory, time etc. and helps for allocating these resources in order to make the attack more efficient. Three most important complexity types are given in the following:

- Data Complexity: Expected number of plaintexts and/or ciphertexts required for performing the attack.
- Memory Complexity: Expected number of memory (storage) units required for the attack.
- Time Complexity: Expected number of operations required for execution of the attack. Usually in block cipher cryptanalysis, operations stand for the concerned cipher encryptions or decryptions.

The three fundamental cryptanalytic techniques [36] which can be applied to any block cipher are described as follows:

- **Dictionary Attack:** In this attack type, the attacker encrypts a plaintext with 2^k possible keys and stores the ciphertexts in a sorted dictionary. If the attacker obtains an

encrypted version of the chosen plaintext, he can find the secret key by checking for a match in the dictionary. Obviously, looking for a match in the sorted dictionary has a negligible time complexity. Also, generating the dictionary table requires 2^k encryptions, but since this precomputation is done in offline phase it has no contribution to the time complexity. Dictionary attack has a data complexity of 1 plaintext and a memory complexity of 2^k n-bit words, where n is the block size.

- **Codebook Attack:** In codebook attack, the attacker tries to construct a table (codebook) consisting 2^n ciphertexts corresponding to all 2^n possible plaintexts. This table is sorted by the ciphertexts, so when the attacker obtains a ciphertext, he searches for a match in the codebook and can find the corresponding plaintext if that key is used. Therefore, this attack requires 2^n plaintexts, 2^n n-bit words of memory and negligible time complexity.

- **Exhaustive Key Search:** In an exhaustive key search or brute force attack, given a plaintext-ciphertext pair, the attacker encrypts the plaintext by trying all 2^k possible keys and looks for a correspondence between the obtained ciphertexts and the given ciphertext. Therefore, this attack has a time complexity of 2^k encryptions, negligible data and memory complexities.

Table 1.1: Complexities of the Elementary Attack Techniques

Attack Type	Time Complexity (Encryptions)	Data Complexity (Chosen Plaintexts)	Memory Complexity (n-bit words)
Dictionary Attack	1	1	2^k
Codebook Attack	1	2^n	2^n
Exhaustive Key Search	2^k	1	1

1.1.3 Differential Cryptanalysis

Differential cryptanalysis is one of the most effective techniques in block cipher cryptanalysis. It was introduced by Biham and Shamir [4] in 1990 to break reduced-round versions of DES [3] and was extended in 1991 to break full 16-round of DES [6]. A similar method was earlier described by Murphy [17] applied on FEAL [16] block cipher.

Differential cryptanalysis analyzes how the difference between two output values C_1 and C_2 is affected when there is a specific difference between the two input values P_1 and P_2, where

C_1 and C_2 are the values after the encryption of P_1 and P_2 for r-rounds of the cipher under the same key, K. In general, the XOR operation (addition modulo 2) is associated with the notion of *difference*, but it can be defined in different ways. More formally,

Definition 1.1.1 *[40] The difference between two bit strings X and X^* is defined as*

$$\Delta X = X \otimes (X^*)^{-1},$$

where $\otimes$ is a group operation used to combine a key with the internal data X in a cipher. $(X^*)^{-1}$ is the inverse of (X^*) with respect to the $\otimes$ operator.

After the key addition under the same key K, the difference between $(X \otimes K)$ and $(X^* \otimes K)$ is

$$(X \otimes K) \otimes (X^* \otimes K)^{-1} = X \otimes K \otimes K^{-1} \otimes X^* = \Delta X.$$

Therefore, the difference between two values is independent of the key. This observation can be generalized for linear parts of the round function since the effect of the key values can be disregarded by looking at the difference between the data. On the other hand, for the nonlinear parts of the function such as S-boxes, some probabilistic observations can help to analyze the propagation of the differences. Consequently, the idea of differential cryptanalysis is to extend this property as many rounds as possible and also with high probability.

Due to the nonlinearity of S-boxes, it is not straightforward to say the output difference of the S-box for a given input difference. If there is a difference between the two values entering the S-box, this difference may produce several output differences, giving us many choices for which way we should go on. For this reason, *difference distribution table*, in other words *XOR table* of an S-box is defined as the table whose entries represent the number of pairs with the given input and output differences. More specifically, given the input difference α, the number of obtaining the output difference β is equal to the cardinality of the set

$$\{x \in \{0,1\}^m \,|S(x) \oplus S(x \oplus \alpha) = \beta\}.$$

Definition 1.1.2 *Assume that S is an $m \times n$ S-box, α is an m-bit block, β is an n-bit block, then the probability of the differential $\alpha \to \beta$ for S is defined as*

$$Pr_S(\alpha \to \beta) = Pr(S(x) \oplus S(x \oplus \alpha) = \beta), \quad x \in \{0,1\}^m$$

By definition, the probability of having $0 \rightarrow 0$ is 1 since zero input difference causes zero output difference all the time. The S-boxes having a nonzero input difference are called as *active* S-boxes.

Proposition 1.1.3 *If S is an $m \times n$ S-box, then the probability of obtaining an output difference β from S when the input difference is α:*

$$Pr_S(\alpha \rightarrow \beta) = \frac{|\{x \in \{0,1\}^m \,|S(x) \oplus S(x \oplus \alpha) = \beta\}|}{2^m}.$$

In general, the probability of a round is the product of probabilities of active S-boxes.

A one-round *differential characteristic* $\alpha \rightarrow \beta$ with probability p means that the probability of having an output difference β is p after 1-round encryption when the input difference is α. Two one-round characteristics can be concatenated provided that the output difference of the first characteristic is equal to the input difference of the second characteristic. In a similar manner, differential characteristics over multiple rounds can be constructed. Then, the probability of the concatenated differential characteristic can be computed as the product of probabilities of one-round characteristics assuming that the rounds are independent from each other.

Definition 1.1.4 *An n-round differential characteristic is a sequence of differences $(\delta_0, \delta_1, \ldots, \delta_n)$ such that the input difference $\Delta P = \delta_0$, output difference $\Delta C = \delta_n$ and each intermediate difference $\Delta I_i = \delta_i$ for $i = 1, \ldots, n-1$.*

Definition 1.1.5 *An n-round differential is a set of differential characteristics that have the same input difference ΔP and same output difference ΔC. A differential is usually represented as $\Delta P \rightarrow \Delta C$.*

Definition 1.1.6 *A right pair for an n-round differential characteristic is a pair of plaintexts (P, P^*) such that*

- $P \oplus P^* = \Delta P$,
- *for each round i, $1 \leq i \leq r$ encryption of the pair has input difference δ_{i-1} and output difference δ_i,*
- $C \oplus C^* = \Delta C$, *where (C, C^*) is the data after the encryption of (P, P^*) for n rounds.*

Any pair, which is not a right pair is called as wrong pair.

In differential cryptanalysis, the first step is to find a differential characteristic with high probability for a distinguisher. This characteristic, with enough number of chosen plaintexts, can help for distinguishing the cipher from a random permutation. Then, by adding rounds to the beginning and/or to the end of the distinguisher, guessing the corresponding subkeys and checking the differences, differential attack can be mounted. The question is what is meant by "enough number" of plaintexts. Given a differential characteristic with probability p provided that $p >> 2^{-n}$ where n is the block size, approximately $1/p$ chosen plaintext pairs are required to distinguish the cipher from a random permutation. However, wrong pairs can also give the right input and output difference without satisfying the intermediate differences which is called as *noise*. In this case, about c/p plaintext pairs should be chosen instead of $1/p$, where c depends on the *Signal to Noise Ratio*.

Definition 1.1.7 *The ratio of the probability of the right key being suggested by a right pair to the probability of a random key being suggested by a random pair with the given initial difference is called the signal to noise ratio and is denoted by S/N,*

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

where k is the number of active bits, p is the probability of the characteristic, α is the number of keys suggested by each pair of plaintexts and β is the fraction of analyzed pairs among all pairs.

In [5], it is stated that when S/N is high enough, i.e. $S/N >> 1$, small number of pairs are required for a successful attack and when $S/N \leq 1$, the number of required pairs to mount the attack becomes unreasonably high.

Overview of an $n + 1$ round differential attack:

- Find an n-round differential $(\Delta X, \Delta Y)$ with possible highest probability.
- Choose a random plaintext P and generate $P^* = P \oplus \Delta X$.
- Encrypt both plaintexts (P, P^*) under the unknown key K to obtain $C = E(P)$ and $C^* = E(P^*)$.

- Assign a counter for each possible subkey value of the last round and for each plaintext pair, increment the corresponding counter of the subkey value under which one round decryption of C and C^* gives the difference ΔY.

- Output the subkey(s) with the highest entry.

1.1.4 Linear Cryptanalysis

Linear cryptanalysis is one of the most powerful cryptanalysis techniques on block ciphers. It is a statistical, known-plaintext attack which was first used to break FEAL cipher in 1992 by Matsui and Yamagishi [8] and developed in 1993 by Matsui to present an attack on the full DES [9].

Linear cryptanalysis exploits statistical relations between linear combinations of plaintext, ciphertext and subkey bits. In order to obtain such a relation, linear cryptanalysis approximates the nonlinear part of the cipher to a linear equation with some probability. A linear approximation is of the form

$$P[i_1, i_2, \ldots, i_a] \oplus C[j_1, j_2, \ldots, j_b] = K[k_1, k_2, \ldots, k_c] \tag{1.1}$$

where $i_1, i_2, \ldots, i_a, j_1, j_2, \ldots, j_b, k_1, k_2, \ldots, k_c$ denote fixed bit locations. For a random cipher, the probability that Equation 1.1 holds is around 1/2. Therefore, if for a cipher, the above equation holds with probability p higher or lower than 1/2, then it can be deduced that the cipher does not have randomness properties. The measure of being far from a probability of 1/2 is called the *linear probability bias* and is shown as $q = \left|p - \frac{1}{2}\right|$. In linear cryptanalysis, the approach is to find linear approximations with sufficiently large bias because the attack will be more efficient and applicable with fewer known plaintexts.

General approach in linear cryptanalysis is to start with approximations to nonlinear components in the round function in order to obtain linear relations for the overall cipher. In a block cipher, it is very important to approximate S-boxes with a linear expression since they are designed to be highly nonlinear. For this reason, Matsui defined the following measure:

Definition 1.1.8 *For a given S-box $S : \mathbb{Z}_2^n \to \mathbb{Z}_2^m$ and fixed bit masks (α, β) such that $\alpha \in \mathbb{Z}_2^n$ and $\beta \in \mathbb{Z}_2^m$, let NS(α,β) be the number of inputs $x \in \mathbb{Z}_2^n$ for which parity under α matches the*

parity of S(x) under β:

$$NS(\alpha,\beta) = \left|\left\{x \in \mathbb{Z}_2^n \mid x \cdot \alpha = S(x) \cdot \beta\right\}\right|$$

where "·" denotes a bitwise AND operation.

A complete list of all linear approximations of an S-box is represented in a table called *Linear Approximation Table (LAT).*

One-round linear approximations can be concatenated into linear approximations for multiple rounds similar to the case in differential characteristics. This is possible when the output mask of the first linear expression is equal to the input mask of the second. In order to illustrate, let the first approximation be $\lambda_P \cdot P \oplus \lambda_T \cdot T = \lambda_{K_1} \cdot K_1$ with bias q_1 and the second approximation be $\lambda_T \cdot T \oplus \lambda_C \cdot C = \lambda_{K_2} \cdot K_2$ with bias q_2, where λ's are bit masks. Then, we have

$$\lambda_P \cdot P \oplus \lambda_C \cdot C = \lambda_{K_1} \cdot K_1 \oplus \lambda_{K_2} \cdot K_2 \tag{1.2}$$

when one of the two following equations holds:

- $\lambda_P \cdot P \oplus \lambda_T \cdot T = \lambda_{K_1} \cdot K_1$ and $\lambda_T \cdot T \oplus \lambda_C \cdot C = \lambda_{K_2} \cdot K_2$
- $\lambda_P \cdot P \oplus \lambda_T \cdot T \neq \lambda_{K_1} \cdot K_1$ and $\lambda_T \cdot T \oplus \lambda_C \cdot C \neq \lambda_{K_2} \cdot K_2$

Note that, $\lambda_{K_1} \cdot K_1$ and $\lambda_{K_2} \cdot K_2$ are unknown but fixed values, either 0 or 1. So, without loss of generality we may assume that they are 0. Now, the probability that Equation 1.2 holds is the sum of the probabilities of the above two events, assuming they are independent. The probability of the first event is $(\frac{1}{2} + q_1)(\frac{1}{2} + q_2)$ and the probability of the second is $(\frac{1}{2} - q_1)(\frac{1}{2} - q_2)$. Therefore, the probability that Equation 1.2 holds is $\frac{1}{2} + 2q_1q_2$. Matsui generalized this case as can be seen in the following lemma:

Piling-Up Lemma [9]: *For n independent, binary random variables $X_1,\ldots,X_n$ with biases $q_i = \left|p_i - \frac{1}{2}\right|$, $1 \leq i \leq n$, the probability that $X_1 \oplus X_2 \oplus \ldots \oplus X_n = 0$ is*

$$p = \frac{1}{2} + 2^{n-1} \prod_{i=1}^{n} (p_i - \frac{1}{2})$$

or, the bias is

$$q = 2^{n-1} \prod_{i=1}^{n} q_i.$$

Once having found an n-round linear approximation with bias q, an $n+1$ round linear attack can be described basically as follows:

Assume n-round linear approximation is

$$P[i_1, i_2, \ldots, i_a] \oplus C[j_1, j_2, \ldots, j_b] \oplus F(C_L, K_n)[\ell_1, \ell_2, \ldots, \ell_d] = K[k_1, k_2, \ldots, k_c] \quad (1.3)$$

- Take about $N = c.q^{-2}$ plaintexts, P.
- Encrypt these plaintexts and obtain the corresponding ciphertexts, C.
- Partially decrypt these ciphertexts for one round by guessing relevant bits of the last round's subkey, K_n.
- For each candidate key, count the number T_i of plaintext-ciphertext pairs such that the left side of Equation 1.3 is zero.
- According to the Matsui's algorithm given in [9],
  - If $|T_{max} - N/2| > |T_{min} - N/2|$, then adopt the key candidate corresponding to T_{max} and guess $K[k_1, k_2, \ldots, k_c] = 0$ (when $p > 1/2$) or 1 (when $p < 1/2$).
  - If $|T_{max} - N/2| < |T_{min} - N/2|$, then adopt the key candidate corresponding to T_{min} and guess $K[k_1, k_2, \ldots, k_c] = 1$ (when $p > 1/2$) or 0 (when $p < 1/2$).

In linear cryptanalysis, required number of known plaintexts to mount the attack is $N = c.q^{-2}$. Success rate of the linear attack changes with respect to the number of known plaintexts. Table 1.2 shows the success rate with different c values:

Table 1.2: Success Rate According to Different N Values

N	$2q^{-2}$	$4q^{-2}$	$8q^{-2}$	$16q^{-2}$
Success Rate	48.6%	78.5%	96.7%	99.9%

1.2 Structure of the Thesis

This thesis aims to make a survey on combined attacks and give examples of these attacks applied on specific block ciphers in order to clarify the attack methods. Actually, this thesis is a part of the survey that is planned to be made on block cipher cryptanalysis. First part

of the survey is going to cover differential cryptanalysis, linear cryptanalysis and square attacks [45]. Second part is studied in this thesis. The third part covers related-key attacks [44]. Organization of the thesis is as follows. First, a brief introduction is given to cryptanalysis of block ciphers and its fundamentals: differential and linear cryptanalysis. Other chapters explain cryptanalysis methods which were developed by combination of differential and/or linear attacks. The attacks called differential-linear, differential-bilinear, higher order differential-linear, differential-nonlinear and square-nonlinear are described in Chapter 2 and each of these attacks is a combination of differential type distinguisher and a linear, bilinear or nonlinear approximation. Cryptanalysis methods, namely impossible differential and boomerang type attacks are a combination of two (or more) differential type distinguishers and are mentioned in Chapter 3 and Chapter 4, respectively. Finally, Chapter 5 concludes the thesis.

CHAPTER 2

DIFFERENTIAL-LINEAR CRYPTANALYSIS

Differential-linear cryptanalysis was first introduced by Langford and Hellman [11] in 1994. They combined both differential and linear cryptanalysis techniques in which the differential characteristic creates a linear approximation with probability 1.

Differential-linear cryptanalysis method is generally applied to block ciphers which are resistant to differential and linear attacks. Finding long differential characteristics or linear approximations with high probability may be difficult or impossible for a cipher. However, one short differential with high probability can be concatenated to a linear approximation and a high probability differential-linear distinguisher can be constructed. This idea was improved by Biham et al. [12] to enhanced differential-linear cryptanalysis which is described in Section 2.2. Furthermore, extensions of differential and linear cryptanalysis such as higher order differential, bilinear and nonlinear cryptanalysis can be combined to apply new attack techniques. Remaining sections are devoted to differential-bilinear cryptanalysis, higher order differential-linear cryptanalysis, differential-nonlinear and square-nonlinear cryptanalysis.

Proposition 2.0.1 *Let the block cipher* $\boldsymbol{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ *be a cascade of two sub-ciphers* $\boldsymbol{E} = \boldsymbol{E}_1 \circ \boldsymbol{E}_0$. *Assume that there exists a truncated differential* $\Omega_P \rightarrow \Omega_T$ *with probability* 1 *for* E_0^K *and a linear approximation* $\lambda_T \rightarrow \lambda_C$ *with bias* q *for* E_1^K. *If the plaintext pair,* (P, P^*)*, satisfies the difference, i.e.* $P \oplus P^* = \Omega_P$*, then*

$$Pr(\lambda_C \cdot C \oplus \lambda_C \cdot C^* = \Omega_T \cdot \lambda_T) = \frac{1}{2} + 2q^2$$

where $C = E_1^K(T)$ *and* $C^* = E_1^K(T^*)$.

Proof. Assume that there exists a truncated differential $\Omega_P \rightarrow \Omega_T$ with probability 1 for E_0^K

and a linear approximation $\lambda_T \rightarrow \lambda_C$ with bias q for E_1^K. Let (P, P^*) be the plaintext pair such that $P \oplus P^* = \Omega_P$ and $T = E_0^K(P)$, $T^* = E_0^K(P^*)$. Since the differential characteristic is satisfied with probability 1, $T \oplus T^* = \Omega_T$.

For the linear approximation, it is expected that

$$\lambda_T \cdot T \oplus \lambda_C \cdot C = 0 \text{ with bias } q, \tag{2.1}$$

$$\lambda_T \cdot T^* \oplus \lambda_C \cdot C^* = 0 \text{ with bias } q. \tag{2.2}$$

Therefore,

$$\lambda_C \cdot C \oplus \lambda_C \cdot C^* = \lambda_T \cdot (T \oplus T^*) = \lambda_T \cdot \Omega_T$$

with a probability of $1 \cdot \left[(\frac{1}{2} + q) \cdot (\frac{1}{2} + q) + (\frac{1}{2} - q) \cdot (\frac{1}{2} - q)\right] = \frac{1}{2} + 2q^2$, since

- Equations 2.1 and 2.2 are both satisfied with bias q, i.e. with probability $\frac{1}{2} + q$, or
- Equations 2.1 and 2.2 are both not satisfied with probability $\frac{1}{2} - q$.

Consequently, a differential-linear attack with probability $\frac{1}{2} + 2q^2$ requires $O(q^{-4})$ chosen plaintext pairs.

2.1 Differential-Linear Cryptanalysis of DES

Cryptanalysis of DES using differential-linear technique was given in [11]. In this analysis, a 3-round differential characteristic and a 3-round linear approximation of DES are combined to mount an 8-round attack. This 8-round differential-linear attack recovers 10 bits of the key with 512 chosen plaintexts. However, the best 8-round differential and 8-round linear attacks require over 5000 chosen plaintexts and 500.000 known plaintexts, respectively. The purpose of applying this technique is to reduce the amount of required texts in the attack.

2.1.1 Description of DES

DES [3] is a well-known block cipher and has been widely used for many years. In 1973, National Bureau of Standards (NBS) which is currently named as National Institute of Standards

and Technology (NIST), announced the first request for a standard encryption algorithm for the United States. DES was developed based on the block cipher Lucifer and submitted as a candidate algorithm by IBM. The year 1975 is the time that DES was first published and two years later, in 1977, this encryption algorithm was approved as an official Federal Information Processing Standard (FIPS), FIPS PUB 46. DES is much more efficient in hardware than in software. Therefore, DES was subjected to many cryptanalytic attacks. In 1990s, some replacement algorithms were started to be shown up. Algorithms such as Triple DES (which uses DES three times) and DES-X are used in most cases instead of DES.

DES is a Feistel Network block cipher and consists of 16 rounds. It has a block size of length 64-bit and uses a 56-bit key. The encryption procedure roughly is as follows: First, an initial permutation called IP is applied to the plaintext block. Then, the input block is divided into two halves, namely (L_0, R_0), where L_0 and R_0 are 32-bit words. After that, R_0 and the round subkey enter the round function, F, which will be explained below. Output of the F function is XORed with L_0 and the two halves are swapped. These operations are processed in each of the 16 rounds. Finally, after 16 rounds, the final permutation, FP which is the inverse of IP actually, is applied to the output block. An illustration of DES is given in Figure 2.1. The encryption algorithm is the following:

Let $(L_{i-1},\ R_{i-1})$ denote the input to the i^{th} round, $(L_i,\ R_i)$ denote the i^{th} round output and K_i denote the subkey used in i^{th} round. Then, for $i = 1, \ldots, 15$

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

Round Function F:

DES round function F takes two inputs: right half of the data which is 32 bits and the 48-bit round subkey. In F, first, the 32-bit data is expanded to 48 bits through the expansion function E. Then, the expanded data and the subkey are XORed. Result of the XOR operation is divided into eight parts and becomes the input to the S-boxes. DES round function has 8 different 6×4 bit S-boxes, namely $S_1, \ldots, S_8$, each of which are essentially nonlinear table look ups and satisfy confusion in the cipher. Output of the S-boxes which is in fact 32 bits is permuted according to the permutation table P and the resulting value becomes the output of the F-function. Schematic description of F is given in Figure 2.2. Expansion and permutation

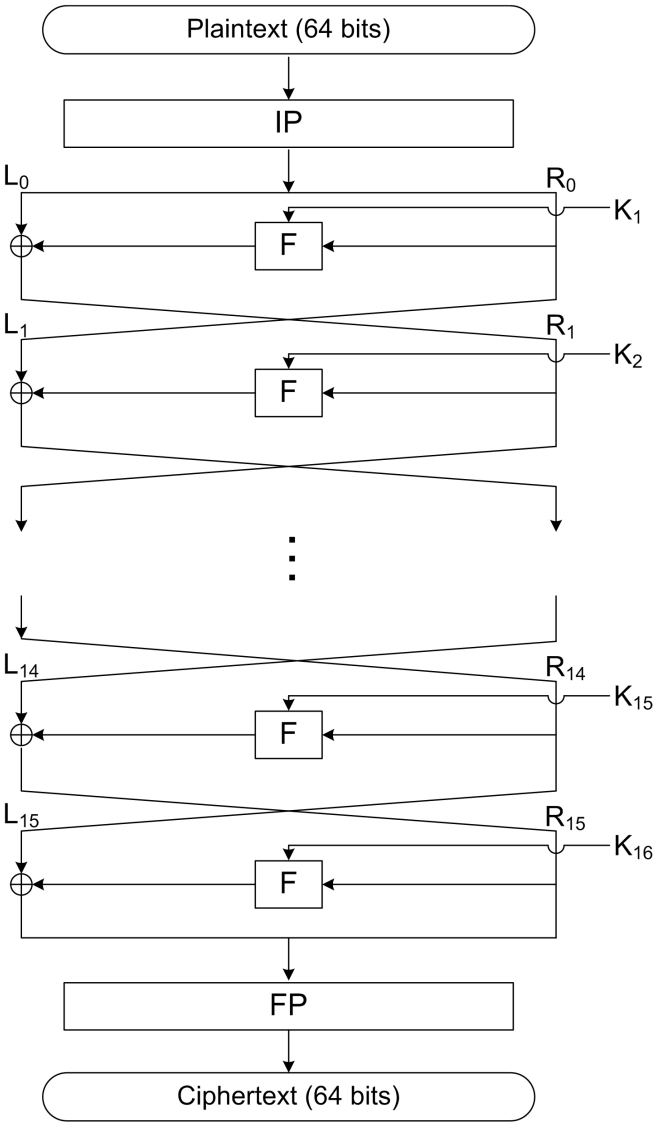

Figure 2.1: Structure of DES

tables are presented in Table 2.1 and Table 2.2, respectively.

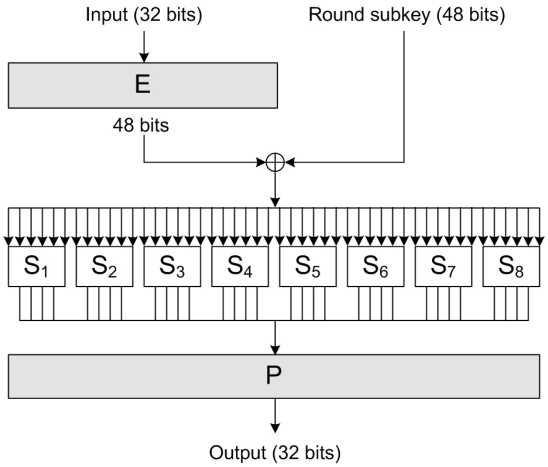

Figure 2.2: Round Function F of DES

Table 2.1: Expansion Table E

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Table 2.2: Permutation Table P

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	5

If $X = \{x_1, x_2, \ldots, x_{32}\}$, then $E(X) = \{x_{32}, x_1, x_2, \ldots, x_{32}, x_1\}$.

If $Y = \{y_1, y_2, \ldots, y_{32}\}$, then $P(Y) = \{y_{16}, y_7, y_{20}, \ldots, y_4, y_5\}$.

Description of the key scheduling algorithm was omitted since we will not deal with the details of the key schedule in this chapter. Also, initial and final permutation tables were skipped because of the same reason. Full description of the cipher can be found in [3].

Notation:

In this section, bits of the data blocks are numbered from left to right by taking the leftmost bit as the 1^{st} bit. Similarly, the input to S-box is taken as $(x_1, x_2, x_3, x_4, x_5, x_6)$. Furthermore, initial and final permutation are not taken into consideration since they have no cryptanalytic importance. Hence, (L_0, R_0) will denote the 64-bit block as plaintext and (L_r, R_r) will denote

the ciphertext block after r rounds. $X[i]$ is referred to i^{th} bit of data block X and $X[i, j, \ldots, k] = X[i] \oplus X[j] \oplus \ldots \oplus X[k]$.

2.1.2 6-Round Differential-Linear Distinguisher

This 6-round distinguisher was composed of a 3-round truncated differential characteristic concatenated with a 3-round linear approximation of DES.

3-Round Linear Approximation:

In the differential-linear attack [11], the best 3-round linear approximation of DES, which was found by Matsui [9] and has a probability of 0.695 or a bias of 0.195, is used. 3-round linear characteristic is depicted in Figure 2.3.

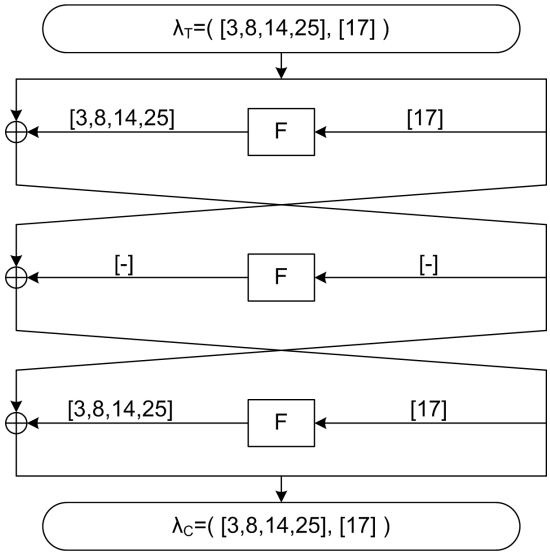

Figure 2.3: 3-Round Linear Characteristic of DES

3-Round Differential Characteristic:

Once having found the 3-round linear approximation with highest probability, 3-round differential path with probability 1 which was shown in Figure 2.4, was constructed in such a way that output bits of the truncated differential satisfy the input parity relation of the linear approximation:

1. In order to satisfy input parity relation of the linear characteristic, output bits ($L_4[3, 8, 14, 25]$, $R_4[17]$) and ($L_4{}^*[3, 8, 14, 25]$, $R_4{}^*[17]$) must be unchanged.

2. Since $L'_3 = R_4$, $L_3[17]$ and $L_3{}^*[17]$ must be unchanged. Bit [17] is output of S_1, then which input bits to the S-boxes must change so that the output bit [17] remains unchanged? The answer is, if the input bits 9,17,23 and 31 change (these bits are output of S_1), output of S_2, S_3, S_4, S_5, S_6 and S_8 change since bit [9] is the input of S_2 and S_3 because of the expansion, bit [17] is the input of S_4 and S_5, bit [23] is the input of S_6 and bit [31] is the input of S_8. So, only S_1 and S_7's output don't change.

3. $L'_2 = R_3$, then L'_2={only S_1's output change}. In order to have only S_1's output change, then only S_1's input must be changed. [32,1,2,3,4,5] are the input bits to S_1 but [32, 1], [4, 5] are also input bits to S_8 and S_2 respectively. This means that, if [32,1,4,5] change, then outputs of S_8 and S_2 change also. Hence, only bits 2 and/or 3 must be changed in order to have only S_1's output change.

4. Finally, input difference is Ω_P = (only bits 2 and/or 3 change, no change), output difference is Ω_T = ([17], [3, 8, 14, 25] don't change) and $\Omega_P \rightarrow \Omega_T$ with probability 1.

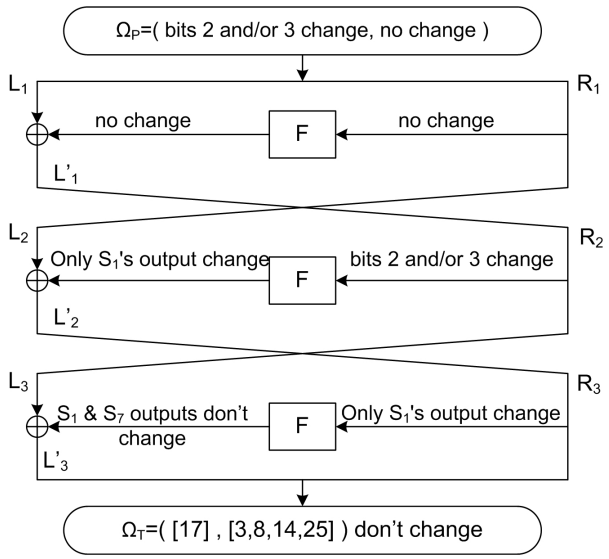

Figure 2.4: 3-Round Differential Characteristic of DES

Consequently, 6-round differential-linear distinguisher with probability $p = \frac{1}{2} + 2q^2 = \frac{1}{2} + (0.195)^2 = 0.576$ or bias $q' = 0.076$ is obtained.

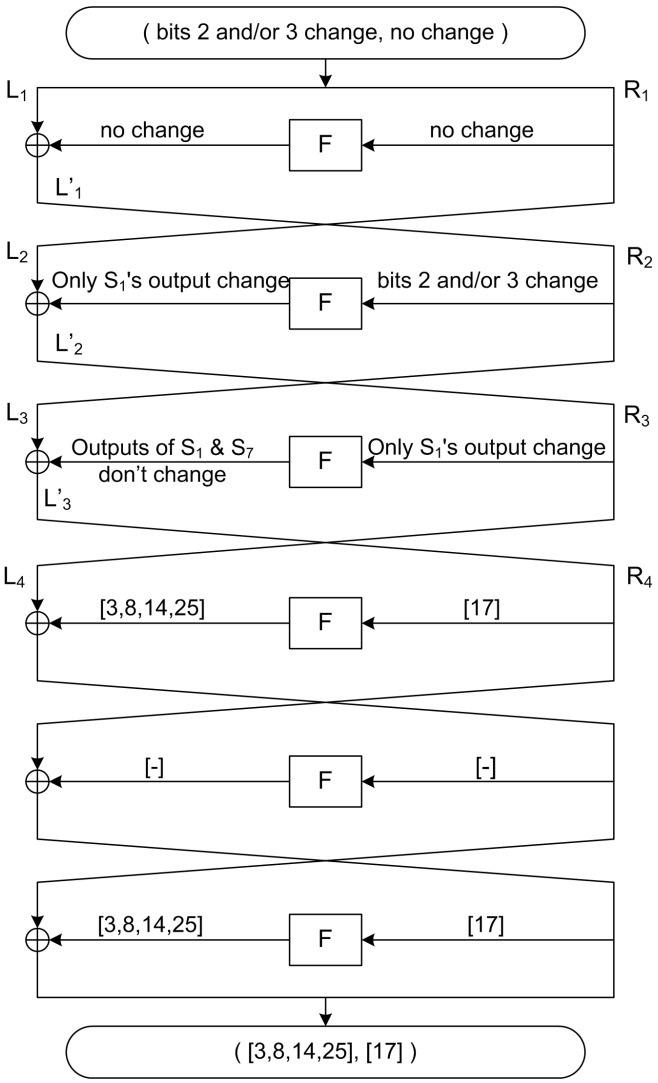

Figure 2.5: 6-Round Differential-Linear Distinguisher of DES

2.1.3 8-Round Differential-Linear Attack

In this attack, a 6-round differential-linear distinguisher was constructed first. Then, one round to the beginning and one round to the end of the 6-round distinguisher was added to apply the 8-round attack.

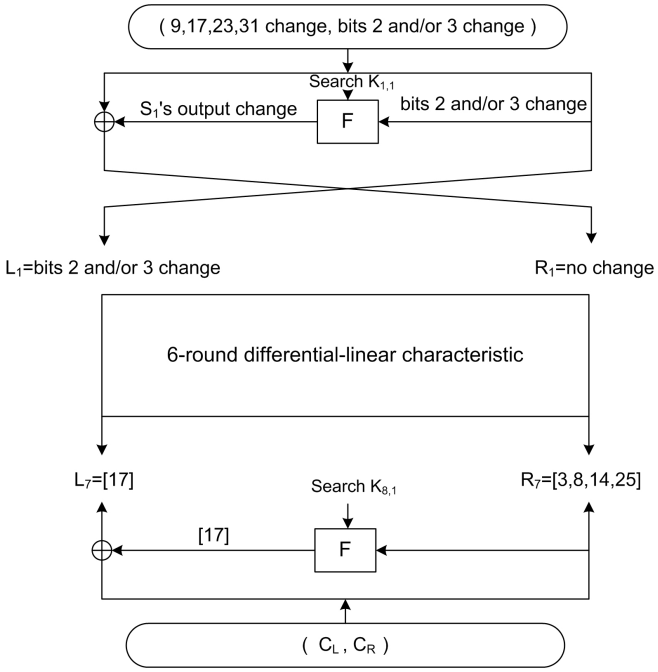

Figure 2.6: 8-Round Differential-Linear Attack on DES

Attack Procedure:

1. Choose any plaintext $P = (P_L, P_R)$.

2. Obtain plaintexts $P_1, \ldots, P_{63}$ by varying bits 9,17,23,31 of P_L and bits 2 and 3 of P_R.

3. Ask for the encryption of these plaintexts and get the corresponding ciphertexts.

4. *Construction of the pairs:* 96 different plaintext pairs can be constructed using the above 64 plaintexts so that after the first round, the difference between the right part of the pairs will be zero and the difference between the left part of the pairs will be in 2^{nd} bit or 3^{rd} bit or in both bits. (See Figure 2.6) This process is explained as follows:

   There are 16 different values (varying bits 9,17,23,31) for the left part P_L. Let L^i, $i \in \{1, \ldots, 16\}$ denotes these 16 different values. For the right part P_R, there are 4

different values R^j, $j \in \{1, \ldots, 4\}$. Let R^1 denotes the reference plaintext, R^2, R^3 and R^4 denote the plaintexts obtained by changing second bit of R^1, third bit of R^1 and the second and the third bits of R^1, respectively. Now, the plaintext pairs

- (L,R^0) and (L,R^1), - (L,R^0) and (L,R^2), - (L,R^0) and (L,R^3)

- (L,R^1) and (L,R^2), - (L,R^1) and (L,R^3), - (L,R^2) and (L,R^3)

have the desired difference after the first round. So, for each L, 6 pairs can be obtained which gives a total of 96 pairs for 16 different values of L.

5. In the first round, bits 2 and 3 are the inputs of S_1. Let $K_{1,1}$ be the part of the subkey K_1 corresponding to S_1. If $K_{1,1}$ is known, then the left half of P^*, namely ${P_L}^*$, can be find by partially encrypting plaintexts P and P^* for one round and checking for the equation:
$$[P_L \oplus F(P_R, K_{1,1})] \oplus [{P_L}^* \oplus F({P_R}^*, K_{1,1})] = 0 \tag{2.3}$$

6. For each guess of $K_{1,1}$,
   (a) Partially encrypt each plaintext for one round and obtain ${P_L}^*$ from Equation 2.3.
   (b) Guess the value of $K_{8,1}$, decrypt the ciphertext pairs by one round.
   (c) For each possible value of $K_{8,1}$, assign a counter and increment that counter by 1 when the following equation is satisfied:
$$[F(C_R, K_{8,1}) \oplus C_L] \cdot [17] = [F({C_R}^*, K_{8,1}) \oplus {C_L}^*] \cdot [17]$$

Probability that the 6-round differential-linear characteristic holds is $p = \frac{1}{2}+0.076$ and the bias is $q' = 0.076$. Number of required chosen plaintext pairs to mount the differential-linear attack is approximately $c \cdot q^{-2}$ where c is a constant which affects the success rate of the attack and q is the bias of the differential-linear characteristic. Hence, for $c = 8$ and $q' = 0.076$, number of required plaintext pairs can be computed approximately as 1384. From each structure of 64 chosen plaintexts, 96 pairs can be obtained. So, for 1384 pairs, $\frac{1384 \cdot 64}{96} \approx 900$ chosen plaintexts are needed. Langford et al. [11] stated that success rate of this attack is 80% with 512 chosen plaintexts and 95% with 768 chosen plaintexts.

Time complexity of the attack with 512 plaintexts is about $2^9 \cdot 2^{10} \cdot \frac{2}{8} \cdot \frac{1}{8} = 2^{14}$ 8-round encryptions of DES. There are 12 subkey bits to be guessed but 2 of the bits are common in

$K_{1,1}$ and $K_{8,1}$ which makes a total of 2^{10} subkey values. 2/8 comes from 2 S-box computations over 8 S-boxes and 1/8 comes from 1 round partial encryption over 8 rounds. Also, time complexity of the attack with 768 chosen plaintexts is about $2^{14.6}$ 8-round DES encryptions.

2.2 Enhanced Differential-Linear Cryptanalysis

This technique was presented by Biham et al. [12] and is an enhancement of differential-linear cryptanalysis explained in previous section. Similar to differential-linear cryptanalysis given in [11], the enhanced version also combines a differential characteristic with a linear approximation. The reason why this method called as enhanced differential-linear cryptanalysis is that the differential characteristic induces a linear approximation with probability less than 1 different from ordinary differential-linear cryptanalysis [11] in which the probability of the differential is equal to 1.

Proposition 2.2.1 *Let the block cipher* $\boldsymbol{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ *be a cascade of two sub-ciphers* $\boldsymbol{E} = \boldsymbol{E}_1 \circ \boldsymbol{E}_0$. *Assume that there exists a truncated differential* $\Omega_P \rightarrow \Omega_T$ *with probability p for* E_0^K *and a linear approximation* $\lambda_T \rightarrow \lambda_C$ *with bias q for* E_1^K. *If two plaintexts* (P, P^*) *satisfies the difference, i.e.* $P \oplus P^* = \Omega_P$, *then*
$Pr(\lambda_C \cdot C \oplus \lambda_C \cdot C^* = \Omega_T \cdot \lambda_T) = \frac{1}{2} + 2pq^2$ *where* $C = E_1^K(T)$ *and* $C^* = E_1^K(T^*)$.

Proof. Proof is similar to that of Proposition 2.0.1. Assume that the differential $\Omega_P \rightarrow \Omega_T$ with probability $p < 1$. Hence, a pair (P, P^*) with input difference $P \oplus P^* = \Omega_P$ has an output difference $T \oplus T^* = \Omega_T$ with probability p. Also, assume that $\lambda_T \rightarrow \lambda_C$ with a bias q.

- If the differential characteristic is satisfied for a pair with probability p, then the difference $\lambda_T \cdot T \oplus \lambda_T \cdot T^*$ is known and equals to $\lambda_T \cdot \Omega_T$. For such pairs, $\lambda_C \cdot C \oplus \lambda_C \cdot C^* = \lambda_T \cdot \Omega_T$ holds with probability $(\frac{1}{2} + q) \cdot (\frac{1}{2} + q) + (\frac{1}{2} - q) \cdot (\frac{1}{2} - q) = \frac{1}{2} + 2q^2$,
- If the differential is not satisfied with probability $1 - p$, then behavior of the value $\lambda_C \cdot C \oplus \lambda_C \cdot C^*$ is assumed as random, i.e. $\lambda_C \cdot C \oplus \lambda_C \cdot C^* = \lambda_T \cdot \Omega_T$ holds with probability $(1 - p) \cdot \frac{1}{2}$.

Therefore, $\lambda_C \cdot C \oplus \lambda_C \cdot C^*$ is biased towards the value $\lambda_T \cdot \Omega_T$ with probability

$$p \cdot \left[(\frac{1}{2} + q) \cdot (\frac{1}{2} + q) + (\frac{1}{2} - q) \cdot (\frac{1}{2} - q) \right] + (1 - p) \cdot \frac{1}{2} = \frac{1}{2} + 2pq^2.$$

The attack requires $O(p^{-2}q^{-4})$ chosen plaintext pairs.

2.2.1 7-Round Differential-Linear Distinguisher of DES

7-round distinguisher is constructed by combining a 4-round differential characteristic with probability $p = \frac{14}{64}$ and a 3-round linear approximation with a bias $q = 0.195$. 4-round differential characteristic is obtained by adding one round to the beginning of the 3-round differential characteristic explained in Section 2.1. An illustration of the 4-round characteristic is given in Figure 2.7.

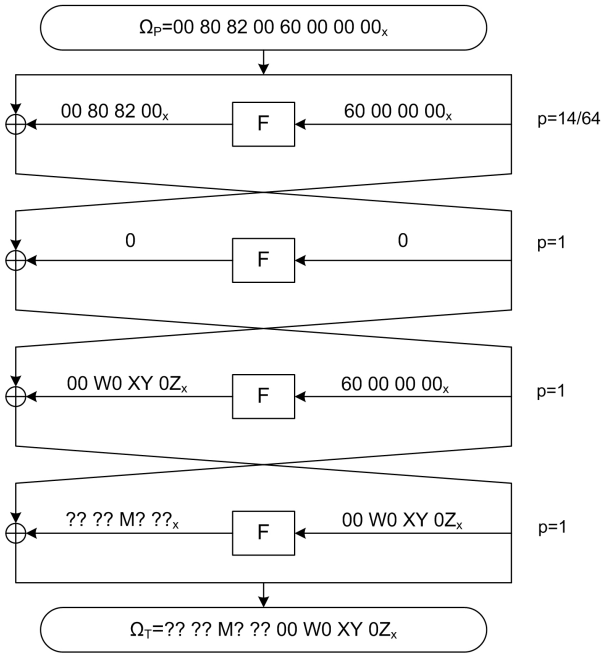

Figure 2.7: 4-Round Differential Characteristic

For the linear approximation, the same 3-round linear approximation given in Section 2.1 is used. Total probability of the 7-round differential-linear distinguisher can be computed as:

$$\frac{1}{2} + 2pq^2 = \frac{1}{2} + 2 \cdot \frac{14}{64} \cdot (0.195)^2 = 0.5167$$

2.2.2 8-Round Differential-Linear Attack on DES

This 8-round attack can be applied to DES by adding one round to the end of the 7-round distinguisher as shown in Figure 2.8.

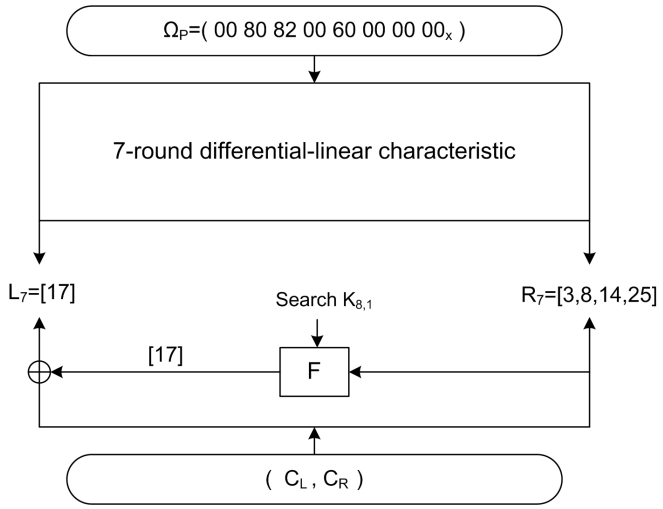

Figure 2.8: 8-Round Enhanced Differential-Linear Attack

Attack Procedure:

1. Take $N = 2^{13.81}$ plaintext pairs (P, P^*) such that $P \oplus P^* = \Omega_P$ = 00 80 82 00 60 00 00 00.
2. Encrypt these plaintext pairs (under the unknown key K) and obtain their corresponding ciphertext pairs (C, C^*).
3. Initialize an array of 2^6 counters to zeroes for the subkey guess of $K_{8,1}$.
4. For each ciphertext pair (C, C^*),
   (a) Try all 2^6 possible values of $K_{8,1}$.
   (b) For each value of $K_{8,1}$, compute $F(K_{8,1}, C_R) \cdot [17]$ and $F(K_{8,1}, C_R{}^*) \cdot [17]$.
   (c) Check whether
   $$[F(K_{8,1}, C_R) \oplus C_L] \cdot [17] = [F(K_{8,1}, C_R{}^*) \oplus C_L{}^*] \cdot [17]$$
   If they are equal, increment the corresponding counter for $K_{8,1}$.
5. The counter with highest entry will give the correct value of $K_{8,1}$.

The probability of 8-round differential-linear distinguisher is $\frac{1}{2} + 2pq^2 = 0.5167$ as computed above. Therefore, required number of chosen plaintext pairs is $O(p^{-2}q^{-4})$, where $p = 14/64$ and $q = 0.195$ in this attack. Then, N will be approximately $(14/64)^{-2} \cdot (0.195)^{-4} \approx 2^{13.81}$. Authors of [12] state that for $N = 2^{13.81}$, this attack succeeds with probability 77.27% or more.

Time complexity of the attack is $2^{14.81} \cdot 2^6 \cdot \frac{1}{8} \cdot \frac{1}{8} = 2^{14.81}$ 8-round encryptions of DES since there are $2^{14.81}$ chosen plaintexts, 2^6 subkeys tried. Each trial of subkey is one round partial encryption (1/8 comes from here since there are 8 rounds) and it takes one S-box computation (other 1/8 comes from here since there are 8 S-boxes in a round).

Data complexity of the attack is $2^{13.81} \cdot 2 = 2^{14.81}$ chosen plaintexts.

2.2.3 9-Round Differential-Linear Attack on DES

The 8-round attack explained in the previous section can be extended to 9 rounds by adding one more round to the beginning. In this attack, again a 7-round differential-linear distinguisher was used. However, first round of the differential characteristic was slightly modified in order to reduce the number of active S-boxes so that less number of subkey bits will be guessed. Hence, a differential characteristic with probability 12/64 was used instead of the one with probability 14/64. The modified 4-round differential characteristic is depicted in Figure 2.9.

The probability of this 7-round differential-linear characteristic is :

$$\frac{1}{2} + 2pq^2 = \frac{1}{2} + 2 \cdot 0.1875 \cdot (0.195)^2 = 0.5143$$

Attack Procedure:

1. Take $N = 2^{16}$ plaintexts, consisting of 2^7 structures, chosen by selecting
   (a) Any plaintext P_0,
   (b) The plaintexts $P_1,\ldots,P_{255}$ from P_0 by varying eight bits (output of S_6 and S_8 in round 1) masked by 18 22 28 28 00 00 00 00,
   (c) The plaintexts $P_{256}, \ldots, P_{511}$ from $P_i = P_{i-256} \oplus 40\ 00\ 00\ 00\ 00\ 00\ 02\ 02$, for $i = 256, \ldots, 511$.

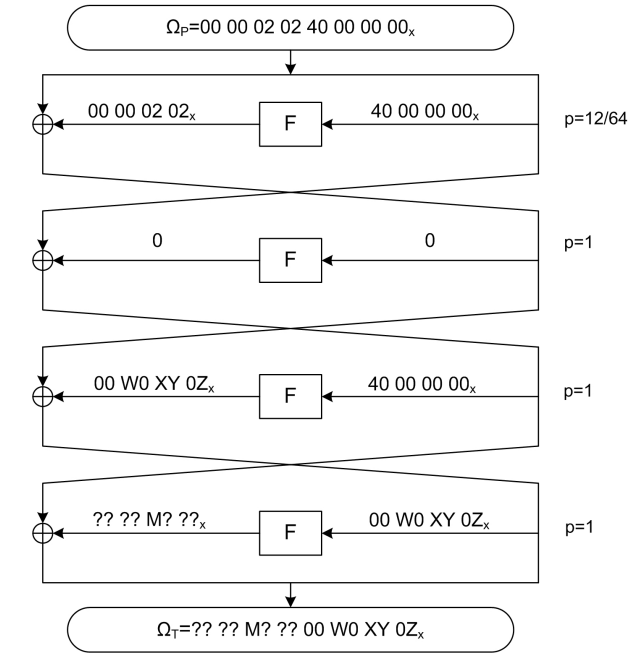

Figure 2.9: The Modified 4-Round Differential Characteristic

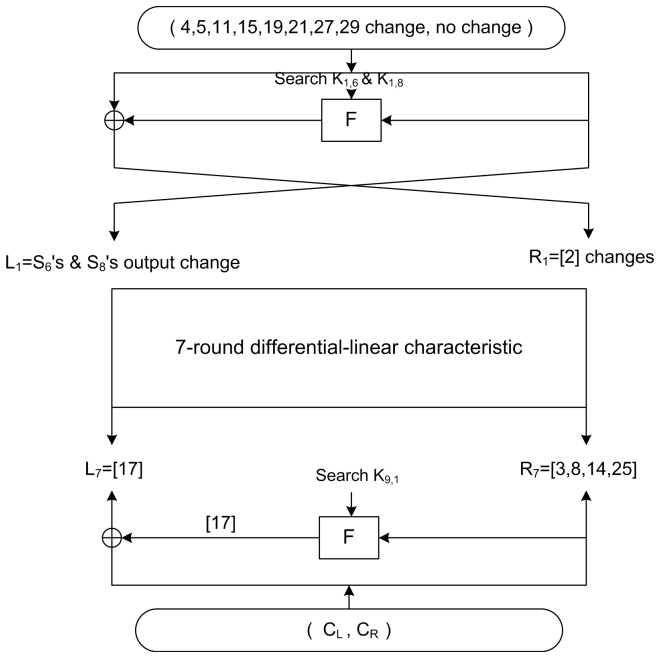

Figure 2.10: 9-Round Enhanced Differential-Linear Attack

2. Encrypt these plaintext pairs (under the unknown key K) and obtain their corresponding ciphertext pairs.

3. For each value of the 12 bits of $K_{1,6}$ and $K_{1,8}$,

   (a) Partially encrypt all plaintexts in the first round and find the pairs that have difference Ω_P = 00 00 02 02 40 00 00 00 before round 2.

   (b) Apply the 8-round attack for the pairs satisfying the previous step, i.e.,

   For each ciphertext pair,

      i. Guess all 2^6 values of $K_{9,1}$,

      ii. For each value of the subkey, compute output subset parity. If the parities are equal, increment the corresponding counter of $K_{9,1}$.

4. Output the subkey guess corresponding to the highest counter of $K_{9,1}$ along with the guess of $K_{1,6}$ and $K_{1,8}$.

Time complexity of the attack is $2^{12} \cdot (2^{16} \cdot \frac{2}{8} \cdot \frac{1}{9} + 2^{16} \cdot \frac{1}{8} \cdot \frac{1}{9}) \approx 2^{28}$ 9-round encryptions of DES. Again, 1/8 comes from 1 S-box computation over 8 S-boxes and 1/9 comes from 1 round over 9 rounds.

Data complexity is 2^{16} chosen plaintexts.

2.3 Differential-Bilinear Cryptanalysis

Bilinear Cryptanalysis

A bilinear map is a function of two variables that is linear with respect to each of its variables. Bilinear cryptanalysis [14] analyzes approximations involving bilinear functions of plaintext, ciphertext and the key bits. It is a generalization of linear cryptanalysis. It is worth noting that a bilinear approximation may also include linear terms. Bilinear attacks were aimed especially at ciphers of Feistel network, because finding bilinear approximations for Feistel ciphers is easier than for others.

Bilinear expressions in a Feistel cipher:

Let $(L_r[1,\ldots,n],\ R_r[1,\ldots,n])$ be the input value of the r^{th} round in a Feistel cipher where

L and R denote the left and the right half of the data, respectively. Also, let $I_r[1,\ldots,n]$ and $O_r[1,\ldots,n]$ be the input and output values of F-function in the r^{th} round. Then, the general equations for Feistel ciphers are the following:

$$I_r = R_r \tag{2.4}$$

$$L_{r+1} = R_r \tag{2.5}$$

$$R_{r+1} = L_r \oplus O_r. \tag{2.6}$$

Combining Equations 2.4, 2.5 and 2.6, 1-round bilinear approximation can be written as in Equation 2.7.

$$\begin{aligned} L_{r+1} \cdot R_{r+1} &= R_r \cdot (L_r \oplus O_r) \\ L_{r+1} \cdot R_{r+1} &= R_r \cdot L_r \oplus R_r \cdot O_r \\ L_{r+1} \cdot R_{r+1} \oplus L_r \cdot R_r &= I_r \cdot O_r \end{aligned} \tag{2.7}$$

For n round;

$$\begin{aligned} L_2 \cdot R_2 \oplus L_1 \cdot R_1 &= I_1 \cdot O_1 \\ L_3 \cdot R_3 \oplus L_2 \cdot R_2 &= I_2 \cdot O_2 \\ &\vdots \\ L_{n+1} \cdot R_{n+1} \oplus L_n \cdot R_n &= I_n \cdot O_n \end{aligned}$$

Constructing Bilinear Characteristics:

In this part, constructing bilinear characteristic for one round will be explained at first, then it will be extended to the second round. Same notation as in [14] will be used and details of the properties of bilinear characteristics can be found in [14], also.

Let $S : GF(2^n) \times GF(2^n) \rightarrow GF(2)$ such that $S(L_1,\ldots,L_n;R_1,\ldots,R_n) = \sum s_{ij}L_iR_j$ be a homogeneous bilinear Boolean function.

Let f_K be the round function, $(I_1,\ldots,I_n)$ be the input and $(O_1,\ldots,O_n)$ be the output of the round function f_K. Then, $f_K(I_1,\ldots,I_n) = (O_1,\ldots,O_n)$. Since $(I_1,\ldots,I_n) = (R_1,\ldots,R_n)$, we can write this equation as $f_K(R_1,\ldots,R_n) = (O_1,\ldots,O_n)$.

Assume that there exist two linear combinations u and v such that the equation

$$\sum s_{ij}O_iR_j \oplus \sum u_iO_i \oplus \sum v_iR_i = 0 \tag{2.8}$$

holds with probability $p \neq 1/2$. As can be seen from the Figure 2.11, $C_i = L_i \oplus O_i$. From bilinearity, we have

$$\sum s_{ij} L_i R_j \oplus \sum s_{ij} O_i R_j = \sum s_{ij} C_i R_j.$$

By using Equation 2.8, we have the following equation with probability $p(K)$:

$$\sum s_{ij} L_i R_j \oplus \sum u_i L_i \oplus \sum v_i R_i \quad = \quad \sum s_{ij} C_i R_j \oplus \sum u_i C_i. \tag{2.9}$$

Furthermore, since v is a linear combination, it can be arbitrarily split in two parts such that $v_i = v_i^{(1)} \oplus v_i^{(2)}$. Then, $\sum v_i R_i = \sum v_i^{(1)} R_i \oplus \sum v_i^{(2)} R_i$ for all $i = 1, \ldots, n$ and Equation 2.9 becomes

$$\sum s_{ij} L_i R_j \oplus \sum u_i L_i \oplus \sum v_i^{(1)} R_i \quad = \quad \sum s_{ij} C_i R_j \oplus \sum u_i C_i \oplus \sum v_i^{(2)} R_i.$$

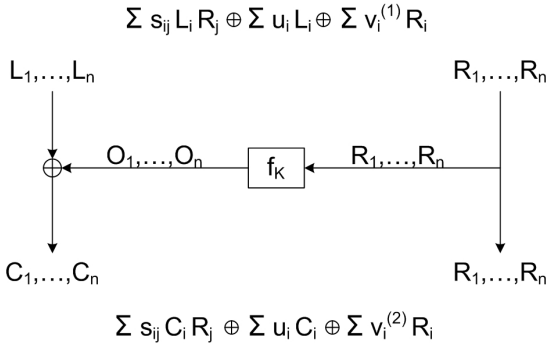

Figure 2.11: 1-Round Bilinear Characteristic for Feistel Networks

Extending one round bilinear characteristic to the second round:

Applying the same procedure as in the previous part, bilinear approximation for the second round can be found. Again, we assume that there exist two linear combinations x and w such that the equation

$$\sum t_{ij} C_i P_j \oplus \sum w_i C_i \oplus \sum x_i P_i \quad = \quad 0 \tag{2.10}$$

holds with probability $p' \neq 1/2$. From the equality $D_j = R_j \oplus P_j$ and bilinearity, we obtain

$$\sum t_{ij} C_i D_j \oplus \sum t_{ij} C_i R_j = \sum t_{ij} C_i P_j.$$

Combining this equation with Equation 2.10, we get the following equation with probability $p(K') \neq 1/2$:

$$\sum t_{ij} C_i R_j \oplus \sum x_i R_i \oplus \sum w_i^{(1)} C_i \quad = \quad \sum t_{ij} C_i D_j \oplus \sum x_i D_i \oplus \sum w_i^{(2)} C_i.$$

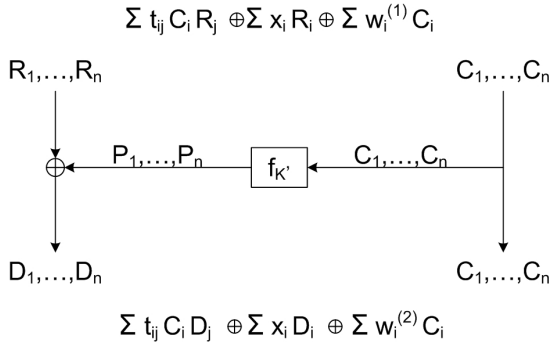

Figure 2.12: 2^{nd} Round of the Bilinear Characteristic

Finally, general expression for an n-round bilinear characteristic is

$$L_1[\alpha_0] \cdot R_1[\beta_0] \oplus R_1[\gamma_0] \oplus L_1[\delta_0] \oplus L_n[\alpha_n] \cdot R_n[\beta_n] \oplus R_n[\gamma_n] \oplus L_n[\delta_n] = 0$$

with some bias q.

Differential-Bilinear Cryptanalysis

Let the block cipher $\mathbf{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ be a cascade of two sub-ciphers $\mathbf{E} = \mathbf{E}_1 \circ \mathbf{E}_0$. Assume that there exists a differential $\Omega_P \rightarrow \Omega_T$ with probability p for E_0^K and a bilinear approximation with bias q for E_1^K. If the plaintext pair, (P, P^*), satisfies the difference, i.e. $P \oplus P^* = \Omega_P$, then $T \oplus T^* = \Omega_T$ with probability p, where $T = E_0^K(P)$ and $T^* = E_0^K(P^*)$. T and T^* satisfy the following equation:

$$T_L[\alpha_0] \cdot T_R[\beta_0] \oplus T_L[\gamma_0] \oplus T_R[\delta_0] \quad = \quad T_L^*[\alpha_0] \cdot T_R^*[\beta_0] \oplus T_L^*[\gamma_0] \oplus T_R^*[\delta_0] \qquad (2.11)$$

with probability $p \cdot 1 + (1-p) \cdot \frac{1}{2} = \frac{1}{2} + \frac{p}{2}$, where T_L is the left and T_R is the right half of T. Then, for the ciphertext pairs (C, C^*) who fulfill the differential-bilinear characteristic, it is expected for that pairs to satisfy the equation

$$C_L[\alpha_n] \cdot C_R[\beta_n] \oplus C_L[\gamma_n] \oplus C_R[\delta_n] \quad = \quad C_L^*[\alpha_n] \cdot C_R^*[\beta_n] \oplus C_L^*[\gamma_n] \oplus C_R^*[\delta_n] \qquad (2.12)$$

with probability $p \cdot \left[(\frac{1}{2} + q) \cdot (\frac{1}{2} + q) + (\frac{1}{2} - q) \cdot (\frac{1}{2} - q) \right] + (1-p) \cdot \frac{1}{2} = \frac{1}{2} + 2pq^2$.

Differential-bilinear cryptanalysis resembles differential-linear cryptanalysis in the sense of the attack process. In both attacks, first, a set of plaintext pairs each having a difference Ω_P are chosen, then these pairs are encrypted for n rounds and finally, by guessing some subkey bits, it is checked whether the obtained ciphertext pairs satisfy the (bi)linear approximation or not. That is the way how differential-(bi)linear attacks work. However, there is a difference between these two attacks in the way of combination of the differential characteristic and the (bi)linear approximation. In differential-linear cryptanalysis, any differential can be concatenated to the linear approximation whereas in differential-bilinear cryptanalysis, the differential characteristic should satisfy some conditions. Knowing the difference $T_L[\alpha_0] \oplus T_L^*[\alpha_0]$ and $T_R[\beta_0] \oplus T_R^*[\beta_0]$ does not give concrete results on the difference $T_L[\alpha_0] \cdot T_R[\beta_0] \oplus T_L^*[\alpha_0] \cdot T_R^*[\beta_0]$. Therefore, the differential that will be combined with the linear approximation should guarantee the knowledge of the difference $T_L[\alpha_0] \cdot T_R[\beta_0] \oplus T_L^*[\alpha_0] \cdot T_R^*[\beta_0]$.

Moreover, linear terms in the approximation does not affect the attack since the attacker is interested in the difference in the output mask of two encryptions and the sign of the bias is not important.

2.3.1 6-Round Differential-Bilinear Distinguisher

This 6-round differential-bilinear distinguisher is constructed by combining 3-round differential characteristic with a 3-round bilinear approximation.

3-Round Bilinear Approximation:

This 3-round bilinear approximation was presented in [14] and has a bias of $q = 1.66 \cdot 2^{-3}$. Note that, the best 3-round linear approximation of DES has a bias of $q = 1.56 \cdot 2^{-3}$ which is lower than that of the bilinear approximation. However, in this case the probability of the differential characteristic is less than that in differential-linear attack.

This bilinear approximation is the following:

$$L_1[3, 8, 14, 25] \oplus R_1[17] \oplus L_1[3] \cdot R_1[16, 17, 20] \oplus L_4[3, 8, 14, 25] \oplus R_4[17] \oplus$$

$$L_4[3] \cdot R_4[16, 17, 20] = K[sth] \oplus L_1[3] \cdot K[sth'] \oplus L_4[3] \cdot K[sth'']$$

3-Round Differential Characteristic:

After obtaining the bilinear approximation, differential characteristic can be found. However,

the differential that the bilinear approximation will be concatenated to, should give zero difference in $L_1[3] \cdot R_1[16, 17, 20]$. The best 3-round differential characteristic that meets this condition was given in [18] and depicted in Figure 2.13.

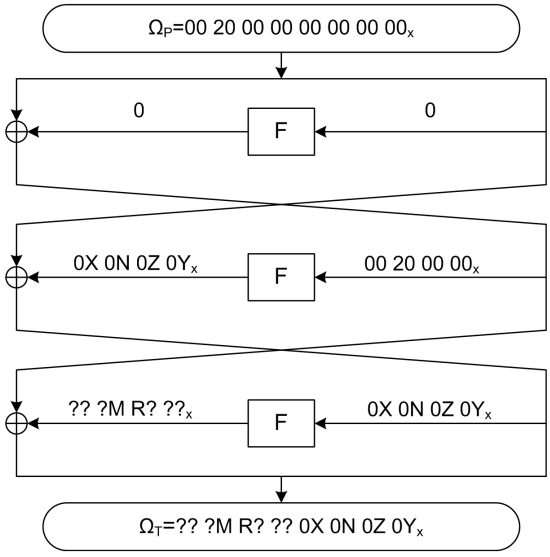

Figure 2.13: 3-Round Differential Characteristic of DES

This differential has probability p of 83/128.

Therefore, as stated before, the probability of the differential-bilinear distinguisher can be calculated as $\frac{1}{2} + 2pq^2 = \frac{1}{2} + 2 \cdot \frac{83}{128} \cdot (1.66 \cdot 2^{-3})^2 \approx 0.562$

2.3.2 8-Round Differential-Bilinear Attack on DES

In this section, an application of the differential-bilinear cryptanalysis will be explained on DES cipher. This attack was proposed by Biham et al. [18] in 2005.

Attack Procedure:

1. Take $N = 2^{12}$ plaintexts, consisting of 2^8 structures, chosen by selecting

   (a) Any plaintext P_0,

   (b) The plaintexts $P_1,\ldots,P_{15}$ from P_0 by varying four bits (output of S_3 in round 1) masked by 04 01 01 04 00 00 00 00,

(c) The plaintexts $P_{16}, \ldots, P_{31}$ from $P_i = P_{i-16} \oplus 00\ 00\ 00\ 00\ 00\ 20\ 00\ 00$.

2. Encrypt these plaintext pairs and obtain their corresponding ciphertext pairs.

3. For each value of the 4 bits of $K_{1,3}$ and 8 bits of $K_{8,1}$ and $K_{8,2}$

   (a) Partially encrypt all plaintexts in the first round and find the pairs that have difference Ω_P before round 2.

   (b) Partially decrypt the ciphertext pairs and count how many have the same parity of the subset of the output mask.

4. Output the subkey guess with the highest counter corresponding to $K_{8,1}$ and $K_{8,2}$ along with the guess $K_{1,3}$.

Time complexity of the attack is $2^4(2^{12} \cdot \frac{1}{8} \cdot \frac{1}{8} + 2^{12} \cdot 2^8 \cdot \frac{2}{8} \cdot \frac{1}{8}) \approx 2^{19}$ 8-round encryptions of DES.

Required number of chosen plaintexts is approximately $p^{-2}q^{-4} \approx 2^{10}$ for this attack. For $N = 2^{12}$, success rate of the attack is over 75% [41]. Therefore, data complexity of the 8-round differential-bilinear attack is 2^{12} chosen plaintexts.

Table 2.3: Comparison of the Differential-Linear and Differential-Bilinear Attacks on DES

Attack Type	Reference	Round	Data Complexity (CP)	Time Complexity
Differential-Linear	[11]	8	2^9	2^{14}
Enhanced Differential-Linear	[12]	8	$2^{14.8}$	$2^{14.8}$
Enhanced Differential-Linear	[12]	9	2^{16}	2^{29}
Differential-Bilinear	[41]	8	2^{12}	2^{19}

2.4 Higher Order Differential-Linear Cryptanalysis

Higher order differential-linear cryptanalysis is another cryptanalysis method which combines higher order differential technique with linear cryptanalysis. Higher order differential method was developed by Knudsen [15] in 1994 and is a generalization of differential cryptanalysis in the sense of using differentials of more than two plaintexts. In some cases, looking at the difference between a plaintext/ciphertext pair may not be exploited. Instead, analyzing the XOR value of the plaintexts/ciphertexts in a structured set can be more advantageous. In a

higher order differential attack, the development of the XOR value of the intermediate data during the encryption of a plaintext set is analyzed.

In higher order differential-linear cryptanalysis, a differential characteristic with probability p is concatenated with a linear approximation having a bias q.

Proposition 2.4.1 *Let the block cipher* $\boldsymbol{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ *be a cascade of two sub-ciphers* $\boldsymbol{E} = \boldsymbol{E}_1 \circ \boldsymbol{E}_0$. *Assume that there exists a higher order differential* $\Omega_P \rightarrow \Omega_T$ *with probability p for* E_0^K *and a linear approximation* $\lambda_T \rightarrow \lambda_C$ *with bias q for* E_1^K. *If a plaintext set* $\{P_1, \cdots, P_k\}$ *such that the higher order differential predicts the value of* $\bigoplus_{i=1}^{k} T_i$, *then*

$$Pr(\lambda_T \cdot (T_1 \oplus \cdots \oplus T_k) = \lambda_C \cdot (C_1 \oplus \cdots \oplus C_k)) = \frac{1}{2} + 2^{k-1} pq^k \text{ where } C_i = E_1^K(T_i) \text{ for } i = 1, \ldots, k.$$

2.4.1 Higher Order Differential-Linear Cryptanalysis of FEAL

The idea of higher order differential-linear cryptanalysis of Fast Data Encipherment Algorithm (FEAL) was given in [18]. Biham et al. proposed a generic higher order differential characteristic with probability 1 for the ciphers of Feistel structure and having a bijective round function. This 3-round differential characteristic will be explained in Section 2.4.1.2. FEAL was subjected to too many cryptanalytic attacks so far. Performing higher order differential-linear attack on FEAL will not give a satisfactory result in terms of cryptanalysis, but the purpose of this attack is just to show the applicability of this type of an attack on FEAL.

2.4.1.1 Description of FEAL

FEAL is a Feistel Network block cipher operating on 64-bit blocks. It was designed by NTT (Nippon Telegraph and Telephone Corporation) in 1987 as software for 8-bit microprocessors in smart cards. Actually, FEAL was designed to be a replacement of DES since it was very fast in software. However, the primary versions, namely FEAL-4 and FEAL-8 were discovered as insufficient in terms of security. For this reason, in 1990, FEAL-N and FEAL-NX were developed to improve the security where FEAL-N uses 64-bit keys, FEAL-NX uses 128-bit keys for $N \geq 32$ rounds. Only difference between these versions is the number of rounds and the key size. From now on, FEAL specifications will be given according to FEAL-8 since the

versions of FEAL for greater number of rounds are out of interest in this section. FEAL-8 is represented in Figure 2.14.

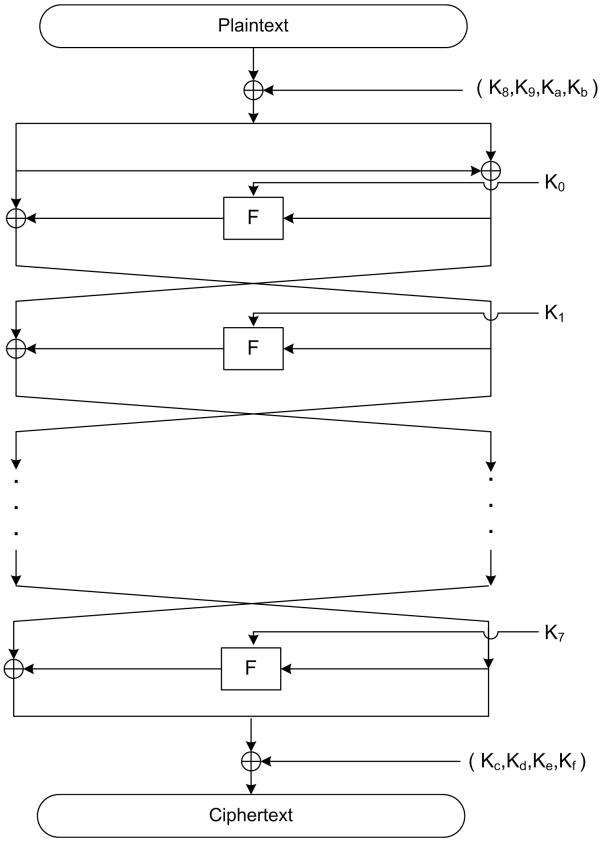

Figure 2.14: Structure of FEAL-8

FEAL F-Function:

The F-function is a function that maps 32-bit input to 32-bit output and composed of two S-boxes and XOR operations. F-function of FEAL is shown in Figure 2.15.

S-boxes are defined as follows:

$$S_0(X, Y) = ROL2((X + Y)\ mod\ 256) \text{ and } S_1(X, Y) = ROL2((X + Y + 1)\ mod\ 256)$$

where X and Y are 8-bit blocks and $ROL2$ is 2-bit left rotation operation.

2.4.1.2 6-Round Higher Order Differential-Linear Distinguisher of FEAL

The basic 6-round higher order differential-linear distinguisher was generated by combining a 3-round higher order differential characteristic with probability 1 and a 3-round linear char-

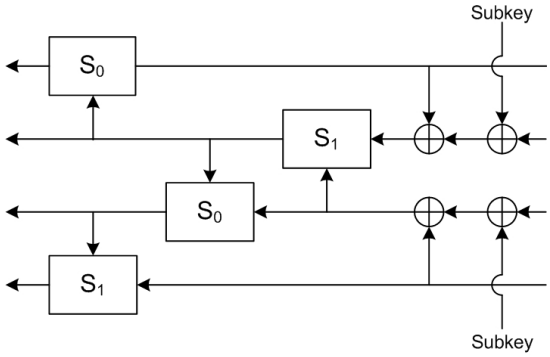

Figure 2.15: FEAL F-Function

acteristic with probability 1.

3-Round Higher Order Differential Characteristic:

This generic 3-round higher order differential path was given in [18]. In constructing this path, structured sets satisfying some conditions are used. The following proposition clarifies these conditions.

Proposition 2.4.2 *Let E be a block cipher of Feistel Network and having a bijective round function. Let C be a constant value for all plaintexts, P be the set of all permutations and B be the sum of all texts in the structure which is equal to zero. Then, a plaintext set of the form (P, C) will result in a set of the form (B, P) with probability 1 after three rounds of E.*

This generic higher order differential characteristic is depicted in Figure 2.16.

3-Round Linear Characteristic:

The following 3-round linear characteristic was introduced in [8]. This approximation is one of the probability 1 linear characteristics so, another 3-round linear path can be used in constructing 6-round distinguisher. Also, it's worth noting that the bit orientation used in Figure 2.17 is different than the one used in this section. In Figure 2.17, bits are numbered from right to left starting from 0. Therefore, in order to provide consistency, it is better to modify this characteristic according to the notation followed in this section. The modified 3-round linear characteristic is depicted in Figure 2.18.

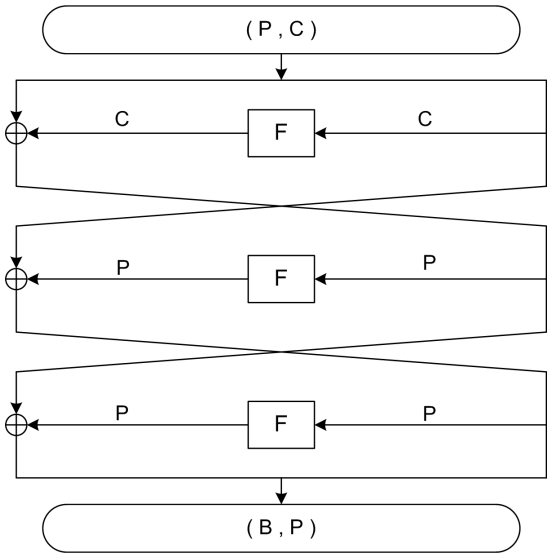

Figure 2.16: 3-Round Higher Order Differential Characteristic of FEAL

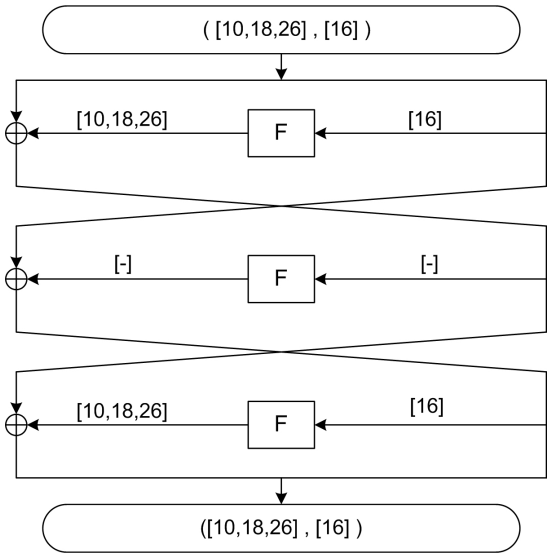

Figure 2.17: 3-Round Linear Characteristic of FEAL in [8]

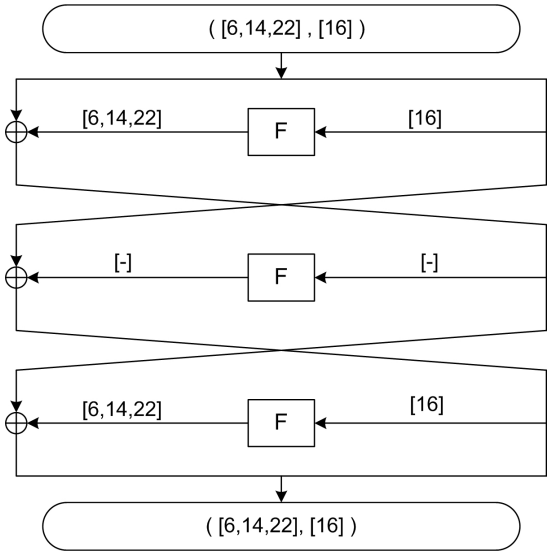

Figure 2.18: The Modified 3-Round Linear Characteristic of FEAL

2.4.1.3 7-Round Higher Order Differential-Linear Attack on FEAL

Attack Procedure:

1. Take any plaintext $P_0 = (P_{0_L}, P_{0_R})$

2. Obtain 2^{32} different plaintexts from all possible permutations of P_{0_L} and remain the right halves constant. So, the plaintext set is of the form (P, C).

3. Encrypt this set of plaintexts for seven rounds under the unknown key and get the corresponding ciphertexts, $C_i = (C_{i_L}, C_{i_R})$ for $i = 0, \ldots, 2^{32} - 1$.

4. Guess the last round's subkey and check whether $\bigoplus_{i=0}^{2^{32}-1}[F(K_7, C_{i_R}) \oplus C_{i_L}] \cdot [16] = 0$.

2.5 Differential-Nonlinear Cryptanalysis

In differential-nonlinear cryptanalysis, a nonlinear approximation is concatenated to the differential characteristic with some probability. The reason why a nonlinear approximation is used instead of a linear approximation comes/arises from the algebraic structure of the cipher. In some ciphers, output bits can be represented as a linear expression in terms of the input bits. However, there are some cases that you cannot express the input and output bits of the

round function in a linear way. Also, there are some other cases that you can linearize the nonlinear expression up to some extend by adding some constraints, e.g., fixing specific bits. This operation may cause a decrease in probability. Therefore, using a nonlinear approximation is effective when the probability of this nonlinear approximation is larger than those of any linear approximations.

2.5.1 A Differential-Nonlinear Attack on 32-Round SHACAL-2

2.5.1.1 Description of SHACAL-2

In 2000, Handschuh and Naccache proposed a block cipher named SHACAL [19] as a submission of the NESSIE (New European Schemes for Signatures, Integrity, and Encryption) project. It has 160-bit block length and based on the hash function SHA-1[21]. Then, in 2001, they proposed two versions of SHACAL, namely SHACAL-1 and SHACAL-2 [20]. SHACAL-1 is same as SHACAL, but SHACAL-2 has a block length of 256-bit and is based on the hash function SHA-256 [22]. Both SHACAL-1 and SHACAL-2 were submitted to the project, but only SHACAL-2 was selected as one of the 17 NESSIE finalists.

SHACAL-2 consists of 64 rounds and supports variable key sizes up to 512 bits. Encryption procedure of SHACAL-2 is as follows:

The 256-bit plaintext is divided into eight 32-bit words, namely (A, B, C, D, E, F, G, H). Let X^i denote the word X before i^{th} round. Then, the plaintext and the ciphertext can be represented as $(A^0,B^0,C^0,D^0,E^0,F^0,G^0,H^0)$ and $(A^{64},B^{64},C^{64},D^{64},E^{64},F^{64},G^{64},H^{64})$. Let W^i be the 32-bit round subkeys and K^i be the 32-bit round constants. Then, i^{th} round encryption is :

$$
\begin{aligned}
T_1^{i+1} &= H^i \boxplus \sum_1(E^i) \boxplus Ch(E^i, F^i, G^i) \boxplus K^i \boxplus W^i \\
T_2^{i+1} &= \sum_0(A^i) \boxplus Maj(A^i, B^i, C^i) \\
H^{i+1} &= G^i \\
G^{i+1} &= F^i \\
F^{i+1} &= E^i \\
E^{i+1} &= D^i \boxplus T_1^{i+1} \\
D^{i+1} &= C^i \\
C^{i+1} &= B^i \\
B^{i+1} &= A^i \\
A^{i+1} &= T_1^{i+1} \boxplus T_2^{i+1}
\end{aligned}
$$

for $i = 0, \ldots, 63$ where $\boxplus$ denotes the addition modulo 2^{32}. The round functions, namely Ch, Maj, Σ_0 and Σ_1 are defined as follows:

$$\begin{aligned} Ch(X,Y,Z) &= (X\&Y) \oplus (\neg X\&Z) \\ Maj(X,Y,Z) &= (X\&Y) \oplus (X\&Z) \oplus (Y\&Z) \\ \Sigma_0(X) &= S_2(X) \oplus S_{13}(X) \oplus S_{22}(X) \\ \Sigma_1(X) &= S_6(X) \oplus S_{11}(X) \oplus S_{25}(X) \end{aligned}$$

where $\neg X$ means the complement of 32-bit word X and $S_i(X)$ means the right rotation of X by i bit positions. Figure 2.19 shows i^{th} round of SHACAL-2.

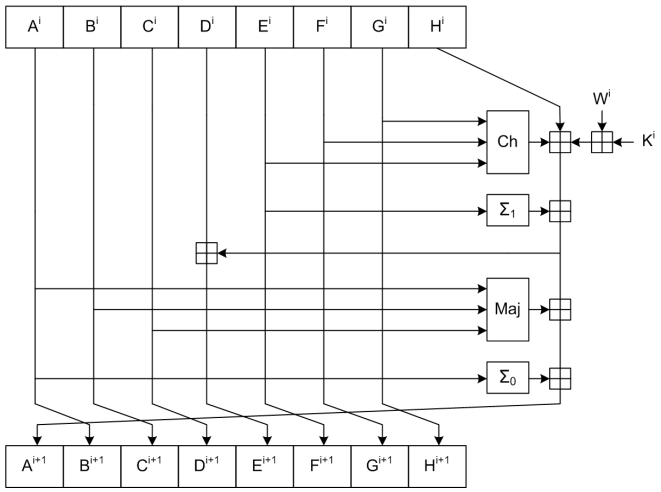

Figure 2.19: i^{th} Round of SHACAL-2

Key Scheduling Algorithm:

SHACAL-2 has a variable key size from 128 up to 512 bits. The keys which are shorter than 512 bits are padded with zeros to a 512-bit string. Let $W = W^0 \| W^1 \| \ldots \| W^{15}$ be the 512-bit key string. The key expansion which extends the 512-bit key W to 2048 bits is defined as follows:

$$\begin{aligned} W^i &= \sigma_1(W^{i-2}) \boxplus \sigma_0(W^{i-15}) \boxplus W^{i-16} \\ \sigma_0(X) &= S_7(X) \oplus S_{18}(X) \oplus R_3(X) \\ \sigma_1(X) &= S_{17}(X) \oplus S_{19}(X) \oplus R_{10}(X) \end{aligned}$$

for $16 \leq i \leq 63$, where $R_i(X)$ denotes the right shift of X by i bit positions.

2.5.1.2 A 17-Round Differential-Nonlinear Distinguisher

As in the previous sections, this 17-round differential-nonlinear distinguisher is constructed by concatenating a nonlinear approximation to a differential characteristic. Here, a 14-round truncated differential characteristic with probability $2^{-18.7}$ is combined with a 3-round nonlinear approximation of SHACAL-2 to obtain 17-round distinguisher. Total probability of this 17-round differential-nonlinear distinguisher is $2^{-18.7} \cdot 1 + (1 - 2^{-18.7}) \cdot \frac{1}{2} = \frac{1}{2} + 2^{-19.7}$.

14-Round Differential Characteristic:

Before explaining how the differential trail was constructed, it is better to give some differential properties of SHACAL-2.

Differential Property 1 of SHACAL-2: The first differential property is related to some observations on XOR and modular addition operations. Assume that $Z = X \boxplus Y$ and $Z^* = X^* \boxplus Y^*$ where X,Y and X^*,Y^* are 32-bit words.

- If $X \oplus X^* = e_j$ and $Y = Y^*$, then $Z \oplus Z^* = e_{j,j+1,\ldots,j+k-1}$ with probability $\frac{1}{2^k}$ for $j < 31$, $k \geq 1$ and $j+k-1 \leq 30$. As a corollary of this property, for the case $j = 31$, $Z \oplus Z^* = e_{31}$ with probability 1.
- If $X \oplus X^* = e_j$ and $Y \oplus Y^* = e_j$, then $Z \oplus Z^* = e_{j+1,\ldots,j+k-1}$ with probability $\frac{1}{2^k}$ for $j < 31$, $k \geq 1$ and $j + k - 1 \leq 30$. For $j = 31$, $Z \oplus Z^* = 0$ with probability 1.
- If $X \oplus X^* = e_{i,\sim}$, $Y \oplus Y^* = e_{j,\sim}$ and $i > j$ then $Z \oplus Z^* = e_{j,\sim}$ and this also means that $Z \oplus Z^* = z_k$ where $0 \leq k < 1$.

Differential Property 2 of SHACAL-2: The second differential property of SHACAL-2 is followed from the observations on the functions Ch and Maj. In order to compute the probability of the differential characteristic which is given in Table 2.5, we need to know the relation between the input and output differences of Ch and Maj functions. Table 2.4 shows the XOR difference distribution of these functions.

For example, if the input differences for Δx, Δy and Δz are respectively 1,1,0, then output difference of the Ch and Maj functions is either 0 or 1 with probability $1/2$ depending on the inputs. Therefore, if we can fix some input bits, it is possible to increase the probability. More specifically, let 0,1,0 and 1,0,0 be the inputs for the Ch function. Since Ch(0,1,0)=0

Table 2.4: Difference Distribution Table of Ch and Maj Functions

Δx	Δy	Δz	Ch	Maj
0	0	0	0	0
0	0	1	0/1	0/1
0	1	0	0/1	0/1
1	0	0	0/1	0/1
0	1	1	1	0/1
1	0	1	0/1	0/1
1	1	0	0/1	0/1
1	1	1	0/1	1

and $Ch(1,0,0)=0$, then $Ch(0,1,0) \oplus Ch(1,0,0) = 0$ with probability 1. In order to improve the probability (of the first few round) which results from the Ch and Maj functions, the following bits of the plaintext were fixed:

$a_9 = b_9,\ a_{18} = b_{18},\ a_{29} = b_{29},\ a_{31} = b_{31}$

$e_6 = 1,\ e_9 = 1,\ e_{18} = 1,\ e_{20} = 1,\ e_{25} = 1,\ e_{29} = 1,\ e_{31} = 1$

Let $P = (A, B, C, D, E, F, G, H)$ and $P^* = (A^*, B^*, C^*, D^*, E^*, F^*, G^*, H^*)$ be the plaintext pairs. If the plaintext difference, $P \oplus P^*$, is chosen as $(0,0,e_{M_1},0,0,e_{31},e_{M_2},0)$ where $M_1 = \{9, 18, 29\}$, $M_2 = \{6, 9, 18, 20, 25, 29\}$ and the plaintext bits given above are fixed, then after 14 rounds, the least significant bit of output difference in the eighth word, $\Delta h_0^{14} = 0$ with probability 2^{-22}. 14-round differential characteristic is depicted in Table 2.5:

The probabilities given in Table 2.5 can be computed by using differential properties of SHACAL-2. Besides, other 14-round truncated differential characteristics can be found by choosing different values for the difference ΔE^{10} such that $\Delta h_0^{14} = 0$. This will cause finding differential trails with probability less than 2^{-22} however the total probability can be increased up to

$$1 \cdot 2^{-22} + 4 \cdot 2^{-23} + 9 \cdot 2^{-24} + 16 \cdot 2^{-25} + 16 \cdot 2^{-26} + 42 \cdot 2^{-27} + 51 \cdot 2^{-28} \approx 2^{-18.7}$$

by considering all these 14-round truncated differentials. Some of the possible differences for ΔE^{10} are presented in Table 2.6.

Now, we have a 14-round truncated differential characteristic with probability $2^{-18.7}$ and we are going to combine this characteristic with a 3-round nonlinear approximation to obtain a

Table 2.5: 14-Round Differential Characteristic for SHACAL-2

($M_1 = \{9, 18, 29\}$, $M_2 = \{6, 9, 18, 20, 25, 29\}$, $M_3 = \{6, 9, 18, 20, 25\}$)

Round(r)	ΔA^r	ΔB^r	ΔC^r	ΔD^r	ΔE^r	ΔF^r	ΔG^r	ΔH^r	Prob.
Input(r=0)	0	0	e_{M_1}	0	0	e_{31}	e_{M_2}	0	1
1	e_{31}	0	0	e_{M_1}	e_{31}	0	e_{31}	e_{M_2}	2^{-10}
2	0	e_{31}	0	0	0	e_{31}	0	e_{31}	2^{-2}
3	0	0	e_{31}	0	0	0	e_{31}	0	2^{-2}
4	0	0	0	e_{31}	0	0	0	e_{31}	1
5	e_{31}	0	0	0	0	0	0	0	2^{-4}
6	e_{M_1}	e_{31}	0	0	0	0	0	0	1
7	z_0	e_{M_1}	e_{31}	0	0	0	0	0	1
8	?	z_0	e_{M_1}	e_{31}	0	0	0	0	1
9	?	?	z_0	e_{M_1}	e_{31}	0	0	0	2^{-4}
10	?	?	?	z_0	$e_{M_{3,\sim}}$	e_{31}	0	0	1
11	?	?	?	?	z_0	$e_{M_{3,\sim}}$	e_{31}	0	1
12	?	?	?	?	?	z_0	$e_{M_{3,\sim}}$	e_{31}	1
13	?	?	?	?	?	?	z_0	$e_{M_{3,\sim}}$	1
Output(r=14)	?	?	?	?	?	?	?	z_0	

Table 2.6: Possible ΔE^{10} Values

ΔE^{10}	Prob.	ΔE^{10}	Prob.	ΔE^{10}	Prob.
$e_{6,9,18,20,25,\sim}$	2^{-22}	$e_{6,7,9,18,20,25,\sim}$	2^{-23}	$e_{6,9,10,18,20,25,\sim}$	2^{-23}
$e_{6,9,18,19,20,25,\sim}$	2^{-23}	$e_{6,9,18,20,21,25,\sim}$	2^{-23}	$e_{6,7,9,10,18,20,25,\sim}$	2^{-24}
$e_{6,7,9,18,19,20,25,\sim}$	2^{-24}	$e_{6,7,9,18,20,21,25,\sim}$	2^{-24}	$e_{6,9,10,18,19,20,25,\sim}$	2^{-24}
$e_{6,9,10,18,20,21,25,\sim}$	2^{-24}	$e_{6,9,18,19,20,21,25,\sim}$	2^{-24}	$e_{6,7,8,9,18,20,25,\sim}$	2^{-24}
$e_{6,9,18,19,25,\sim}$	2^{-24}	$e_{6,9,18,20,21,22,25,\sim}$	2^{-24}		

17-round differential-nonlinear distinguisher.

3-Round Nonlinear Approximation:

Definition 2.5.1 *The nonlinear function NF^{r+3} is a function of*

$$NF(A^{r+3}, B^{r+3}, \ldots, H^{r+3}, K^r, K^{r+1}, K^{r+2}, W^r, W^{r+1}, W^{r+2})$$

where $0 \leq r \leq 61$.

Since output difference of the truncated differential characteristic is $\Delta h_0^{14} = 0$, we need to determine the nonlinear aproximation of h_0 between rounds 14 and 17. So, we try to express h_0^{14} in terms the outputs of the nonlinear function NF^{17}.

The value h_0^r can be represented as the output of NF^{r+3} as follows:

$$\begin{aligned} h_0^r &= c_0^{r+3} \oplus d_2^{r+3} \oplus d_{13}^{r+3} \oplus d_{22}^{r+3} \oplus (d_0^{r+3} \& (e_0^{r+3} \oplus t_{1,0}^{r+3})) \oplus (d_0^{r+3} \& (f_0^{r+3} \oplus t_{1,0}^{r+2})) \\ &\oplus ((e_0^{r+3} \oplus t_{1,0}^{r+3}) \& (f_0^{r+3} \oplus t_{1,0}^{r+2})) \oplus h_6^{r+3} \oplus h_{11}^{r+3} \oplus h_{25}^{r+3} \oplus (h_0^{r+3} \& h_0^{r+2}) \\ &\oplus ((\neg h_0^{r+3}) \& h_0^{r+1}) \oplus k_0^r \oplus w_0^r \end{aligned}$$

In order to get rid of the terms $h_0^{r+1},\ h_0^{r+2},\ t_{1,0}^{r+2},\ t_{1,0}^{r+3}$, represent these terms as follows:

$$h_0^{r+1} = t_{1,0}^{r+2} \oplus g_6^{r+3} \oplus g_{11}^{r+3} \oplus g_{25}^{r+3} \oplus (g_0^{r+3} \& h_0^{r+3}) \oplus ((\neg g_0^{r+3}) \& h_0^{r+2}) \oplus k_0^{r+1} \oplus w_0^{r+1}$$

$$h_0^{r+2} = t_{1,0}^{r+3} \oplus f_6^{r+3} \oplus f_{11}^{r+3} \oplus f_{25}^{r+3} \oplus (f_0^{r+3} \& g_0^{r+3}) \oplus ((\neg f_0^{r+3}) \& h_0^{r+3}) \oplus k_0^{r+2} \oplus w_0^{r+2}$$

$$t_{1,0}^{r+2} = b_0^{r+3} \oplus c_2^{r+3} \oplus c_{13}^{r+3} \oplus c_{22}^{r+3} \oplus (c_0^{r+3} \& d_0^{r+3}) \oplus (c_0^{r+3} \& (e_0^{r+3} \oplus t_{1,0}^{r+3})) \oplus (d_0^{r+3} \& (t_{1,0}^{r+3} \oplus e_0^{r+3}))$$

$$t_{1,0}^{r+3} = a_0^{r+3} \oplus b_2^{r+3} \oplus b_{13}^{r+3} \oplus b_{22}^{r+3} \oplus (b_0^{r+3} \& c_0^{r+3}) \oplus (b_0^{r+3} \& d_0^{r+3}) \oplus (c_0^{r+3} \& d_0^{r+3})$$

Now, given the plaintext pairs P, P^* such that $P \oplus P^* = (0,0,e_{M_1},0,0,e_{31},e_{M_2},0)$ where M_1 and M_2 as defined above, it holds that $h_0^{14} = h_0^{*14}$ with probability $2^{-18.7}$. Also, we have $NF^{17} = NF^{*17}$ with probability $\frac{1}{2} + 2^{-19.7} = 2^{-18.7} + \frac{1}{2} \cdot (1 - 2^{-18.7})$ because

- if the truncated differential is satisfied with probability $2^{-18.7}$, then the equation $NF^{17} = NF^{*17}$ holds with probability 1 and
- if the differential is not satisfied, then it is assumed that this equation behaves random and holds with probability $1/2$.

Therefore, a 17-round differential-nonlinear distinguisher with probability $\frac{1}{2} + 2^{-19.7}$ was constructed and now it will be used to present a key recovery attack on 32-round SHACAL-2.

2.5.1.3 32-Round Attack on SHACAL-2

Attack Procedure:

1. Choose $2^{42.4} (= 8 \cdot (2^{-19.7})^{-2})$ plaintext pairs such that $P \oplus P^*$=(0,0,e_{M_1},0,0,e_{31},e_{M_2},0) and some bits of P are fixed as stated before.

2. Encrypt these plaintext pairs for 32 rounds and obtain the corresponding ciphertext pairs.

3. Guess a 463-bit key $W^{31}, W^{31}, \ldots, W^{20}, w_0^{19}, w_1^{19}, \ldots, w_{25}^{19}, w_0^{18}, w_1^{18} \ldots, w_{25}^{18}, w_0^{17}, w_1^{17} \ldots, w_{24}^{17}, w_0^{16}$ and w_0^{15} required for computing the value ΔNF^{17}.

4. Using the guessed keys in the previous step, partially decrypt each of the ciphertext pairs between the last 15 rounds and count how many times the equation $NF^{17} = NF^{*17}$ is satisfied. If this number is greater than or equal to $2^{41.4} + 2^{22.7} = 2^{42.4} \cdot (\frac{1}{2} + 2^{-19.7})$, then store the guessed key. If not, go to Step 3 and guess another key.

5. Do an exhaustive search for the remaining 49-bit keys.

Complexity Analysis:

- *Data Complexity:* $2^{42.4}$ chosen plaintext pairs $\Rightarrow 2^{43.4}$ chosen plaintexts.

- *Time Complexity:* Time complexity of Step 2 is $2^{43.4}$ 32-round SHACAL-2 encryptions. Step 4 requires $\frac{1}{2} \cdot \frac{15}{32} \cdot 2^{43.4} \cdot 2^{463} \approx 2^{504.2}$ 32-round SHACAL-2 encryptions on average. Expected number of remaining keys after Step 4 is about $2^{449.7}$, so time complexity of Step 5 is about $2^{449.7} \cdot 2^{49} = 2^{498.7}$ 32-round SHACAL-2 encryptions.

- *Memory Complexity:* Each ciphertext requires 32 memory bytes, so in total $2^{43.4} \cdot 32 = 2^{48.4}$ memory bytes are required.

2.6 Square-Nonlinear Cryptanalysis

In this section, another combined attack method which assembles square characteristics with nonlinear approximations is mentioned. Then, an application of this attack on 28-round SHACAL-2 is given as an example.

Square attack which is also known as integral cryptanalysis was introduced by Knudsen as a dedicated attack to the block cipher SQUARE [25]. The idea behind the square attack is based on differential cryptanalysis. However, in a square attack, propagation of complete sets of carefully chosen plaintexts is analyzed instead of pairs of plaintexts with a fixed XOR difference.

2.6.1 28-Round Square-Nonlinear Attack on SHACAL-2

A square-nonlinear attack is applied on 28-round SHACAL-2 [24] based on 13-round distinguisher. Before giving the distinguisher, some special sets which are used in the attack are given in the following:

- CS (Constant Set) : A set containing a single value, repeated 2^{32} times.
- PS (Permutation Set) : A set containing all 2^{32} possible values once in an arbitrary order.
- $-PS$: A set containing all 2^{32} possible values once, in ordering -x in case x is occurred in PS at the same round.
- BS (Balanced Set) : A set containing 2^{32} elements with arbitrary values such that their sum (modulo 2^{32}) is zero. If this property only holds for the $0-th$ bit, then it will be denoted as BS_0.

2.6.1.1 A 13-Round Square-Nonlinear Distinguisher

Similar to the cases in other combined attacks, a 13-round square-nonlinear distinguisher is generated by first finding a square characteristic and then concatenate a nonlinear approximation to this characteristic.

- **10-Round Square Characteristics:**

  If a plaintext set is of the form (0,0,PS,CS,1,CS,$-PS$,CS) where 0 and 1 represent constant sets composed of the 32-bit words 0×00000000 and $0 \times FFFFFFFF$, respectively, then the least significant bits of the eighth words after 10 rounds are balanced, i.e., $\bigoplus_{i=0}^{2^{32}-1} h_{i,0}^{10} = 0$ with probability 1. Table 2.7 shows this 10-round square characteristic.

Table 2.7: 10-Round Square Characteristic for SHACAL-2

Round(r)	A	B	C	D	E	F	G	H
Input(r=0)	0	0	PS	CS	1	CS	$-PS$	CS
1	CS	0	0	PS	CS	1	CS	$-PS$
2	PS	CS	0	0	CS	CS	1	CS
3	BS_0	PS	CS	0	CS	CS	CS	1
4	?	BS_0	PS	CS	CS	CS	CS	CS
5	?	?	BS_0	PS	CS	CS	CS	CS
6	?	?	?	BS_0	PS	CS	CS	CS
7	?	?	?	?	BS_0	PS	CS	CS
8	?	?	?	?	?	BS_0	PS	CS
9	?	?	?	?	?	?	BS_0	PS
Output(r=10)	?	?	?	?	?	?	?	BS_0

- **3-Round Nonlinear Approximation:**

  In this attack, again the same 3-round nonlinear approximation which was given in Section 2.5 was used since we have the equation $\bigoplus_{i=0}^{2^{32}-1} h_{i,0}^{10} = 0$.

$$\begin{aligned} h_0^{10} &= c_0^{13} \oplus d_2^{13} \oplus d_{13}^{13} \oplus d_{22}^{13} \oplus (d_0^{13} \& (e_0^{13} \oplus t_{1,0}^{13})) \oplus (d_0^{13} \& (f_0^{13} \oplus t_{1,0}^{12})) \\ &\oplus ((e_0^{13} \oplus t_{1,0}^{13}) \& (f_0^{13} \oplus t_{1,0}^{12})) \oplus h_6^{13} \oplus h_{11}^{13} \oplus h_{25}^{13} \oplus (h_0^{13} \& h_0^{12}) \\ &\oplus ((\neg h_0^{13}) \& h_0^{11}) \oplus k_0^{10} \oplus w_0^{10} \end{aligned}$$

  where

$h_0^{11} = t_{1,0}^{12} \oplus g_6^{13} \oplus g_{11}^{13} \oplus g_{25}^{13} \oplus (g_0^{13} \& h_0^{13}) \oplus ((\neg g_0^{13}) \& h_0^{12}) \oplus k_0^{11} \oplus w_0^{11}$,

$h_0^{12} = t_{1,0}^{13} \oplus f_6^{13} \oplus f_{11}^{13} \oplus f_{25}^{13} \oplus (f_0^{13} \& g_0^{13}) \oplus ((\neg f_0^{13}) \& h_0^{13}) \oplus k_0^{12} \oplus w_0^{12}$,

$t_{1,0}^{12} = b_0^{13} \oplus c_2^{13} \oplus c_{13}^{13} \oplus c_{22}^{13} \oplus (c_0^{13} \& d_0^{13}) \oplus (c_0^{13} \& (e_0^{13} \oplus t_{1,0}^{13})) \oplus (d_0^{13} \& (t_{1,0}^{13} \oplus e_0^{13}))$,

$t_{1,0}^{13} = a_0^{13} \oplus b_2^{13} \oplus b_{13}^{13} \oplus b_{22}^{13} \oplus (b_0^{13} \& c_0^{13}) \oplus (b_0^{13} \& d_0^{13}) \oplus (c_0^{13} \& d_0^{13})$.

Attack Procedure:

1. Choose 463 plaintext sets of the form (0,0,PS,CS,1,CS,$-PS$,CS).
2. Encrypt these plaintext sets for 28-rounds and get the corresponding ciphertext sets.
3. Guess a 463-bit key $W^{27}, W^{26}, \ldots, W^{16}, w_0^{15}, w_1^{15}, \ldots, w_{25}^{15}, w_0^{14}, w_1^{14} \ldots, w_{25}^{14}, w_0^{13}, w_1^{13} \ldots, w_{24}^{13}, w_0^{12}$ and w_0^{11}.
4. Using the guessed keys in the previous step, partially decrypt each of the ciphertext pairs for the last 15 rounds and check whether the equation $\bigoplus_{i=0}^{2^{32}-1} NF_i^{13} = 0$ is satisfied. If all the ciphertext sets satisfy this equation, then store the guessed key. If not, go to Step 3 and guess another key.
5. Do an exhaustive search for the remaining 49-bit keys.

Complexity Analysis:

- *Data Complexity:* $463 \cdot 2^{32}$ chosen plaintexts.
- *Time Complexity:* Step 1 requires $463 \cdot 2^{32}$ 28-round SHACAL-2 encryptions. Time complexity of Step 4 is $\frac{1}{2} \cdot \frac{15}{28} \cdot (2^{463} \cdot 2^{32} + 2^{462} \cdot 2^{32} + \ldots + 2^1 \cdot 2^{32}) \approx 2^{494.1}$ encryptions since in each 463 ciphertext set, possible subkeys are tried for each 2^{32} ciphertext. At each trial, about half of the subkeys are eliminated since the probability that equation holds is 1/2. Also, Step 5 requires about 2^{49} encryptions since the expected number of suggested keys after Step 4 is 1 ($=2^{463} \cdot 2^{-463}$). Therefore, total time complexity is about $2^{494.1}$ 28-round SHACAL-2 encryptions.
- *Memory Complexity:* Each ciphertext requires 32 memory bytes, so in total $463 \cdot 2^{32} \cdot 32 \approx 2^{45.9}$ memory bytes are required.

CHAPTER 3

IMPOSSIBLE DIFFERENTIAL CRYPTANALYSIS

This chapter is mainly about impossible differential cryptanalysis, a method which uses a combination of two differential distinguishers. Impossible differential idea was introduced in 1999 by Biham et al. to break 31 rounds of the 32-round cipher Skipjack [26]. Independently, an attack based on similar principles was proposed by Knudsen in 1998 to cryptanalyze 6-rounds of the cipher DEAL [27] which was one of the proposals for AES.

Impossible differential attack is a chosen plaintext attack and based on differential cryptanalysis. However, impossible differential cryptanalysis exploits differentials having low probabilities whereas differential cryptanalysis searches for high probability differentials. Actually, impossible differential attacks try to find events that never occur and use differentials with probability zero, called impossible differentials. Then, such impossible differentials are used to

- *Distinguish the cipher from a random permutation:*

  For a cipher, assume that we know the input difference α cannot produce (under any key) an output difference β , shown as $\alpha \nrightarrow \beta$. For a random permutation, a pair with an input difference α has an output difference β, shown as $\alpha \rightarrow \beta$, with probability 2^{-n}, where n is the block size. Therefore, we need about $O(2^n)$ pairs to distinguish the cipher from a random permutation. When we deal with impossible differential attacks, we should consider truncated differentials. In this case, for a random permutation $\alpha \rightarrow \beta$ with probability $|\beta|/2^n$. If the cardinality of β is 2^q, then in this case we need about $O(2^{n-q})$ pairs.

- *Recover the key:*

  We can find the key of a cipher by analyzing the rounds before and after the impossible differential and guessing the subkeys of these rounds. The keys satisfying the impossible differential will be wrong keys and with adequate number of pairs, we can eliminate all wrong keys. The impossible event in this case plays the role of a sieve; rejecting the wrong key guesses and leaving only the correct key. It is important to note that the miss-in-the-middle technique is only one of the ways to construct impossible events and that the sieving technique is only one of the possible ways to exploit them.

The main idea of impossible differential cryptanalysis is to detect two events with probability 1, whose conditions cannot met together. Combination of these two events will give us the impossible differential. Let our block cipher $\mathbf{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ be a cascade of three sub-ciphers $\mathbf{E} = \mathbf{E}_2 \circ \mathbf{E}_1 \circ \mathbf{E}_0$ as in [42, 41], where $\mathbf{E}_1$ denotes the rounds for which $\alpha \nrightarrow \beta$ holds, $\mathbf{E}_0$ denotes the rounds before $\mathbf{E}_1$, and $\mathbf{E}_2$ denotes the rounds after $\mathbf{E}_1$.

If a plaintext pair produces a difference of α after $\mathbf{E}_0$ under the guess for the subkeys used in $\mathbf{E}_0$, and its corresponding ciphertext pair produces a difference of β before $\mathbf{E}_2$ under the guess for the subkeys used in $\mathbf{E}_2$, then the combination of subkey guesses suggests that the pair satisfies an impossible differential. Since this cannot happen, subkey guesses is obviously wrong.

More specifically, assume K_0 is the guess for the subkeys used in $\mathbf{E}_0$, and K_2 is the guess for the subkeys used in $\mathbf{E}_2$. Then the guess (K_0,K_2) for the subkeys used in $\mathbf{E}_0$ and $\mathbf{E}_2$ is wrong if there is a pair of known plaintext/ciphertext pairs ((P, P^*), (C,C^*)) satisfying the following two conditions;

$$\mathbf{E}_0\left(K_0, P\right) \oplus \mathbf{E}_0\left(K_0, P^*\right) = \alpha, \tag{3.1}$$

$$\mathbf{E}_2^{-1}\left(K_2, C\right) \oplus \mathbf{E}_2^{-1}\left(K_2, C^*\right) = \beta. \tag{3.2}$$

Thus, given a sufficient number of matching plaintext/ciphertext pairs, an attacker can find the correct subkey by discarding the wrong guesses.

Assume that we want to recover r bits of the key. Then, there are $R = 2^r$ different keys which includes wrong keys and the correct key. Suppose we can eliminate k different keys with

using one plaintext pair (P, P^*). How many plaintext pairs are required to get rid of all wrong keys, i.e. to have the number of remaining wrong keys less than 1?

Now, in the first elimination, k keys are removed which is $\frac{k}{R}$ of all R keys. Then, number of remaining keys is $R - k = R \cdot (1 - \frac{k}{R})$.

Assuming that these k keys are selected randomly, the second pair removes $(R-k) \cdot \frac{k}{R}$ keys among the remaining keys and $k \cdot \frac{k}{R}$ keys which will be the same with the keys in the first elimination. After second elimination, number of remaining keys is $(R-k) - (R-k) \cdot \frac{k}{R} = (R-k) \cdot (1 - \frac{k}{R}) = R \cdot (1 - \frac{k}{R})^2$. Continuing in this manner, after p pairs, there wil be $R \cdot (1 - \frac{k}{R})^p$ keys left. In order to obtain the correct key, this number must be less than 1:

$$R \cdot (1 - \frac{k}{R})^p < 1 \Rightarrow R \cdot e^{-\frac{kp}{R}} < 1 \Rightarrow -\frac{kp}{R} < -\ln R \Rightarrow p > \frac{R \cdot \ln R}{k}$$

Therefore, by choosing p at least $\frac{R \cdot \ln R}{k}$, the correct key can be found with high probability.

Overview of this chapter is as follows. Section 3.1 explains an impossible differential attack on the block cipher IDEA. Section 3.2 shows an attack applied on AES and Section 3.3 mentions impossible differential cryptanalysis of CLEFIA. These block ciphers are chosen in order to exemplify the impossible differential attack on different types of ciphers.

3.1 Impossible Differential Cryptanalysis of IDEA

The attacks explained in this section are given by Biham et al. [29].

3.1.1 Description of IDEA

IDEA (International Data Encryption Algorithm) is first proposed by Lai and Massey in 1991 [28]. It is an 8.5 round non-Feistel block cipher with 64-bit block length and 128-bit key length.

IDEA uses two different half-round operations:

1. **Key mixing (T)** : This operation, denoted by T, divides the 64-bit block into four 16-bit

words and mixes the key with the data by using multiplication modulo $2^{16}+1$ (denoted by $\odot$) with $0 \equiv 2^{16}$ on the first and the fourth words, and addition modulo 2^{16} (denoted by $\boxplus$) on the second and the third words.

2. **M mixing** (M) : $M = s \circ MA$, where MA denotes a multiplication-addition structure and s denotes a swap of two middle words.

   Then, 8.5 round IDEA can be written as $T \circ s \circ (M \circ T)^8$.

   The input to the key mixing step T in round i is denoted by X^i, and the output of T which is the input to M is denoted by Y^i. Hence, the plaintext is denoted by X^1. One round of IDEA is depicted in Figure 3.1.

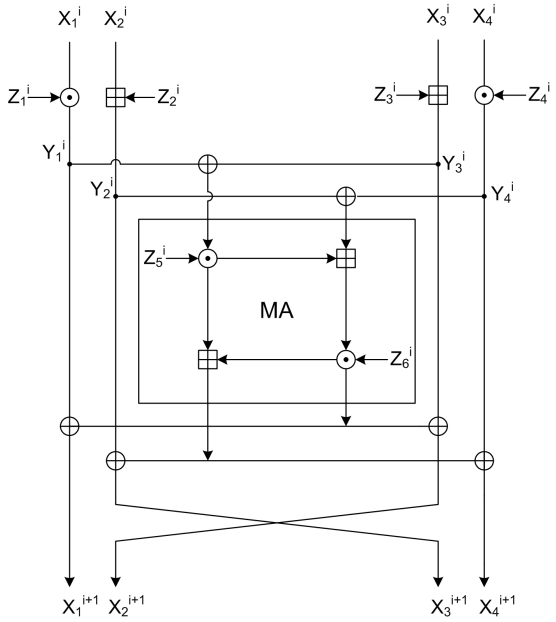

Figure 3.1: One Round of IDEA

Key schedule of IDEA

The key schedule of IDEA produces fifty two 16-bit subkeys from the 128-bit initial key. In the key mixing part, four subkeys are used and two subkeys are used in the M-mixing part. So, in each round a total of six subkeys are used. The initial 128-bit key is first splitted into eight 16-bit words and these are used as the first eight subkeys. This operation continues by rotating the 128-bit block by 25 bits to the left, then splitting into eight words and so on. Table 3.1 shows the corresponding subkeys to the rounds.

3.1.2 A 2.5-Round Impossible Differential of IDEA

Claim: Consider 2.5 rounds as $M \circ (T \circ M)^2$ of IDEA. Then, the input difference $(a, 0, a, 0)$ gives the output difference $(b, b, 0, 0)$ with probability zero, where a and b are nonzero 16-bit words.

Proof: Consider a pair with an input difference $(a, 0, a, 0)$ to M, for $a \neq 0$. In this case, as can be seen from Figure 3.2, the input difference to the first MA-structure is zero, hence the output difference of MA is zero. After swapping of the two middle words, the output difference of M becomes $(a, a, 0, 0)$. When the first T is applied, this difference becomes $(c, d, 0, 0)$, for some $c \neq 0$ and $d \neq 0$.

On the other hand, consider a pair with an output difference $(b, b, 0, 0)$, for $b \neq 0$ after 2.5 rounds. By going backwards, the input difference to M will be $(b, 0, b, 0)$ and the output difference of the last T will be $(e, 0, f, 0)$, for some $e \neq 0$ and $f \neq 0$.

So, we have the difference $(c, d, 0, 0)$ as the input to the second M and $(e, 0, f, 0)$ as the output of M. After the swap operation, the difference will be $(e, f, 0, 0)$. Hence, the input difference to MA of the half round is $(c, d) = (e, f)$ which is nonzero, but the output difference of this structure is $(0, 0)$. This leads to a contradiction since MA structure is a permutation.

Consequently, $(a, 0, a, 0) \nrightarrow (b, b, 0, 0)$, for $a \neq 0$, $b \neq 0$. Moreover, because of the symmetry, we can say that $(0, a, 0, a) \nrightarrow (0, 0, b, b)$.

Table 3.1: Subkey Bits Obtained from the 128-bit Initial Key

Round(i)	Z_1^i	Z_2^i	Z_3^i	Z_4^i	Z_5^i	Z_6^i
1	1-16	17-32	33-48	49-64	65-80	81-96
2	97-112	113-128	26-41	42-57	58-73	74-89
3	90-105	106-121	122-9	10-25	51-66	67-82
4	83-98	99-114	115-2	3-18	19-34	35-50
5	76-91	92-107	108-123	124-11	12-27	28-43
6	44-59	60-75	101-116	117-4	5-20	21-36
7	37-52	53-68	69-84	85-100	126-13	14-29
8	30-45	46-61	62-77	78-93	94-109	110-125
Last half round	23-38	39-54	55-70	71-86	-	-

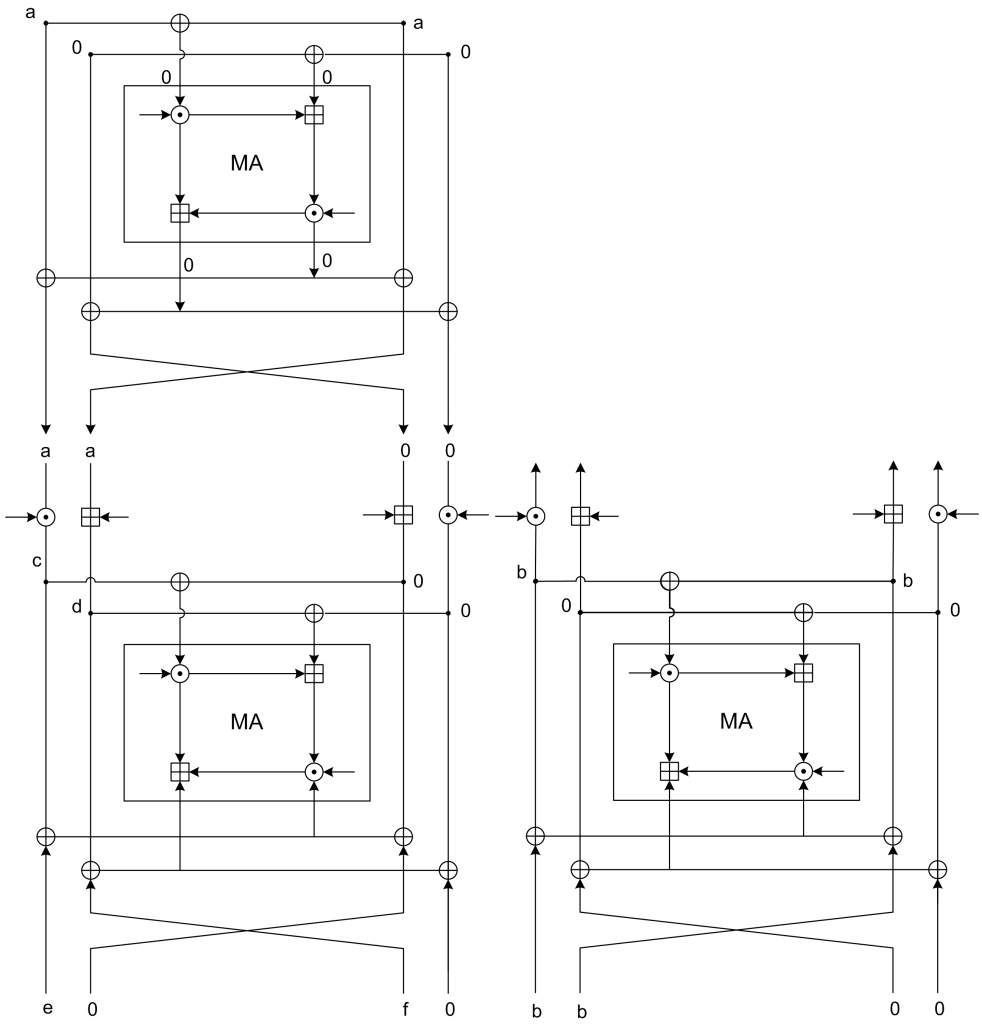

Figure 3.2: 2.5-Round Impossible Differential of IDEA

3.1.3 An Attack on 3.5-Round IDEA

This attack was applied to first 3.5 rounds of IDEA: $T \circ (M \circ T)^3$ by adding a half-round both to the beginning and to the end of the 2.5-round differential. A schematic representation of the attack given in Figure 3.3.

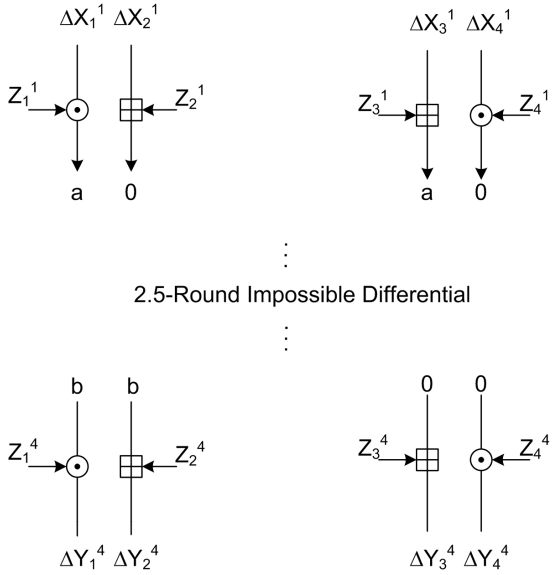

Figure 3.3: 3.5-Round Impossible Differential of IDEA

Attack Procedure:

1. Choose a structure of 2^{32} plaintexts $X^1 = ({X_1}^1, {X_2}^1, {X_3}^1, {X_4}^1)$ where ${X_2}^1$, ${X_4}^1$ are fixed values and ${X_1}^1$, ${X_3}^1$ take all possible values.

2. Choose about 2^{31} pairs (X^1, X^{1^*}) from the structure such that the difference between their corresponding ciphertexts (Y^4, Y^{4^*}) satisfy ${Y_3}^4 \oplus {Y_3}^{4^*} = 0$ and ${Y_4}^4 \oplus {Y_4}^{4^*} = 0$ (Number of possible (X^1, X^{1^*}) pairs is $2^{32} \cdot 2^{32}/2 = 2^{63}$ and the probability that the ciphertexts satisfy the above two equations is 2^{-32}. So, there are $2^{63} \cdot 2^{-32} = 2^{31}$ plaintext pairs passing this condition.)

3. For each such (X^1, X^{1^*}) pair;

   - Try all 2^{32} possible values of ${Z_1}^1$ and ${Z_3}^1$, and partially encrypt ${X_1}^1 \rightarrow {Y_1}^1$, ${X_1}^{1^*} \rightarrow {Y_1}^{1^*}$, ${X_3}^1 \rightarrow {Y_3}^1$, ${X_3}^{1^*} \rightarrow {Y_3}^{1^*}$.

Store the 32-bit subkeys which satisfy ${Y_1}^1 \oplus {Y_1}^{1^*} = {Y_3}^1 \oplus {Y_3}^{1^*}$. There are about 2^{16} possible such 32-bit subkeys. This step requires 2^{16} time and memory complexity.

- Try all 2^{32} possible values of ${Z_1}^4$ and ${Z_2}^4$, and partially decrypt ${Y_1}^4 \rightarrow {X_1}^4$, ${Y_1}^{4^*} \rightarrow {X_1}^{4^*}$, ${Y_2}^4 \rightarrow {X_2}^4$, ${Y_2}^{4^*} \rightarrow {X_2}^{4^*}$.
  Store the 32-bit subkeys which satisfy ${X_1}^4 \oplus {X_1}^{4^*} = {X_2}^4 \oplus {X_2}^{4^*}$. There are about 2^{16} possible such 32-bit subkeys. This step requires 2^{16} time and memory complexity.
- Then, there are 2^{32} 64-bit subkeys $({Z_1}^1, {Z_3}^1, {Z_1}^4, {Z_2}^4)$ which satisfy the impossible differential. Hence, they can not be the real subkey values.

4. Repeat this analysis for each of the 2^{31} pairs in each structure and use about 90 structures to eliminate all wrong key values.

5. Analyze the second differential $(0, a, 0, a)$ to find the other key bits. This differential gives 46 new key bits because 16 bits out of 64 are in common with the bits found in the first differential and 2 bits are common between the 1^{st} and the 4^{th} round of the differential. The remaining 18 bits of the 128-bit key can be searched exhaustively.

Data complexity : 90 structures are used, each of which has 2^{31} plaintext pairs $\approx 2^{37.5}$ pairs are used $\approx 2^{38.5}$ plaintexts.
Time complexity : 2^{53} steps of analysis.
Memory complexity : 2^{37} memory.

3.1.4 An Attack on 4-Round IDEA

The attack is applied to IDEA reduced to 4 rounds: $(M \circ T)^8$ between rounds 2 and 5. In this attack, one half-round and two half-rounds are added to the beginning and to the end of the same 2.5-round impossible differential, respectively. In other words, 3.5-round attack was extended to 4 rounds by simply adding one half-round to the end. An illustration of this attack was given in Figure 3.4.
The subkeys used in this attack are :

${Z_1}^2[97, \ldots, 112]$, ${Z_3}^2[26, \ldots, 41]$, ${Z_1}^5[76, \ldots, 91]$,

${Z_2}^5[92, \ldots, 107]$, ${Z_5}^5[12, \ldots, 27]$, ${Z_6}^5[28, \ldots, 43]$.

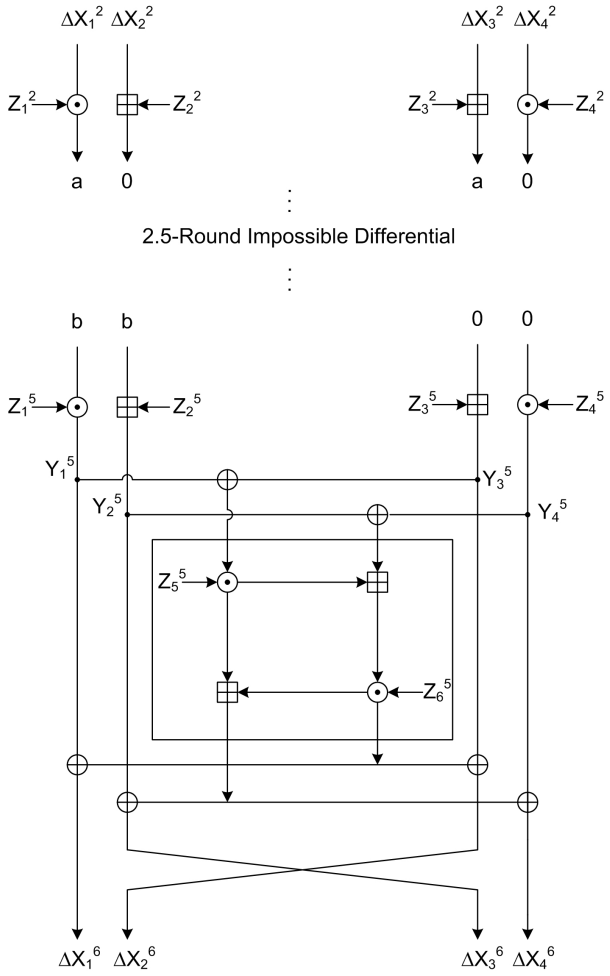

Figure 3.4: 4-Round Impossible Differential of IDEA

Notice that, because of the key schedule of IDEA, there are some common bits between the subkeys. Hence, this attack recovers 69 distinct key bits instead of 96. The approach of the attack is similar to the one described in previous section together with guessing the subkeys used in the last *MA* structure.

Attack Procedure:

1. For each guess of ${Z_5}^5$, ${Z_6}^5$;
   - Decrypt the last half round of all the structures
   - Find all pairs such that ${X_3}^5 \oplus {X_3}^{5^*} = 0$ and ${X_4}^5 \oplus {X_4}^{5^*} = 0$. This condition leaves 2^{31} pairs per structure.
   - For each pair;
     - Calculate the difference $({Z_3}^2 \boxplus {X_3}^2) \oplus ({Z_3}^2 \boxplus {X_3}^{2^*})$. Note that ${Z_3}^2$ is known since some of bits are common with ${Z_5}^5$ and the rest are common with ${Z_6}^5$. Then, find the key ${Z_1}^2$ by using the equation:
     $$({Z_1}^2 \odot {X_1}^2) \oplus ({Z_1}^2 \odot {X_1}^{2^*}) = ({Z_3}^2 \boxplus {X_3}^2) \oplus ({Z_3}^2 \boxplus {X_3}^{2^*})$$
     - ${Z_1}^2$ and ${Z_2}^5$ have 11 common key bits. So, 2^5 choices remain for ${Z_2}^5$. Find ${Z_1}^5$ as in the previous step by using the equation:
     $$({Z_1}^5 \odot {Y_1}^5) \oplus ({Z_1}^5 \odot {Y_1}^{5^*}) = ({Z_2}^5 \boxplus {Y_2}^5) \oplus ({Z_2}^5 \boxplus {Y_2}^{5^*})$$

3.1.5 An Attack on 4.5-Round IDEA

The subkeys used in this attack are :

${Z_5}^1[65, \ldots, 80]$, ${Z_6}^1[81, \ldots, 96]$, ${Z_1}^2[97, \ldots, 112]$, ${Z_3}^2[26, \ldots, 41]$, ${Z_1}^5[76, \ldots, 91]$, ${Z_2}^5[92, \ldots, 107]$, ${Z_2}^5[92, \ldots, 107]$, ${Z_5}^5[12, \ldots, 27]$, ${Z_6}^5[28, \ldots, 43]$.

The attack is as follows:

1. Encrypt all 2^{64} possible plaintexts to get their ciphertexts.
2. Define a structure to be the set of all 2^{32} encryptions in which ${X_2}^2$ and ${X_4}^2$ are fixed values, and ${X_1}^2$ and ${X_3}^2$ take all possible values.
3. Try all possible values of the 80 bits of the subkeys. For each subkey;

- Partially decrypt by one half-round using the keys ${Z_5}^1$ and ${Z_6}^1$ to get the 2^{32} plaintexts.
- For each plaintext, find the corresponding ciphertext, partially decrypt these ciphertexts two half-rounds using the subkeys ${Z_5}^5$, ${Z_6}^5$, ${Z_1}^5$ and ${Z_2}^5$. Partially encrypt all pairs in the structure using the subkeys ${Z_1}^2$ and ${Z_3}^2$.
- Check for the pairs whether the following equations are satisfied: ${Y_1}^2 \oplus {Y_1}^{2^*} = {Y_3}^2 \oplus {Y_3}^{2^*}$, ${X_1}^5 \oplus {X_1}^{5^*} = {X_2}^5 \oplus {X_2}^{5^*}$, ${Y_3}^5 \oplus {Y_3}^{5^*} = 0$, ${Y_4}^5 \oplus {Y_4}^{5^*} = 0$.
- If there is a pair passing previous step, then the 80-bit value of the subkeys is wrong.
- If there are no pairs, continue with another structure.
- Use about 100 structure to get the right key.

4. Find the remaining 48 bits of the key via exhaustive search.

Data complexity : 2^{64} plaintexts.

Time complexity : 2^{112} steps of analysis.

Memory complexity : 2^{32} memory.

3.2 Impossible Differential Cryptanalysis of Reduced-Round AES

In 1997, NIST made an announcement for the competition to select the new encryption standard as a replacement to DES. The block cipher Rijndael which was designed by Rijmen and Daemen became the winner of the competition. In 2002, it was standardized and adopted as the Advanced Encryption Standard (AES). Ever since its selection, it became one of the most widely used block ciphers in the world and inspired too much cryptanalytic attention. A great number of papers have been published on the analysis of AES. However, AES is still considered as a cryptographically secure block cipher.

3.2.1 Description of AES

AES is SPN type symmetric block cipher and operates on bytes. It has a block length of 128 bits and three variable key sizes, namely 128, 192 and 256 bits. Number of rounds varies

according to the key sizes through 10, 12 and 14, respectively. Through the rest of the section, AES will be mentioned according to the size of the key used. For example, AES-128 will refer to AES with 128-bit key size.

Notation:

In this section, ${x_I}^i$ denotes the input of the i'th round and ${x_S}^i$, ${x_R}^i$, ${x_M}^i$, ${x_O}^i$ denote the intermediate values after the operations SubBytes, ShiftRows, MixColumns and AddRoundKey of the i'th round, respectively. It is obvious that ${x_O}^{i-1} = {x_I}^i$.

Let K_i denote the subkey in the i^{th} round, and K_w denote the initial whitening subkey. In some cases, for the sake of simplicity the order of the MixColumns and the AddRoundKey operations in the same round is changed so, the subkey K_i is changed with $W_i = MC^{-1}(K_i)$

Let $(x_i)_{col(k)}$ denote the k^{th} column of x_i, where k=0,1,2,3 and $(x_i)_j$ is the j^{th} byte of x_i (j=0,1,...,15). Here, as seen in Figure 3.6, Column(0) includes bytes 0,1,2,3 and Column(1) includes bytes 4,5,6,7 etc.

Operations:

There are four main operations that are used in the round function:

- **Sub Bytes (SB)** is a non-linear byte substitution which uses an invertible 8×8-bit S-box. It operates on each of the state bytes independently.
- **Shift Rows (SR)** is a linear function that shifts the rows of the state cyclically over different offsets: row i is shifted to the left by i bytes, $0 \leq i \leq 3$.
- **Mix Columns (MC)** is an invertible, linear transformation which combines the four bytes of each column by multiplying columns of the state with a matrix M. This multiplication is done in $GF(2^8)$. The matrix M is given as follows:

$$M = \begin{bmatrix} 0x02 & 0x03 & 0x01 & 0x01 \\ 0x01 & 0x02 & 0x03 & 0x01 \\ 0x01 & 0x01 & 0x02 & 0x03 \\ 0x03 & 0x01 & 0x01 & 0x02 \end{bmatrix}$$

- **Add Round Key (AR)** is a linear operation which combines the round key with the state by using a simple bitwise XOR operation.

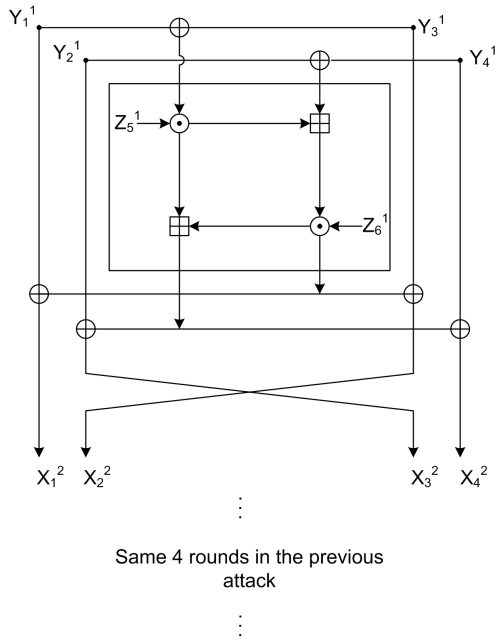

Figure 3.5: 4.5-Round Impossible Differential of IDEA

0	4	8	12
1	5	9	13
2	6	10	14
3	7	11	15

Figure 3.6: 4×4 Byte Indexing of 128-bit AES Data Block

In each round, SB, SR, MC and AR operations are applied in order. Before the first round, there is a key whitening in which the AR operation is applied to the state and in the last round MC operation is excluded to make the encryption and decryption similar. For more details about the round operations, the reader is referred to [30].

Key Schedule:

The AES key schedule generates $R + 1$ round keys from the secret key where R is the number of rounds. Consider the secret key of AES consisting of N 32-bit words. Then, N=4,6,8 for AES-128, AES-192 and AES-256 respectively. The key passes through the key schedule and $4 \times (R + 1)$ expanded key words are produced. Finally, these $4 \times (R + 1)$ key words are used to form the round subkeys, $RK_i's$, by taking 4 words at a time.

The first N words, $W[0], \cdots, W[N - 1]$ are directly initialized by the N words of the secret key and the remaining key words $W[N], \cdots, W[4 \times (R + 1) - 1]$ are generated via the key scheduling algorithm whose details can be found in [36].

3.2.2 Short History of Impossible Differential Attacks on AES

In recent years, several impossible differential attacks on AES have been proposed. In 2000, Biham and Keller presented an attack on 5-round Rijndael using a 4-round impossible differential which is the first impossible differential attack on AES [32]. The attack eliminates wrong keys of the first round by showing that the impossible differential property holds in the last four rounds. In [33], this impossible differential attack was expanded to six rounds by using the same 4-round impossible differential. They put this impossible differential in the middle of six rounds and covered some bits of the first and last round's subkeys. Both in [32] and [33], the attacks were applied to AES-128 and based on the weaknesses which results from the characteristic of the optimal linear layer. These attacks are chosen plaintext attacks and they are independent of the specific choice of S-box, the multiplication polynomial of the MC operation and the key schedule. Therefore, the same attacks in [32] and [33] can also be applied to AES-192 and AES-256. However, in [36], Phan proposed an impossible differential attack on 7-round AES-192 and AES-256 which works by exploiting the weaknesses in the AES key schedule and improves the data and time complexities significantly. In 2007, Zhang et al. presented some new results on impossible differential cryptanalysis of

AES. They introduced new attacks on 6-round AES whose complexity is much lower than that in [33]. Moreover, they extended the attack to both 7-round (which can be applied to all key variants of AES) and 8-round AES-256 and made an improvement of the 7-round attack on AES-192 which was given in [36]. Also again in 2007, Chen et al. presented two methods of impossible differential cryptanalysis of 7-round AES-192 and 8-round AES-256 combined with time-memory trade off by exploiting weaknesses in their key schedule [39]. Complexities of their attacks are slightly better than that of [37]. In 2008, Lu et al. [46] presented a new attack on 7-round AES-128 and AES-192 and two attacks on 8-round AES-256. The attacks on AES-128 and AES-192 are an improvement of the attacks given in [37] and [36] respectively.

In [34], Phan and Siddiqi proved that there exists no impossible differential greater than four rounds that can be constructed with the miss-in-the-middle technique. In all of the above attacks, the same impossible differential property explained in Section 3.2.3 is used.

3.2.3 4-Round Impossible Differentials of AES

4-round impossible differentials are constructed as combining two 2-round differentials with probability 1 in opposite direction where the intermediate differences induce a contradiction.

Theorem 3.2.1 *Given a pair of plaintexts which are equal in all bytes except one, then the ciphertexts after four rounds can not be equal in any combination of the following impossible byte positions: (0,7,10,13), (1,4,11,14), (2,5,8,15) nor (3,6,9,12).*

Proof. Assume that there is a one-byte-difference between the plaintext pairs. This difference is preserved through the SB and SR operations however, it diffuses to one column after MC operation. In the second round, after SR, every column has a one-byte-difference in different byte positions. These one-byte-differences give a data that differs in all bytes after the application of MC operation.

On the other hand, if the ciphertexts are equal in one of the four impossible combinations of bytes, then before SR in the fourth round, data pairs are equal in one column and so are after SR in the third round. When SR^{-1} and SB^{-1} are performed, data pairs are equal in four different bytes. This contradicts with the fact that data differs in all bytes after the second MC.

A 4-round impossible differential of AES is depicted in Figure 3.7.

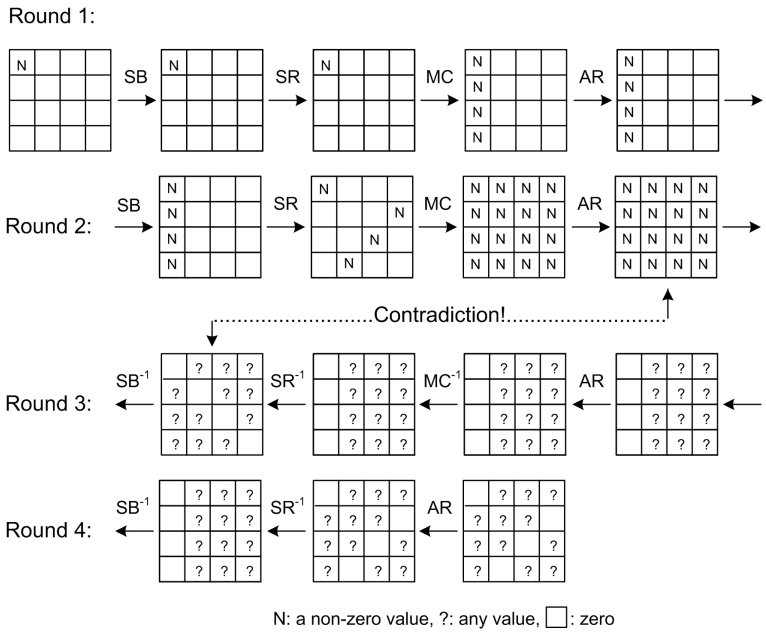

Figure 3.7: A 4-Round Impossible Differential of AES

3.2.4 An Impossible Differential Attack on 6-Round AES

Main idea of this attack is applying the 4-round impossible differential given in Figure 3.7 between the second and the fifth rounds, guessing some key bytes of the first and the last rounds for partial decryption and, then eliminating all wrong keys by using impossible differentials. Illustration of the 6-round attack where the prob. means a probability different from 1, is given in Figure 3.8.

Precomputation phase: For all the 2^{32} possible pairs of $({x_M}^1)_{col(0)}$, $({x^*_M}^1)_{col(0)}$ such that $\Delta({x_M}^1)_{col(0)} \in \{(N,0,0,0),(0,N,0,0),(0,0,N,0),(0,0,0,N)\}$, where N is any nonzero byte:

- Compute the bytes in positions (0,5,10,15) of ${x_I}^1$ and ${x^*_I}^1$
- Store these $2^{32} \times 4 \times (2^8 - 1) \approx 2^{42}$ (there are 2^{32} pairs, 4 column differences for $\Delta({x_M}^1)_{col(0)}$, $(2^8 - 1)$ values for N) pairs of 4-byte values in a hash table H indexed by $\Delta{x_I}^1$ in these four bytes.

One indexed value corresponds to 2^{10} pairs on average.

Attack Procedure:

Structure: A set of 2^{32} plaintexts which have all different values in bytes (0,5,10,15).

1. Generate n such structures.

2. Choose plaintext pairs whose ciphertext pairs have nonzero difference in the two bytes (3,6) and zero difference in all other bytes.

3. Guess the value of the subkey bytes $(K_6)_3$, $(K_6)_6$ and

   (a) Make a list L of all possible values of the bytes (0,5,10,15) of K_0.

   (b) Partially decrypt the bytes (3,6) of the ciphertext pairs in order to get the corresponding bytes of the fifth round outputs, ${x_O}^5$ and ${x_O^*}^5$.

   (c) Calculate the difference in the last column of ${x_O}^5$ through MC^{-1} operation. Check whether the difference in the four bytes of the last column are all nonzero. If so, discard the pairs. Remaining pairs satisfy the impossible differential.

   (d) For every remaining ciphertext pair, consider their plaintexts (P_1, P_2) and compute $\Delta P = P_1 \oplus P_2$. Access the bin ΔP in H. For each pair (x, y) in that bin, delete the values $P_1 \oplus x$ from the list L.

   (e) If L is not empty, output the values in L along with the guess of $(K_6)_3$, $(K_6)_6$.

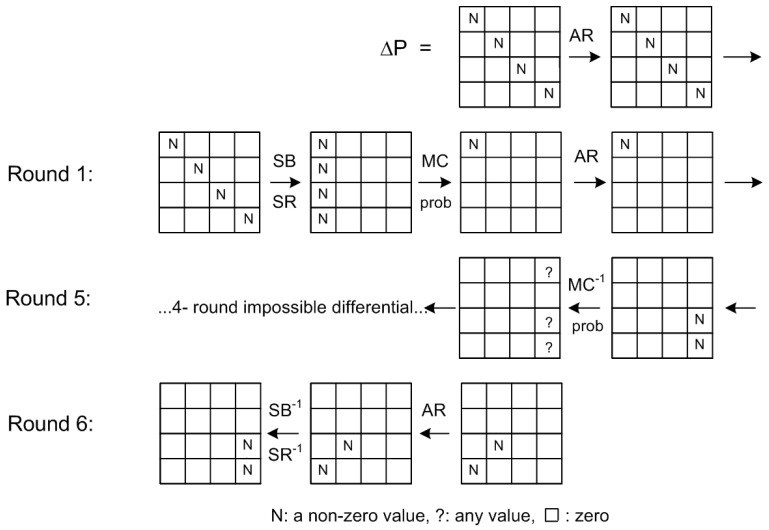

Figure 3.8: Impossible Differential of 6-Round AES

Complexity Analysis

- **Data Complexity:**
  - There are 2^{32} plaintexts in 1 structure. So, $\frac{2^{32}\cdot 2^{32}}{2} = 2^{63}$ pairs can be derived. As a result, n structures constitute $n \cdot 2^{63}$ plaintext pairs.
  - In step 2, the probability of obtaining such ciphertext pairs is $2^{8\cdot 14} = 2^{-112}$ since there are 14 zero bytes. Therefore, $2^{63}n \cdot 2^{-112} = 2^{-49}n$ pairs remain after step 2.
  - The probability that a pair passes step 3.*c* is about $4 \cdot 2^{-8} = 2^{-6}$ since there are 4 positions for one byte zero difference. So, $2^{-49}n \cdot 2^{-6} = 2^{-55}n = n'$ pairs remain after step 3.*c*.
  - In step 3.*d*, each pair deletes 2^{10} subkey candidates K_0 on average, and there are 2^{32} subkey candidates for K_0 in all. Hence, in order to have remaining wrong subkey candidates for K_0 is less than 1, so that only the right key remains,

$$\begin{aligned} 2^{32} \cdot (1 - \frac{2^{10}}{2^{32}})^{n'} &< 1 \\ 2^{32} \cdot e^{-\frac{2^{10}}{2^{32}} \cdot n'} &< 1 \\ -\frac{n'}{2^{22}} \cdot &< -\ln 2^{32} \\ n' &> \ln 2^{32} \cdot 2^{22} \approx 2^{26.5} \end{aligned} \tag{3.3}$$

    n' should be chosen greater than $2^{26.5}$.

If $n' = 2^{27.5}$ then $2^{32} \cdot (1 - \frac{2^{10}}{2^{32}})^{n'} \approx 2^{-33}$ and probability of getting the wrong value of subkeys $(K_0)_{0,5,10,15}|(K_6)_3|(K_6)_6$ will be $2^{-33} \cdot 2^{16} = 2^{-17}$ which is very small. So, in order to obtain the right subkey value, $n = 2^{82.5}$ structures are needed. Accordingly, this makes the data complexity of this attack $n = 2^{82.5} \cdot 2^{32} = 2^{114.5}$ chosen plaintexts.

- **Time Complexity:**
  - Precomputation stage requires 2^{32} 1-round encryptions which is equivalent to $\frac{2^{32}}{6} \approx 2^{29.5}$ 6-round encryptions.
  - Step 3.*b* requires $2^{33.5} \cdot 2 \cdot 2^{16} \cdot \frac{1}{4} = 2^{48.5}$ 1-round encryptions since there are $2^{33.5}$ ciphertext pairs, 2^{16} (two bytes) key guesses. $\frac{1}{4}$ comes from the partial decryption of four bytes.

- Step 3.d requires $2^{27.5} \cdot 2^{10} \cdot 2^{16} = 2^{53.5}$ memory accesses to H since there are $2^{27.5}$ ciphertext pairs after step 3.c and, $2^{53.5}$ memory accesses is equivalent to 2^{50} 6-round encryptions.

Therefore, time complexity is 2^{50} encryptions.

- **Memory Complexity:**
  - Required memory is 2^{45} bytes due to the hash table H.

In this attack, a data-time trade off can be done by guessing more bytes of subkey K_6 so that after partial decryption the number of nonzero bytes in the output of the fifth round reach the most possible. This trade-off reduces the data complexity to $2^{75.5}$ chosen plaintexts and increases the time complexity to 2^{104} encryptions.

3.2.5 Extending 6-Round Attack to 7 Rounds

The above attack can be improved to attack 7-round AES. The main idea is to guess some bytes of the last round subkey K_7, decrypt the last round and apply the 6-round attack as described above. In this extension, different from the 6-round attack, the order of MC and AR in the fifth and sixth rounds is changed in order to guess less key material and the subkeys K_5 and K_6 are replaced with equivalent subkeys.

Data complexity of this attack is $2^{115.5}$ chosen plaintexts, time complexity is 2^{119} encryptions, and the required memory is 2^{45} bytes.

Note that, these two attacks are applicable to all key variants of AES and can be improved to attack 8-round AES-256.

3.3 Impossible Differential Cryptanalysis of CLEFIA

In this section, impossible differential cryptanalysis of CLEFIA is analyzed. CLEFIA is a 128-bit block cipher designed by Shirai et al. [47], Sony Corporation, in 2007. There are not so many impossible differential attacks on CLEFIA. First attacks are proposed by its designers [47, 48] which can be applied on 10 rounds of CLEFIA with all key lengths, on 11 rounds with

Table 3.2: Comparison of Attack Complexities

CP: Chosen Plaintext, MA:Memory Access

AES	Paper	Round	Data (CP)	Time	Memory
128	[32]	5	$2^{29.5}$	2^{31}	2^{42}
	[33]	6	$2^{91.5}$	2^{122}	2^{89}
	[37]	6	$2^{114.5}$	2^{50}	2^{45}
	[38]	7	$2^{115.5}$	2^{119}	2^{109}
	[37]	7	$2^{115.5}$	2^{119}	2^{45}
	[46]	7	$2^{112.2}$	$2^{117.2}$ MA	-
192	[36]	7	2^{92}	2^{186}	2^{153}
	[37]	7	2^{92}	2^{162}	-
	[39]	7	$2^{94.5}$	2^{157}	2^{129}
	[46]	7	$2^{113.8}$	$2^{118.8}$ MA	-
	[46]	7	$2^{91.2}$	$2^{139.2}$	-
256	[36]	7	$2^{92.5}$	$2^{250.5}$	2^{153}
	[46]	7	$2^{113.8}$	$2^{118.8}$ MA	-
	[46]	7	2^{92}	2^{163} MA	-
	[37]	8	$2^{166.5}$	$2^{247.5}$	-
	[39]	8	2^{101}	2^{228}	2^{201}
	[46]	8	$2^{111.1}$	$2^{227.8}$ MA	-
	[46]	8	$2^{89.1}$	$2^{229.7}$ MA	-

192 and 256-bit key lengths and on 12 rounds with only 256-bit key length. Later, in 2008, Tsunoo et al. [50] presented new attacks on 12 rounds (for 128, 192, 256-bit key length), 13 rounds (for 192, 256-bit key length) and 14 rounds (for 256-bit key length). Within the issue of impossible differential cryptanalysis of CLEFIA, the attacks given in [50] were described in this section.

3.3.1 Description of CLEFIA

CLEFIA is a block cipher having a four-branch generalized Feistel structure. It has a block length of 128 bits and key lengths of 128, 192 and 256 bits. According to these key bits, number of rounds varies through 18, 22 and 26, respectively.

There are two parallel F functions, F_0 and F_1, per round and for r-round CLEFIA, $2r$ 32-bit subkeys $(RK_0, \ldots, RK_{2r-1})$ are employed. Also, there are four 32-bit whitening keys $(WK_0, \ldots, WK_3)$ two of which are used in the first round and the other two are used in the last round.

Encryption process for r-round CLEFIA is depicted in Figure 3.9 and is defined as follows:

1. $T_0|T_1|T_2|T_3 \leftarrow {X_0}^0|{X_1}^0 \oplus WK_0|{X_2}^0|{X_3}^0 \oplus WK_1|$

2. For i=0 to $r-1$ do the following:

   (a) $T_1 \leftarrow T_1 \oplus F_0(RK_{2i}, T_0), T_3 \leftarrow T_3 \oplus F_1(RK_{2i+1}, T_2)$

   (b) $T_0|T_1|T_2|T_3 \leftarrow T_1|T_2|T_3|T_0$

3. ${X_0}^r|{X_1}^r|{X_2}^r|{X_3}^r \leftarrow T_3|T_0 \oplus WK_2|T_1|T_2 \oplus WK_3$

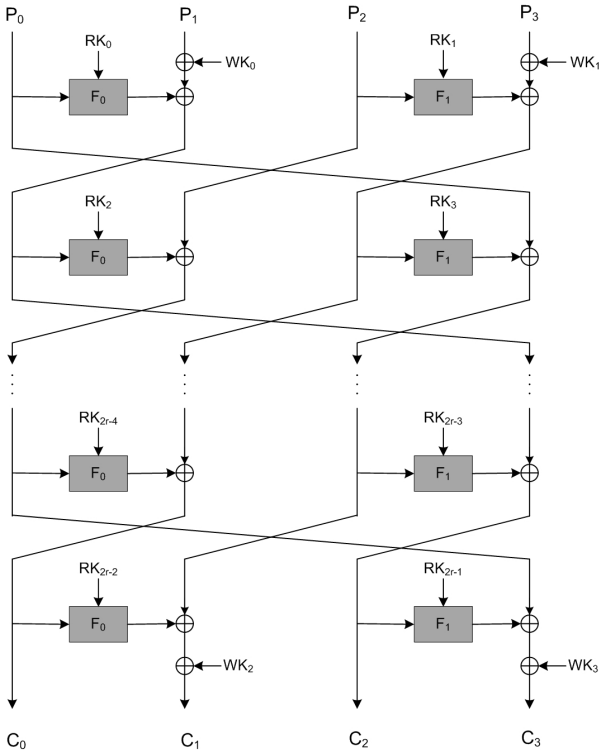

Figure 3.9: Encryption Process of r-round CLEFIA

F functions $F_0, F_1 : (RK, x) \mapsto y$ are defined as follows:

F-function F_0	F-function F_1
$T \leftarrow RK \oplus x$	$T \leftarrow RK \oplus x$
Let $T = T_0\|T_1\|T_2\|T_3, T_i \in \{0,1\}^8$	*Let* $T = T_0\|T_1\|T_2\|T_3, T_i \in \{0,1\}^8$
$T_0 \leftarrow S_0(T_0), T_1 \leftarrow S_1(T_1)$	$T_0 \leftarrow S_1(T_0), T_1 \leftarrow S_0(T_1)$
$T_2 \leftarrow S_0(T_2), T_3 \leftarrow S_1(T_3)$	$T_2 \leftarrow S_1(T_2), T_3 \leftarrow S_0(T_3)$
Let $y = y_0\|y_1\|y_2\|y_3, y_i \in \{0,1\}^8$	*Let* $y = y_0\|y_1\|y_2\|y_3, y_i \in \{0,1\}^8$
${}^t(y_0, y_1, y_2, y_3) = M_0^t(T_0, T_1, T_2, T_3)$	${}^t(y_0, y_1, y_2, y_3) = M_1^t(T_0, T_1, T_2, T_3)$

Figure 3.10 shows the F-functions:

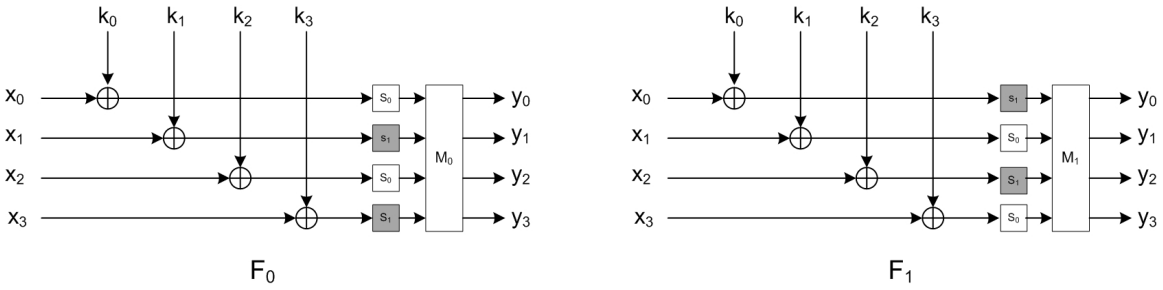

Figure 3.10: The F-Functions F_0 and F_1

The matrices M_0 and M_1 are defined as:

$$M_0 = \begin{pmatrix} 0x01 & 0x02 & 0x04 & 0x06 \\ 0x02 & 0x01 & 0x06 & 0x04 \\ 0x04 & 0x06 & 0x01 & 0x02 \\ 0x06 & 0x04 & 0x02 & 0x01 \end{pmatrix}, M_1 = \begin{pmatrix} 0x01 & 0x08 & 0x02 & 0x0a \\ 0x08 & 0x01 & 0x0a & 0x02 \\ 0x02 & 0x0a & 0x01 & 0x08 \\ 0x0a & 0x02 & 0x08 & 0x01 \end{pmatrix}$$

S_0 and S_1 are non-linear 8-bit S-boxes. Tables of S-boxes and key scheduling algorithm can be found in [47].

3.3.2 9-Round Impossible Differentials of CLEFIA

In this section, 9-round impossible differentials of CLEFIA are described. The following two impossible differentials are proposed in [47, 48] :

- $(0, \alpha, 0, 0) \overset{9r}{\nrightarrow} (0, \alpha, 0, 0)$ and $(0, 0, 0, \alpha) \overset{9r}{\nrightarrow} (0, 0, 0, \alpha)$ with probability 1, where $\alpha \in \{0, 1\}^{32}$ is a nonzero value.

However, Tsunoo et al. [50] found new 9-round impossible differentials which yield better cryptanalytic results in the sense of reducing time complexity and attacking more rounds of the cipher:

- $(0, \alpha_{in}, 0, 0) \overset{9r}{\nrightarrow} (0, \alpha_{out}, 0, 0)$ and $(0, 0, 0, \alpha_{in}) \overset{9r}{\nrightarrow} (0, 0, 0, \alpha_{out})$ with probability 1.

Different from the ones given in [47, 48], α_{in} and α_{out} do not have to be equal values in these new differentials. Differences that α_{in} and α_{out} can take is shown in Table 3.3 in which X and Y are nonzero 8-bit values.

Table 3.3: Differences for α_{in} and α_{out}

α_{in}	α_{out}
$(0,0,0,X)$	$(0,0,Y,0),(0,Y,0,0),(Y,0,0,0)$
$(0,0,X,0)$	$(0,0,0,Y),(0,Y,0,0),(Y,0,0,0)$
$(0,X,0,0)$	$(0,0,0,Y),(0,0,Y,0),(Y,0,0,0)$
$(X,0,0,0)$	$(0,0,0,Y),(0,0,Y,0),(0,Y,0,0)$

Below, 9-round impossible differential with probability 1 is explained:

Theorem 3.3.1 *The input difference* $(0,0,0,\alpha_{in})$ *cannot produce the output difference* $(0,0,0,\alpha_{out})$ *after 9 rounds of CLEFIA, where* $\alpha_{in} = (0,0,0,X)$ *and* $\alpha_{out} = (0,Y,0,0)$.

Proof. Assume that the input difference ${\Delta X_0}^4$ to the fifth round's F_0 when the input difference is $(0,0,0,\alpha_{in})$, is equal to the input difference ${\Delta X'_0}^4$ to the fifth round's F_0 when the output difference is $(0,0,0,\alpha_{out})$.

Let $S_1(X) = X'$ and $S_0(X) = X''$

$$\begin{aligned} {\Delta X_0}^4 &= {M_0}^t(0,0,0,X') \oplus {M_1}^t(0,0,0,X'') \\ &= (M_0|M_1)^t(0,0,0,X',0,0,0,X'') \end{aligned} \tag{3.4}$$

Let $S_1(Y) = Y'$ and $S_0(Y) = Y''$

$$\begin{aligned} {\Delta X'_0}^4 &= {M_0}^t(0,Y',0,0) \oplus {M_1}^t(0,Y'',0,0) \\ &= (M_0|M_1)^t(0,Y',0,0,0,Y'',0,0) \end{aligned} \tag{3.5}$$

Then, from Equation 3.4 and 3.5,

$$\begin{aligned}
\Delta {X_0}^4 \oplus \Delta {X'_0}^4 &= (M_0|M_1)^t(0,0,0,X',0,0,0,X'') \oplus (M_0|M_1)^t(0,Y',0,0,0,Y'',0,0) \\
&= (M_0|M_1)^t\left[(0,0,0,X',0,0,0,X'') \oplus (0,Y',0,0,0,Y'',0,0)\right] \\
&= (M_0|M_1)^t(0,Y',0,X',0,Y'',0,X'') \\
&= {}^t(0,0,0,0) \qquad (3.6)
\end{aligned}$$

Branch number for a function P is defined as $B(P) = min_{a\neq 0}\{w_b(a) + w_b(P(a))\}$, where $w_b(a)$ denotes the number of nonzero a_i's for an $8n$-bit string $a = a_0|a_1|\ldots|a_{n-1}$, $a_i \in \{0,1\}^8$. As specified in the proposal of CLEFIA, the branch number for M_0, M_1 and for the concatenation matrix $M_0|M_1$ is 5.

$$\begin{aligned}
B(M_0|M_1) &= min\left\{w_b((0,Y',0,X',0,Y'',0,X'')) + w_b((M_0|M_1)^t(0,Y',0,X',0,Y'',0,X''))\right\} \\
&= 5
\end{aligned}$$

Hence, $w_b((0,Y',0,X',0,Y'',0,X'')) + w_b((M_0|M_1)^t(0,Y',0,X',0,Y'',0,X'')) \geq 5$
$w_b((0,Y',0,X',0,Y'',0,X'')) = 4$, then $w_b((M_0|M_1)^t(0,Y',0,X',0,Y'',0,X'')) \geq 1$

Equation 3.6 gives $(M_0|M_1)^t(0,Y',0,X',0,Y'',0,X'') = {}^t(0,0,0,0)$. Then, weight of both sides of this equation must be equal. However, $w_b((M_0|M_1)^t(0,Y',0,X',0,Y'',0,X'')) \geq 1$ whereas $w_b((0,0,0,0))$=0 which leads to a contradiction.

Therefore, $\Delta {X_0}^4$ cannot be equal to $\Delta {X'_0}^4$ and the input difference (0, 0, 0, (0, 0, 0, X)) can not cause an output difference (0, 0, 0, (0, Y, 0, 0)) after 9 rounds of CLEFIA. ■

3.3.3 Another 9-Round Impossible Differential of CLEFIA

Sun et al. [51] found a new 9-round impossible differential, improved the previous attacks and proceeded the attack up to 15 rounds by using this differential.

Theorem 3.3.2 *Given in the following tables, the input difference α_{in} cannot produce the output difference α_{out} after 9 rounds of CLEFIA where x and y denote nonzero differences and z denotes any difference.*

α_{in}	α_{out}
$(0, 000\mathbf{x}, 0, 0)$	$(0, 00\mathbf{y}z, 0, 0), (0, 0\mathbf{y}0z, 0, 0), (0, \mathbf{y}00z, 0, 0)$
$(0, 00\mathbf{x}0, 0, 0)$	$(0, 0\mathbf{y}z0, 0, 0), (0, \mathbf{y}0z0, 0, 0), (0, 00z\mathbf{y}, 0, 0)$
$(0, 0\mathbf{x}00, 0, 0)$	$(0, \mathbf{y}z00, 0, 0), (0, 0z0\mathbf{y}, 0, 0), (0, 0z\mathbf{y}0, 0, 0)$
$(0, \mathbf{x}000, 0, 0)$	$(0, z00\mathbf{y}, 0, 0), (0, z0\mathbf{y}0, 0, 0), (0, z\mathbf{y}00, 0, 0)$
$(0, 0, 0, 000\mathbf{x})$	$(0, 0, 0, 00\mathbf{y}z), (0, 0, 0, 0\mathbf{y}0z), (0, 0, 0, \mathbf{y}00z)$
$(0, 0, 0, 00\mathbf{x}0)$	$(0, 0, 0, 0\mathbf{y}z0), (0, 0, 0, \mathbf{y}0z0), (0, 0, 0, 00z\mathbf{y})$
$(0, 0, 0, 0\mathbf{x}00)$	$(0, 0, 0, \mathbf{y}z00), (0, 0, 0, 0z0\mathbf{y}), (0, 0, 0, 0z\mathbf{y}0)$
$(0, 0, 0, \mathbf{x}000)$	$(0, 0, 0, z00\mathbf{y}), (0, 0, 0, z0\mathbf{y}0), (0, 0, 0, z\mathbf{y}00)$

α_{in}	α_{out}
$(0, 00\mathbf{y}z, 0, 0), (0, 0\mathbf{y}0z, 0, 0), (0, \mathbf{y}00z, 0, 0)$	$(0, 000\mathbf{x}, 0, 0)$
$(0, 0\mathbf{y}z0, 0, 0), (0, \mathbf{y}0z0, 0, 0), (0, 00z\mathbf{y}, 0, 0)$	$(0, 00\mathbf{x}0, 0, 0)$
$(0, \mathbf{y}z00, 0, 0), (0, 0z0\mathbf{y}, 0, 0), (0, 0z\mathbf{y}0, 0, 0)$	$(0, 0\mathbf{x}00, 0, 0)$
$(0, z00\mathbf{y}, 0, 0), (0, z0\mathbf{y}0, 0, 0), (0, z\mathbf{y}00, 0, 0)$	$(0, \mathbf{x}000, 0, 0)$
$(0, 0, 0, 00\mathbf{y}z), (0, 0, 0, 0\mathbf{y}0z), (0, 0, 0, \mathbf{y}00z)$	$(0, 0, 0, 000\mathbf{x})$
$(0, 0, 0, 0\mathbf{y}z0), (0, 0, 0, \mathbf{y}0z0), (0, 0, 0, 00z\mathbf{y})$	$(0, 0, 0, 00\mathbf{x}0)$
$(0, 0, 0, \mathbf{y}z00), (0, 0, 0, 0z0\mathbf{y}), (0, 0, 0, 0z\mathbf{y}0)$	$(0, 0, 0, 0\mathbf{x}00)$
$(0, 0, 0, z00\mathbf{y}), (0, 0, 0, z0\mathbf{y}0), (0, 0, 0, z\mathbf{y}00)$	$(0, 0, 0, \mathbf{x}000)$

Proof. Without loss of generality, the case $(0, 000x, 0, 0) \overset{9r}{\nrightarrow} (0, 0y0z, 0, 0)$ where $x \neq 0$ and $y \neq 0$, is proved below.

After the fourth round, the difference ${\Delta X_3}^4$ can be written in terms of the second round F_0 and fourth round F_1:
Let $a = S_1(\beta \oplus x) \oplus S_1(\beta)$, $b = S_0(\delta \oplus x) \oplus S_0(\delta)$ for some $x, \beta, \delta \in F_{2^8}$. Also, $a \neq 0$, $b \neq 0$ since $x \neq 0$ and S_0, S_1 are bijective maps over F_{2^8}. Then,

$$ {\Delta X_3}^4 = M_0(0, 0, 0, a) \oplus M_1(0, 0, 0, b) $$

In the same way, the difference ${\Delta X_1}^6$ can be written in terms of the eighth round F_1 and seventh round F_0:

$$ {\Delta X_1}^6 = M_0(0, e, 0, f) \oplus M_1(0, c, 0, d) \text{ where } e \neq 0 \text{ and } c \neq 0. $$

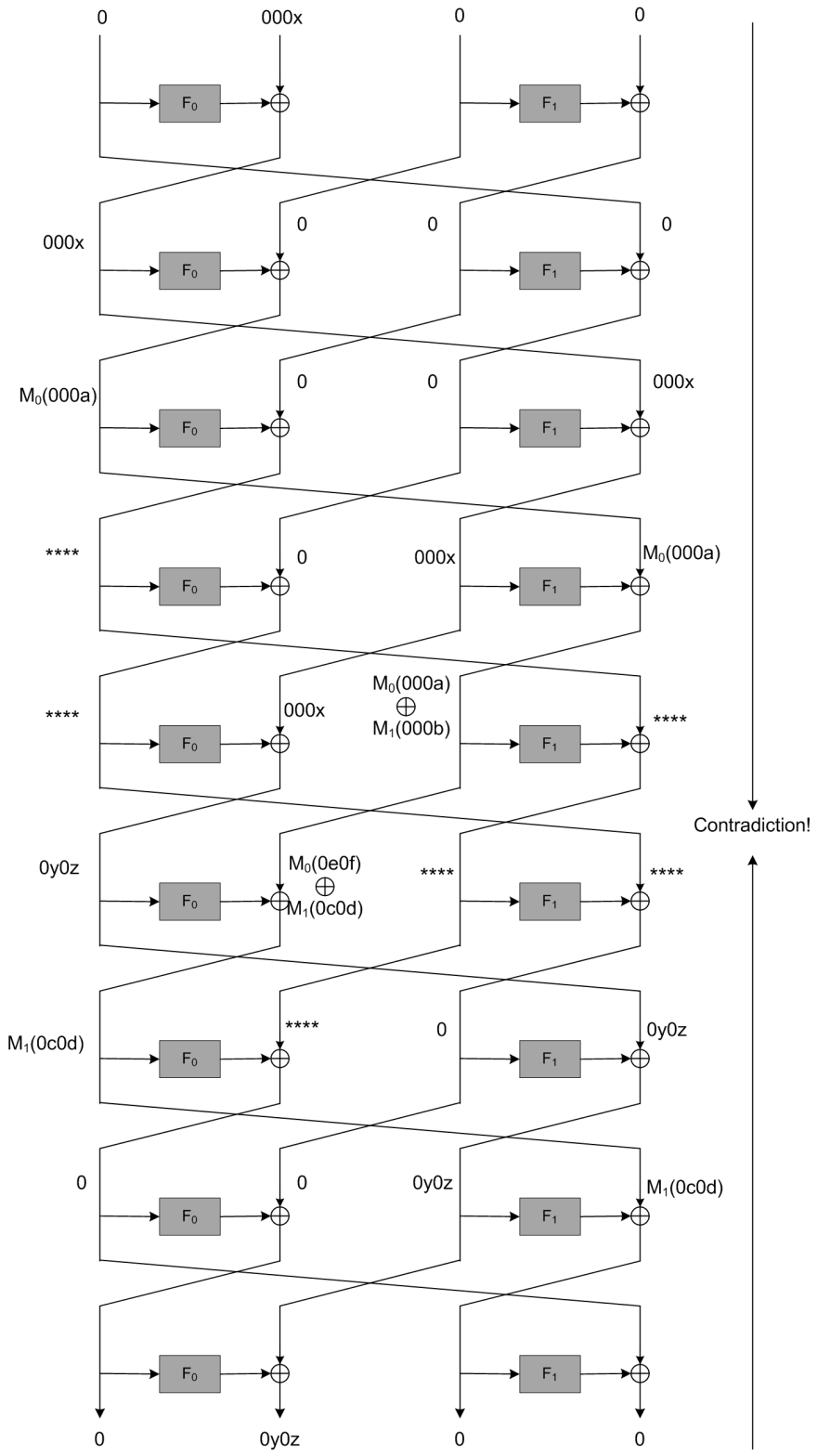

Figure 3.11: 9-Round Impossible Differential of CLEFIA

As can be seen from the Figure 3.11:

$$\begin{aligned}
{\Delta X_3}^4 &= {\Delta X_1}^6 \\
M_0(0,0,0,a) \oplus M_1(0,0,0,b) &= M_0(0,e,0,f) \oplus M_1(0,c,0,d) \\
M_1(0,c,0,b \oplus d) &= M_0(0,e,0,a \oplus f) \\
{M_0}^{-1} M_1(0,c,0,b \oplus d) &= (0,e,0,a \oplus f) \\
\begin{pmatrix} m_{0,1} & m_{0,3} \\ m_{2,1} & m_{2,3} \end{pmatrix} \begin{pmatrix} c \\ b \oplus d \end{pmatrix} &= \begin{pmatrix} 0 \\ 0 \end{pmatrix}
\end{aligned} \tag{3.7}$$

Proposition 3.3.3 *Let $M = {M_0}^{-1} M_1 = (m_{ij})$, $0 \le i \le 3, 0 \le j \le 3$, where M_0 and M_1 are defined as in CLEFIA. Then*

$$\begin{vmatrix} m_{i_1,j_1} & m_{i_1,j_2} \\ m_{i_2,j_1} & m_{i_2,j_2} \end{vmatrix} \neq 0$$

for $0 \le i_1 \le i_2 \le 3$ and $0 \le j_1 \le j_2 \le 3$.

Proof. Details of the proof can be found in [51].

As a result of Proposition 3.3.3, $\begin{vmatrix} m_{0,1} & m_{0,3} \\ m_{2,1} & m_{2,3} \end{vmatrix} \neq 0$, hence Equation 3.7 has only zero solution. Therefore, $c = 0$ which contradicts with the fact that c is nonzero. ■

Note that the case $z = 0$ reduces these differentials to the ones given in [50].

Before explaining the attacks mounted on CLEFIA, it's worth noting some observations given in [49].

Proposition 3.3.4 *Let (In, In') be two 32-bit inputs for the F function (F_0 or F_1) and ΔOut be the difference of the corresponding outputs, then 32-bit round subkey RK involved in F can be deduced with about one F-computation.*

Proposition 3.3.5 *For r-round CLEFIA, let (RK_{2r-3}, RK_{2r-4}) be the subkey in the $(r-1)^{th}$ round, (RK_{2r-1}, RK_{2r-2}) be the subkey in the r^{th} round, (WK_2, WK_3) be the whitening key in*

the final round, and $C^r = (C_0^r, C_1^r, C_2^r, C_3^r)$ *be the ciphertext, the following two equations hold:*

$$\begin{aligned} WK_3 \oplus RK_{2r-4} &= InS_{F_0}^{r-1} \oplus F_1^r(C_2^r, RK_{2r-1}) \oplus C_3^r, \\ WK_2 \oplus RK_{2r-3} &= InS_{F_1}^{r-1} \oplus F_0^r(C_0^r, RK_{2r-2}) \oplus C_1^r \end{aligned}$$

where $InS_{F_0}^{r-1}$ *and* $InS_{F_1}^{r-1}$ *are the inputs to the four S-boxes of* ${F_0}^{r-1}$ *and* ${F_1}^{r-1}$*, respectively.*

3.3.4 An Attack on 12-Round CLEFIA

In this section, an 12-round impossible differential attack by using 9-round impossible differential explained in Section 3.3.3 is described. This 12-round characteristic is constructed by adding one round to the beginning and two rounds to the end of the 9-round differential. 12-round attack is depicted in Figure 3.12.

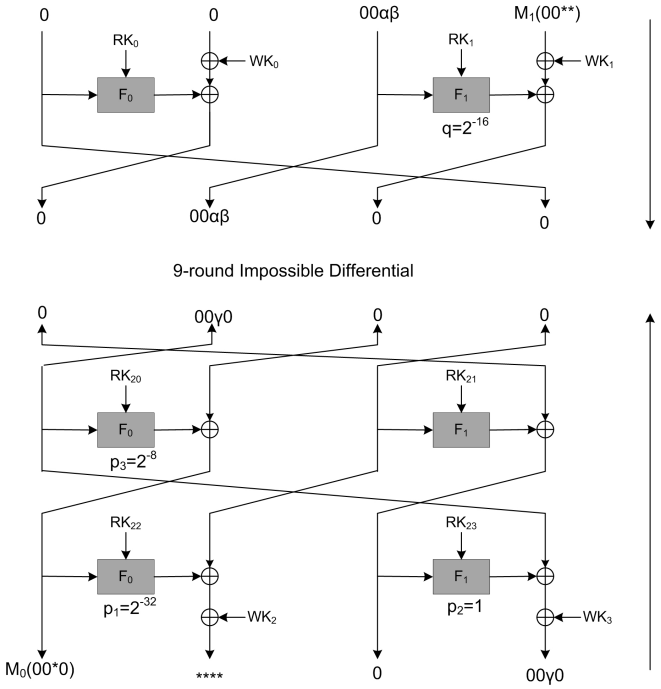

Figure 3.12: 12-Round Impossible Differential Attack on CLEFIA

Structure: Let $\Omega = \{P_0, P_1, P_2 \oplus (00**), P_3 \oplus M_1(00**)\}$ where P_0, P_1, P_2, P_3 are constant values and * is unknown nonzero byte difference. Then, there are $(2^8 - 1)^4 \approx 2^{32}$ elements in Ω. This means that there are $\dfrac{2^{32} \cdot 2^{32}}{2} = 2^{63}$ plaintext pairs (P, P^*) having a difference $(0, 0, 00**, M_1(00**))$.

1. Take $2^{78.93}$ structures which gives $2^{110.93}$ plaintexts, $2^{141.93}$ plaintext pairs.

2. Choose plaintext pairs whose ciphertext pairs (C, C^*) has a difference $(M_0(00*0), **$ $**, 0, 00*0)$. The probability of obtaining such ciphertext pairs is $\frac{2^8-1}{2^{32}} \cdot \frac{2^{32}-1}{2^{32}} \cdot \frac{1}{2^{32}} \cdot \frac{2^8-1}{2^{32}} \approx 2^{-80}$. Then, the expected number of such ciphertext pairs is $2^{141.93} \cdot 2^{-80} = 2^{61.93}$.

3. For every remaining pair (C, C^*) and its corresponding plaintext pair (P, P^*), guess RK_{23} and find $RK_{22}|(WK_3 \oplus RK_{20})_2|(RK_1)_{2,3}$ by using differential table look-ups according to Proposition 3.3.4. The probability of knowing that a $RK_{22}|(WK_3 \oplus RK_{20})_2|(RK_1)_{2,3}$ candidate is wrong by using the differential table for the three F's is 2^{-56}. This probability comes from the average of 2^{-32} for the 12^{th}-round F_0, the average of 2^{-8} for the 11^{th}-round F_0, and the average of 2^{-16} for the 1^{st}-round F_1. Therefore, the number of ciphertext pairs, N, required to narrow the 88-bit key $RK_{23}|RK_{22}|(WK_3 \oplus RK_{20})_2|(RK_1)_{2,3}$ down to the correct key is about $2^{61.93}$ from the equation

$$2^{88}(1 - 2^{-56})^N \approx 1.$$

Time complexity of the attack:

- $2^{110.93}$ encryptions for obtaining the ciphertexts,
- $\leq 2^{32}N = 2^{93.93}$ F function computations.
  (2^{32} RK_{23} guesses, $2^{61.93}$ ciphertext pairs)

Therefore, time complexity of the attack is 2^{111} encryptions.

Data complexity of the attack is $2^{110.93}$ chosen plaintexts.

3.3.5 Extending 12-Round Attack to 13-Round

12-round attack explained in the previous section can be extended to 13 round by adding one more round to the beginning of the 12-round characteristic.

Structure: Let $\Omega = \{P_0 \oplus M_1(00**), P_1 \oplus (****), P_2, P_3 \oplus (00**)\}$ where P_0, P_1, P_2, P_3 are constant values and * is unknown nonzero byte difference. Then, there are $(2^8 - 1)^8 \approx$

2^{64} elements in Ω giving $\frac{2^{63} \cdot 2^{63}}{2} = 2^{127}$ plaintext pairs (P, P^*) which have a difference of $(M_1(00**), ****, 0, (00**))$.

1. Take $2^{47.72}$ structures which gives $2^{111.72}$ plaintexts, $2^{174.72}$ plaintext pairs.

2. Choose plaintext pairs whose ciphertext pairs (C, C^*) has a difference $(M_0(00*0), **$ $**, 0, 00*0)$. The probability of obtaining such ciphertext pairs is again 2^{-80}. Then, the expected number of such ciphertext pair is $2^{174.72} \cdot 2^{-80} = 2^{61.93}$.

3. For every remaining pair (C, C^*) and its corresponding plaintext pair (P, P^*), guess $RK_{25}|RK_1$ and find $RK_0|(WK_1 \oplus RK_2)_{2,3}|RK_{24}|(WK_3 \oplus RK_{22})_3$ by using differential table look-ups according to Proposition 3.3.4. The probability of knowing that a $RK_0|(WK_1 \oplus RK_2)_{2,3}|RK_{24}|(WK_3 \oplus RK_{22})_3$ candidate is wrong by using the differential table for the four F's is 2^{-88}. Therefore, the number of ciphertext pairs, N, required to narrow the 152-bit key $RK_{25}|RK_{21}|RK_0|(WK_1 \oplus RK_2)_{2,3}|RK_{24}|(WK_3 \oplus RK_{22})_3$ down to the correct key is about $2^{94.72}$ from the equation

$$2^{152}(1 - 2^{-88})^N \approx 1.$$

Time complexity of the attack:

- $2^{111.72}$ encryptions for obtaining the ciphertexts,
- $\leq 2^{64}N = 2^{158.72}$ F function computations for reducing the key candidates (2^{64} $RK_{25}|RK_1$ guesses, $2^{94.72}$ ciphertext pairs)

Therefore, time complexity of the attack is $\leq 2^{158}$ encryptions.

Data complexity of this attack is $2^{111.72}$ chosen plaintexts.

3.3.6 Extending 13-Round Attack to 14-Round

13-round attack can be extended to 14 round by adding one more round to the end of the 13-round characteristic.

Structure: Let $\Omega = \{P_0 \oplus M_1(00**), P_1 \oplus (****), P_2, P_3 \oplus (00**)\}$ where P_0, P_1, P_2, P_3 are constant values and * is unknown nonzero byte difference. Then, there are $(2^8 - 1)^8 \approx 2^{64}$

elements in Ω. This means that there are $\frac{2^{64} \cdot 2^{64}}{2} = 2^{127}$ plaintext pairs (P, P^*) having a difference $(M_1(00**), ****, 0, (00**))$.

1. Take $2^{48.23}$ structures which gives $2^{112.23}$ plaintexts, $2^{175.23}$ plaintext pairs.

2. Choose plaintext pairs whose ciphertext pairs (C, C^*) has a difference $(****, ****, 00*0, M_0(00*0) \oplus M_1(00*0))$. The probability of obtaining such ciphertext pairs is $\frac{2^{32}-1}{2^{32}} \cdot \frac{2^{32}-1}{2^{32}} \cdot \frac{2^8-1}{2^{32}} \cdot \frac{(2^8-1)\cdot(2^8-1)}{2^{32}} \approx 2^{-40}$. Then, the expected number of such ciphertext pair is $2^{175.23} \cdot 2^{-40} = 2^{135.23}$.

3. For every remaining pair (C, C^*) and its corresponding plaintext pair (P, P^*), guess $RK_1|(RK_{24} \oplus WK_3)$ and find $RK_{26}|RK_{27}|(WK_2 \oplus RK_{25})|(RK_{22})_2|RK_0|(WK_1 \oplus RK_3)_{2,3}$ by using differential table look-ups. The probability of knowing that a candidate is wrong by using the differential tables is 2^{-128}. Therefore, the number of ciphertext pairs, N, required to narrow the 216-bit key down to the correct key is about $2^{135.23}$ from the equation

$$2^{216}(1 - 2^{-128})^N \approx 1.$$

Time complexity of the attack:

- $2^{112.23}$ encryptions for obtaining the ciphertexts,
- $\leq 2^{64}N = 2^{199.23}$ F function computations for reducing the key candidates (2^{64} $RK_1|(RK_{24} \oplus WK_3)$ guesses, $2^{135.23}$ ciphertext pairs)

Therefore, time complexity of the attack is $\leq 2^{199}$ encryptions.

Data complexity is $2^{112.23}$ chosen plaintexts.

3.3.7 Extending 14-Round Attack to 15-Round

14-round attack can be further extended to 15 rounds by adding one more round to the beginning of the previous 14-round characteristic.

Structure: Let $\Omega = \{P_0 \oplus (00**), P_1 \oplus M_0(00**), P_2 \oplus (****), P_3 \oplus (****)\}$ where $P_0, P_1,$

P_2, P_3 are constant values and * is unknown nonzero byte difference. Then, there are $(2^8 - 1)^{14} \approx 2^{112}$ elements in Ω. This means that there are $\frac{2^{112} \cdot 2^{112}}{2} = 2^{223}$ plaintext pairs (P, P^*) having a difference $(M_1(00**), ****, 0, (00**))$.

1. Take 2 structures which gives 2^{113} plaintexts, 2^{224} plaintext pairs.

2. Choose plaintext pairs whose ciphertext pairs (C, C^*) has a difference $((** **), (** **), (00*0), M_0(00*0) \oplus M_1(00*0))$. The probability of obtaining such ciphertext pairs is $2^{224} \cdot 2^{-40} = 2^{184}$.

3. For every remaining pair (C, C^*) and its corresponding plaintext pair (P, P^*), guess $RK_3 \oplus WK_1 | (RK_{27} \oplus WK_2)(8\ bytes)$ and find $RK_{28} | RK_{29} | (WK_3 \oplus RK_{26}) | (RK_{24})_2 | RK_0 | RK_1 | WK_0 \oplus RK_2 | (RK_5)_{2,3}$ by using differential table look-ups. The probability of knowing that a candidate is wrong by using the differential tables is 2^{-176}. Therefore, the number of ciphertext pairs, N, required to narrow the 280-bit key down to the correct key is about 2^{184} from the equation

$$2^{280}(1 - 2^{-176})^N \approx 1.$$

Time complexity of the attack:

- 2^{113} encryptions for obtaining the ciphertexts,
- $\leq 2^{64}N = 2^{248}$ F function computations for reducing the key candidates

Accordingly, time complexity of the attack is $\leq 2^{248}$ encryptions.

Data complexity is 2^{113} chosen plaintexts.

Table 3.4: Complexity Comparison of Attacks on CLEFIA

CP: Chosen Plaintext, MA:Memory Access

Reference	Round	Key Length	Data(CP)	Time	Memory
[47, 48]	10	128, 192, 256	$2^{101.7}$	2^{102}	2^{32}
[47, 48]	11	192, 256	$2^{103.5}$	2^{188}	2^{121}
[49]	11	128, 192, 256	$2^{103.1}$	$2^{98.1}$	-
[47, 48]	12^{*}	256	$2^{103.8}$	2^{252}	2^{153}
[49]	12	128, 192, 256	$2^{119.3}$	$2^{114.3}$	-
[50]	12	128, 192, 256	$2^{118.9}$	2^{119}	2^{73}
[51]	12	128, 192, 256	$2^{110.93}$	2^{111}	-
[49]	13	192, 256	2^{120}	2^{181}	-
[50]	13	192, 256	$2^{119.8}$	2^{147}	2^{120}
[51]	13	192, 256	$2^{111.72}$	$\leq 2^{158}$	-
[49]	14	256	$2^{120.4}$	$2^{245.4}$	-
[50]	14	256	$2^{120.3}$	2^{211}	2^{121}
[51]	14	256	$2^{112.3}$	$\leq 2^{119}$	-
[51]	15	256	2^{113}	$\leq 2^{248}$	-

CHAPTER 4

BOOMERANG TYPE ATTACKS

This chapter describes the attacks so-called "boomerang type attacks". What is meant by "boomerang type attacks" is the boomerang attack and its improved versions. Section 4.1 explains the basic boomerang attack and gives an example of the attack applied on 8-round Serpent block cipher. An enhanced version of the boomerang attack, called "amplified boomerang attack" is described in Section 4.2. The next section deals with the "rectangle attack". Finally, "impossible boomerang attack" which combines impossible differential and boomerang techniques is mentioned in Section 4.3.

4.1 Boomerang Attack

Boomerang attack is an adaptively chosen plaintext and ciphertext attack and was first introduced by Wagner [52] in 1999. The attack is based on differential cryptanalysis and uses a pair of short differential characteristics instead of one long differential characteristic. The objective of applying boomerang technique is to mount attacks on the ciphers for which it is hard to find long differential characteristics with high probability and hence which are resistant to ordinary differential cryptanalysis. Since finding short differentials with high probability is easier than finding a long one with effective probability, the boomerang attack takes the advantage of this property and uses short characteristics for half of the cipher.

Assume that a block cipher $\mathbf{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ can be expressed as a cascade of two sub-ciphers $\mathbf{E} = \mathbf{E}_0 \circ \mathbf{E}_1$. Assume also that there exists a differential $\alpha \rightarrow \beta$ with probability p for E_0^K and another differential $\gamma \rightarrow \delta$ with probability q for E_1^K. Then, the boomerang distinguisher which is depicted in Figure 4.1 can generally be described as follows:

1. Choose a random plaintext P_1.

2. Generate other plaintext P_2 such that $P_2 = P_1 \oplus \alpha$.

3. Encrypt the plaintexts (P_1, P_2) and obtain the ciphertext pairs (C_1, C_2) through E, where $C_1 = E(P_1)$, $C_2 = E(P_2)$.

4. Form the second ciphertext pair (C_3, C_4) as $C_3 = C_1 \oplus \delta$ and $C_4 = C_2 \oplus \delta$.

5. Decrypt the ciphertexts C_3 and C_4 to obtain the plaintexts $P_3 = E^{-1}(C_3)$, $P_4 = E^{-1}(C_4)$ through E^{-1}.

6. Check whether $P_3 \oplus P_4 = \alpha$.

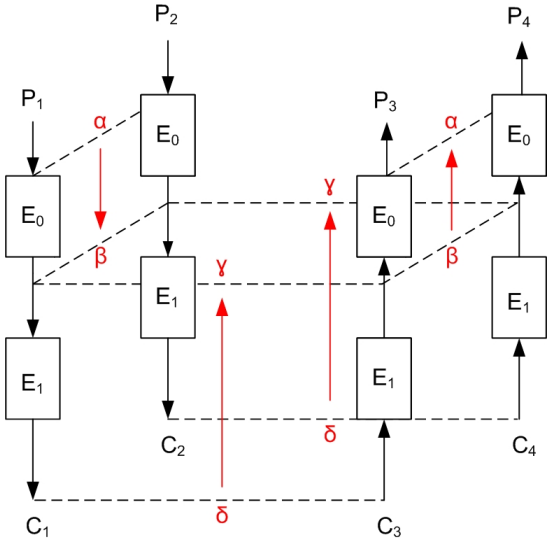

Figure 4.1: The Boomerang Distinguisher

The probability of the boomerang distinguisher can be computed in the following way:

- For a pair (P_1, P_2) such that $P_1 \oplus P_2 = \alpha$, the probability that $E_0(P_1) \oplus E_0(P_2) = \beta$ is p.
- For the pairs (C_1, C_3) and (C_2, C_4) such that $C_1 \oplus C_3 = C_2 \oplus C_4 = \delta$, the probability that $E_1^{-1}(C_1) \oplus E_1^{-1}(C_3) = \gamma$ is q and the probability that $E_1^{-1}(C_2) \oplus E_1^{-1}(C_4) = \gamma$ is also q. Therefore, the probability that both (C_1, C_3) and (C_2, C_4) satisfy the differential $\gamma \rightarrow \delta$ is q^2.
- If the above two conditions hold with the given probabilities, then

$$E_1^{-1}(C_3) \oplus E_1^{-1}(C_4) \quad = \quad E_0(P_3) \oplus E_0(P_4) \; = \; \beta \tag{4.1}$$

holds and thus $P_3 \oplus P_4 = \alpha$ holds with probability p. Explanation of why Equation 4.1 holds is given below.

$$\begin{aligned} E_1^{-1}(C_3) \oplus E_1^{-1}(C_4) &= \\ E_1^{-1}(C_3) \oplus E_1^{-1}(C_4) \oplus E_1^{-1}(C_1) \oplus E_1^{-1}(C_1) \oplus E_1^{-1}(C_2) \oplus E_1^{-1}(C_2) &= \\ \underbrace{E_1^{-1}(C_1) \oplus E_1^{-1}(C_3)} \oplus \underbrace{E_1^{-1}(C_2) \oplus E_1^{-1}(C_4)} \oplus \underbrace{E_1^{-1}(C_1) \oplus E_1^{-1}(C_2)} &= \\ \gamma \qquad \oplus \qquad \gamma \qquad \oplus \qquad \beta \qquad = \beta \end{aligned}$$

So, the overall probability that the boomerang distinguisher holds is p^2q^2. On the other hand, for a random permutation, the probability that $P_3 \oplus P_4 = \alpha$ holds is 2^{-n}. Therefore, $pq \geq 2^{-n/2}$ must be satisfied for the boomerang attack to work.

Furthermore, the boomerang attack can be applied for all possible β's and γ's with probability $(\hat{p}\hat{q})^2$, where

$$\hat{p} = \sqrt{\sum_{\beta'} Pr^2[\alpha \rightarrow \beta']}, \qquad \hat{q} = \sqrt{\sum_{\gamma'} Pr^2[\gamma' \rightarrow \delta]}.$$

4.1.1 8-Round Boomerang Attack on Serpent

8-round boomerang attack was proposed in [54]. The attack is mounted by adding one round to the end of the 7-round distinguisher.

4.1.1.1 Description of Serpent

Serpent [53] is a block cipher designed by Anderson et al. as a candidate for the AES competition. The cipher was selected by NIST as an AES finalist and took the second place among the five finalists. Serpent has a block size of 128 bits and variable key sizes of 128, 192 and 256 bits. It is a 32-round SPN operating on four 32-bit words. Except for the last round, each round of Serpent consists of three layers: key mixing, S-boxes and the linear transformation layer. In the last round, there is an additional key mixing operation instead of the linear transformation. Serpent has two versions in terms of the design criteria; namely a bitsliced version and a non-bitsliced version. Although these two versions are functionally equivalent, bitsliced version makes the cipher more efficient. In this chapter, bitsliced version of Serpent was considered and the same notation as in [53] was used.

The encryption starts with the initial permutation, (IP), continues with the round function, R_i, operated 32 times and ends with the final permutation, (FP). Rounds are numbered from 0 to 31. $\hat{B}_0$ denotes the 128-bit block which is equal to the data after IP is applied to the plaintext. $\hat{B}_{i+1}$ denotes the output of the i^{th} round. Each $\hat{B}_i$ is composed of four 32-bit words X_0, X_1, X_2, X_3. $\hat{K}_i$ represents the 128-bit subkey and $\hat{S}_i$ represents the S-box used in the i^{th} round. There are eight 4×4-bit S-boxes of Serpent. Each round function R_i, $i \in \{0, \ldots, 31\}$ uses $\hat{S}_{i(mod\ 8)}$ 32 times in parallel.

Formal description of the cipher is given by the following equations:

$$\hat{B}_0 := IP(P), \qquad \hat{B}_{i+1} := R_i(\hat{B}_i), \qquad C := FP(\hat{B}_{32})$$

where

$$\begin{aligned} R_i(X) &= LT(\hat{S}_i(X \oplus \hat{K}_i)) \qquad i{=}0,\ldots,30 \\ R_i(X) &= \hat{S}_i(X \oplus \hat{K}_i) \oplus \hat{K}_{32} \qquad i{=}31 \end{aligned}$$

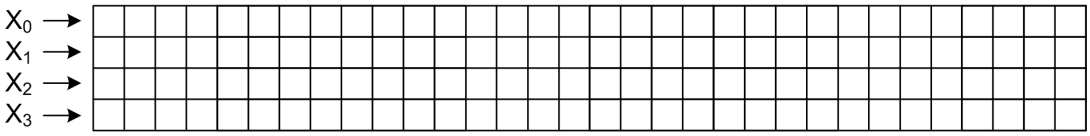

Figure 4.2: $\hat{B}_i$

The linear transformation, LT, is defined by:

$$\begin{aligned} X_0, X_1, X_2, X_3 &:= \hat{S}_i(\hat{B}_i \oplus \hat{K}_i) \\ X_0 &:= X_0 <<< 13 \\ X_2 &:= X_2 <<< 3 \\ X_1 &:= X_1 \oplus X_0 \oplus X_2 \\ X_3 &:= X_3 \oplus X_2 \oplus (X_0 << 3) \\ X_1 &:= X_1 <<< 1 \\ X_3 &:= X_3 <<< 7 \\ X_0 &:= X_0 \oplus X_1 \oplus X_3 \\ X_2 &:= X_2 \oplus X_3 \oplus (X_1 << 7) \\ X_0 &:= X_0 <<< 5 \\ X_2 &:= X_2 <<< 22 \\ \hat{B}_{i+1} &:= X_0, X_1, X_2, X_3 \end{aligned}$$

where <<< denotes rotation and << denotes shift.

4.1.1.2 7-Round Boomerang Distinguisher

The following 7-round boomerang distinguisher of Serpent was constructed by combining a 4-round differential characteristic with a 3-round one. In Figure 4.3 and Figure 4.4 only input and output differences of the characteristics are given. Intermediate differences can be found in [54].

- **4-Round Differential Characteristic E_0:**

  E_0 is a 4-round differential characteristic of Serpent corresponding to rounds 1 through 4 and has probability 2^{-31}.

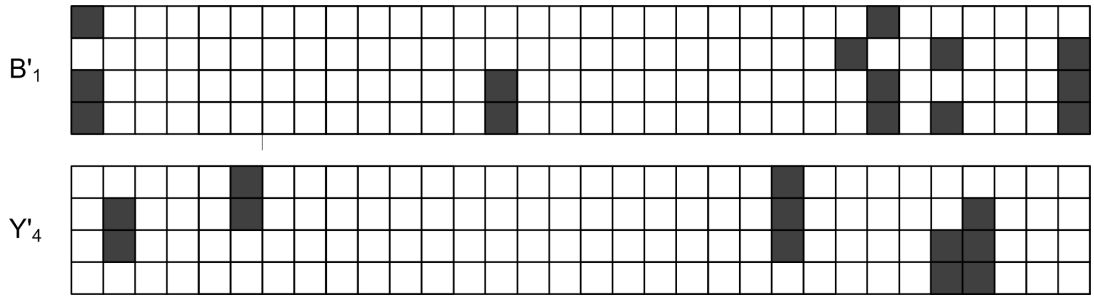

Figure 4.3: 4-Round Differential Characteristic $B'_1 \rightarrow Y'_4$

- **3-Round Differential Characteristic E_1:**

  E_1 is a 3-round differential characteristic corresponding to rounds five through seven and has probability 2^{-16}:

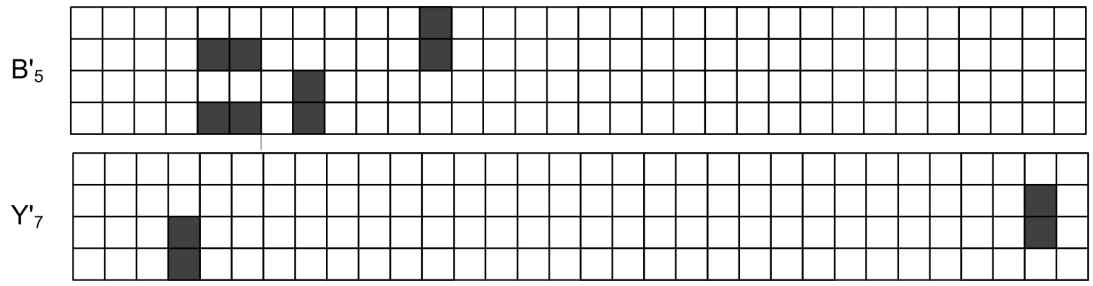

Figure 4.4: 3-Round Differential Characteristic $B'_5 \rightarrow Y'_7$

Therefore, the total probability of this 7-round distinguisher is $(2^{-31})^2 \cdot (2^{-16})^2 = 2^{-94}$.

Attack Procedure:

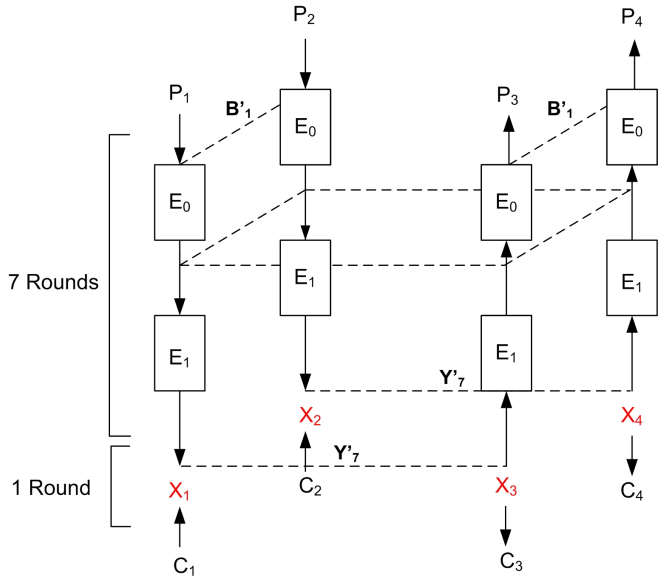

Figure 4.5: 8-Round Boomerang Attack

1. Take 2^{96} plaintext pairs (P_1, P_2) such that $P_1 \oplus P_2 = B'_1$.
2. Encrypt these plaintexts for 8 rounds to get the ciphertexts (C_1, C_2).
3. Guess the 68 subkey-bits corresponding to the 17 active S-boxes of the last round, peel off the last round and obtain (X_1, X_2).
4. Compute $X_3 = X_1 \oplus Y'_7$ and $X_4 = X_2 \oplus Y'_7$.
5. Encrypt (X_3, X_4) for one round with the guessed subkey and obtain (C_3, C_4).
6. Decrypt (C_3, C_4) for 8 rounds under the unknown key and get (P_3, P_4).
7. Check whether $P_3 \oplus P_4 = B'_1$. If the guessed subkey is correct, then $P_3 \oplus P_4 = B'_1$ with probability 2^{-94}.

Complexity Analysis:

In step 2, there are 2^{97} 8-round encryptions. Step 3 requires $2^{68} \cdot 2^{97} = 2^{165}$ 1-round partial decryptions. In step 4, we get the entire codebook of all 2^{128} (X_3, X_4). Hence, there are 2^{128}

1-round encryptions in step 5. Finally, step 6 requires 2^{128} 8-round decryptions. In total, there are approximately 2^{163} 8-round Serpent encryptions and the attack requires access to the entire codebook, i.e. 2^{128} plaintexts and thus 2^{133} bytes random access memory.

4.2 Amplified Boomerang Attack

Boomerang attack is very advantageous in the sense that it uses two short differentials with high probability instead of a long one with lower probability and gives the opportunity to mount attacks on more rounds of the cipher. However, because of the adaptively chosen plaintext and ciphertext property, boomerang attack becomes inefficient. Applying the attack turns out to be harder since it requires more data and memory.

In 2000, Kelsey et al. [55] presented a method called "amplified boomerang" which turns the adaptively chosen plaintext and ciphertext property of the boomerang attack into a chosen plaintext by using the birthday paradox. In amplified boomerang attack, the main idea is the same as in the boomerang attack, i.e., using two short differential characteristics with high probability.

Amplified boomerang attack is based on quartets of plaintexts $((P_1, P_2), (P_3, P_4))$ such that $P_1 \oplus P_2 = P_3 \oplus P_4 = \alpha$ if the following conditions hold:

- $E_0(P_1) \oplus E_0(P_2) = E_0(P_3) \oplus E_0(P_4) = \beta$
- $E_0(P_1) \oplus E_0(P_3) = E_0(P_2) \oplus E_0(P_4) = \gamma$
- $E_1(E_0(P_1)) \oplus E_1(E_0(P_3)) = E_1(E_0(P_2)) \oplus E_1(E_0(P_4)) = \delta$

Assume that we have N pairs (P_i, P_j) such that $P_i \oplus P_j = \alpha$,

- Np of the pairs satisfy the first differential E_0, i.e., for Np pairs, the equation

$$E_0(P_i) \oplus E_0(P_j) = \beta \qquad \text{holds.} \tag{4.2}$$

- Np pairs generate $\binom{Np}{2}$ quartets which is approximately equal to $\frac{(Np)^2}{2}$ quartets,

- assuming that the intermediate encryption values have uniform distribution over all possible values, then

$$E_0(P_1) \oplus E_0(P_3) = \gamma \tag{4.3}$$

holds with probability 2^{-n}. When Equation 4.3 is satisfied, we have $E_0(P_2) \oplus E_0(P_4) = \gamma$ automatically. The reason is as follows. Because $E_0(P_1) \oplus E_0(P_2) = \beta$ and $E_0(P_3) \oplus E_0(P_4) = \beta$,

$$E_0(P_1) \oplus E_0(P_3) \oplus E_0(P_2) \oplus E_0(P_4) = 0 \qquad \text{holds.} \tag{4.4}$$

Combining Equation 4.3 and Equation 4.4, we obtain $E_0(P_2) \oplus E_0(P_4) = \gamma$. So, the plaintext quartets satisfy the intermediate difference with probability 2^{-n} and the number of these quartets is about $\frac{(Np)^2}{2} \cdot 2^{-n} = (Np)^2 \cdot 2^{-n-1}$.

- Remaining $(Np)^2 \cdot 2^{-n-1}$ quartets satisfy the differential E_1 with probability q^2.

Therefore, number of expected right quartets are $N^2 \cdot (pq)^2 \cdot 2^{-n-1}$.

4.2.1 8-Round Amplified Boomerang Attack on Serpent

This attack was given in [54] and [55].

4.2.1.1 7-Round Amplified Boomerang Distinguisher

7-round amplified boomerang distinguisher was constructed by combining the same differential characteristics, E_0 and E_1, as described in Section 4.1.1. However, the probability of the amplified boomerang distinguisher is $(pq)^2 \cdot 2^{-n-1} = (2^{-31} \cdot 2^{-16})^2 \cdot 2^{-129} = 2^{-223}$ whereas the probability of the boomerang distinguisher is 2^{-94}. This means that in order to mount the amplified boomerang attack, more chosen plaintexts are needed.

If we choose 2^{113} plaintext pairs with input difference B'_1, after encrypting with E_0, we will get $2^{113} \cdot 2^{-31} = 2^{82}$ pairs having the output difference Y'_4. From these pairs, approximately $\binom{2^{82}}{2} \approx 2^{163}$ quartets can be formed. Among these quartets, it is expected that there remains $2^{163} \cdot 2^{-128} = 2^{35}$ quartets that satisfy the difference B'_5. After encrypting with E_1, we will end up with $2^{35} \cdot (2^{-16})^2 = 2^3$ quartets that satisfy the boomerang distinguisher.

8-round amplified boomerang attack can be applied by adding one round to the end of the 7-round distinguisher and can be described basically as taking enough number of chosen plaintext pairs, encrypting the pairs, guessing some parts of the last round subkey and checking the difference in the output of the seventh round. The details of the attack are as follows.

Attack Procedure:

1. Take 2^{113} chosen plaintext pairs (P_1, P_2) such that $P_1 \oplus P_2 = B'_1$.

2. Encrypt these plaintexts for 8 rounds of Serpent and obtain the ciphertext pairs (C_1, C_2).

3. Guess 68 bits of the last round subkey and decrypt these ciphertexts for one round.

4. Check whether the difference across the pairs is Y'_7.

Complexity Analysis:

Encrypting 2^{113} chosen plaintext pairs for 8 rounds means 2^{114} 8-round Serpent encryptions. Step 3 takes $2^{68} \cdot 2^{114}$ 1-round Serpent decryptions. Therefore, the time complexity of the 8-round amplified boomerang attack is 2^{179} 8-round Serpent encryptions. Data complexity of the attack is 2^{113} chosen plaintext pairs and memory complexity is $2^{114} \cdot 2^4$ bytes for plaintexts + $2^{114} \cdot 2^4$ bytes for ciphertexts which is 2^{119} bytes of memory in total.

4.3 Rectangle Attack

The name "rectangle" was first suggested by Biham et al. [56] in 2001. Besides with some improvements on the probability, rectangle attack basically depends on the same idea in amplified boomerang attack [55]. Rectangle attack makes use of combinations of two differentials, E_0 and E_1, and is based on quartets of plaintexts as in (amplified) boomerang attack. It searches for the pairs of plaintext pairs which have a fixed difference α and a fixed difference δ after the encryption $E = E_1 \circ E_0$ just like in amplified boomerang attack. However, there are some improvements on the amplified boomerang attack which result in an increase in the probability.

Improvements on the Amplified Boomerang Attack:

- The first improvement was made by Biham et al. [56] which increases the probability of obtaining quartets by a factor of 2. This can be explained as follows:

  Let $(P_1, P_2), (P_3, P_4)$ be the plaintext quartet such that $P_1 \oplus P_2 = \alpha$ and $P_3 \oplus P_4 = \alpha$. If these pairs satisfy the differential E_0, then $E_0(P_1) \oplus E_0(P_2) = \beta$ and $E_0(P_3) \oplus E_0(P_4) = \beta$. The probability that having a difference γ between $E_0(P_1)$ and $E_0(P_3)$ is 2^{-n} and when this difference is satisfied, $E_0(P_2) \oplus E_0(P_4) = \gamma$ automatically by the boomerang condition. The same is valid for $E_0(P_1)$ and $E_0(P_4)$, i.e, $E_0(P_1) \oplus E_0(P_4) = \gamma$ with probability 2^{-n}, then $E_0(P_2) \oplus E_0(P_3) = \gamma$. Hence, there are two ways to use the pairs as a quartet with probability 2^{-n+1}.

- The second improvement was proposed by Kohno et al. [55]. As can be seen from Figure 4.6, input difference γ of the second differential does not affect the process of the amplified boomerang attack. Hence, the attack succeeds for any γ' provided that $\gamma' \rightarrow \delta$ through E_1. Therefore, the probability $Pr^2[\gamma \rightarrow \delta] = q^2$ was enhanced to $\sum_{\gamma'} Pr^2[\gamma' \rightarrow \delta] = \hat{q}^2$ in rectangle attack and the number of right quartets becomes $\binom{Np}{2} \cdot 2^{-n+1} \cdot \hat{q}^2$.

- The third improvement was made by Biham et al. [56] and is quite similar to the previous one. Again, instead of being restricted with one β value, any difference β' can be used where $\alpha \rightarrow \beta'$ for E_0 with sufficient probability. Then, probability of the rectangle distinguisher becomes

$$\sum_{\beta'} Pr^2[\alpha \rightarrow \beta'] \cdot 2^{-n+1} \cdot \sum_{\gamma'} Pr^2[\gamma' \rightarrow \delta] = 2^{-n+1} \cdot \hat{p}^2 \cdot \hat{q}^2$$

  and the number of right quartets reduces to

$$\begin{aligned} \binom{N \cdot Pr(\alpha \rightarrow \beta')}{2} \cdot 2^{-n+1} \cdot \sum_{\gamma'} Pr^2[\gamma' \rightarrow \delta] & \\ = \; & N^2 \cdot 2^{-n} \cdot \sum_{\beta'} Pr^2[\alpha \rightarrow \beta'] \cdot \sum_{\gamma'} Pr^2[\gamma' \rightarrow \delta] \\ = \; & N^2 \cdot 2^{-n} \cdot \hat{p}^2 \cdot \hat{q}^2 \end{aligned}$$

- The last improvement is belong to again Biham et al. [56] and based on the previous two improvements. They stated that if, for the first differential, $\alpha \rightarrow a$ for the first pair

and $\alpha \rightarrow b$ for the second pair, then the characteristics for which $\gamma \rightarrow \delta$ and $\gamma \oplus a \oplus b \rightarrow \delta$ can be used for the second differential. This time, the number of right quartets will be

$$N^2 \cdot 2^{-n} \cdot \sum_{a,b} \Big[Pr(\alpha \rightarrow a) Pr(\alpha \rightarrow b) \cdot \sum_{\gamma} Pr(\gamma \rightarrow \delta) Pr(\gamma \oplus a \oplus b \rightarrow \delta) \Big].$$

They also noted that although this improvement counts all quartets with plaintext difference α and ciphertext difference δ, doing the exact calculation is very difficult.

The rectangle distinguisher is shown in Figure 4.7.

4.3.1 10-Round Rectangle Attack on Serpent

This attack was presented in [56] and based on a 8-round rectangle distinguisher. The distinguisher was constructed by combining two 4-round differential characteristics, namely E_0 and E_1. The differential E_0 ($\alpha \rightarrow \beta$) is through rounds one to four and has probability 2^{-29}. By using the third improvement and searching for all possible output differences β', the probability becomes $\sum_{\beta'} Pr^2[\alpha \rightarrow \beta'] = 2^{-50.8}$ instead of $(2^{-29})^2 = 2^{-58}$. The second differential E_1 ($\gamma \rightarrow \delta$) is through rounds 5 to 8 and has probability 2^{-47}. Using similar arguments, we obtain $\sum_{\beta'} Pr^2[\alpha \rightarrow \beta'] = 2^{-69.8}$ instead of $(2^{-47})^2 = 2^{-96}$. The probability of this attack can be computed as $2^{-128} \cdot 2^{-50.8} \cdot 2^{-69.8} = 2^{-248.6}$. Therefore, in order to get 8 right quartets after running the attack, we should take at least $2^{-125.8}$ plaintext pairs with difference α as the following equation states;

$$N^2 \cdot 2^{-248.6} \geq 2^3.$$

10-round rectangle attack on Serpent has a time complexity of $2^{-208.4}$ 10-round Serpent encryptions and requires $2^{-125.8}$ chosen plaintext pairs and $2^{-131.8}$ bytes of memory. Details of the attack procedure and description of the differential characteristics can be found in [56].

4.4 Impossible Boomerang Attack

In this section, a new extension of differential cryptanalysis called "Impossible Boomerang Attack" is described. This attack was proposed in Ph.D thesis of Lu [42]. As can be understood from its name, impossible boomerang attack combines the impossible differential method with the boomerang technique. The main idea in impossible differential cryptanalysis

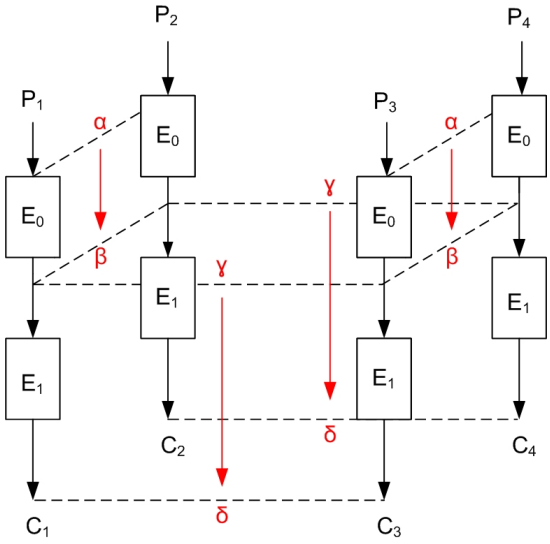

Figure 4.6: The Amplified Boomerang Distinguisher

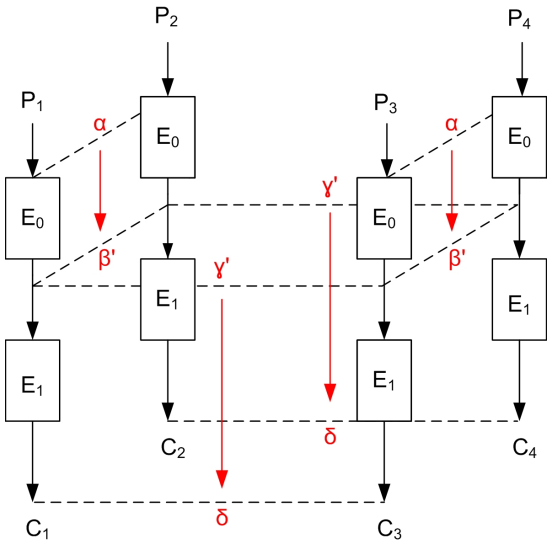

Figure 4.7: The Rectangle Distinguisher

is to find two differential characteristics such that each differential is satisfied with probability 1 and the intermediate differences at the meeting point will never be satisfied. So, roughly speaking impossible boomerang attack operates on quartets and looks for differentials with probability 1 where the intermediate differences induce a contradiction.

4.4.1 Impossible Boomerang Distinguisher

As in the case of most attacks, impossible boomerang attack is based on a distinguisher named impossible boomerang distinguisher. Distinguisher treats the block cipher as a cascade of two sub-ciphers $\mathbf{E} = \mathbf{E}_0 \circ \mathbf{E}_1$. Two (or more) differentials with probability 1 are used for both $\mathbf{E}_0$ and $\mathbf{E}_1$ where the XOR of the intermediate differences of these differentials is nonzero. Impossible boomerang distinguisher is depicted in Figure 4.8.

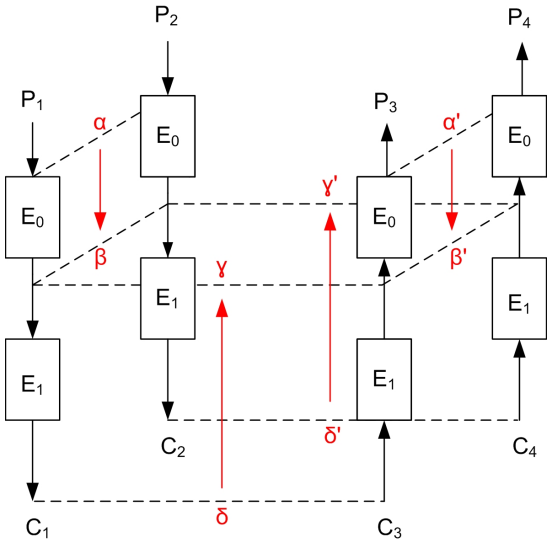

Figure 4.8: Impossible Boomerang Distinguisher

Theorem 4.4.1 *[42] Assume that $\alpha \to \beta$ and $\alpha' \to \beta'$ are differentials with probability 1 for E_0, and $\delta \to \gamma$ and $\delta' \to \gamma'$ are differentials with probability 1 for E_1^{-1}, where $\beta \oplus \beta' \oplus \gamma \oplus \gamma' \neq 0$. Then the following equations cannot both hold:*

$$E(P_1) \oplus E(P_3) = \delta \tag{4.5}$$

$$E(P_2) \oplus E(P_4) = \delta' \tag{4.6}$$

Proof. Suppose that Equation 4.5 and Equation 4.6 holds. Because the differentials $\alpha \to \beta$ and $\alpha' \to \beta'$ are satisfied with probability 1 for E_0, we have

$$\begin{aligned} E_0(P_1) \oplus E_0(P_2) &= \beta \\ E_0(P_3) \oplus E_0(P_4) &= \beta'. \end{aligned}$$

Since the differentials $\delta \to \gamma$ and $\delta' \to \gamma'$ holds with probability 1 for E_1^{-1}, the following equation also holds with probability 1:

$$\begin{aligned} \beta' &= E_0(P_3) \oplus E_0(P_4) \\ &= (E_0(P_3) \oplus E_0(P_1)) \oplus (E_0(P_1) \oplus E_0(P_2)) \oplus (E_0(P_2) \oplus E_0(P_4)) \\ &= ((E_1^{-1})(E(P_3)) \oplus (E_1^{-1})(E(P_1))) \oplus (E_0(P_1) \oplus E(P_2)) \oplus ((E_1^{-1})(E(P_2)) \oplus (E_1^{-1})(E(P_4))) \\ &= \gamma \oplus \beta \oplus \gamma' \end{aligned}$$

This gives $\beta' \oplus \gamma \oplus \beta \oplus \gamma' = 0$ which contradicts with the assumption $\beta \oplus \beta' \oplus \gamma \oplus \gamma' \neq 0$.

Therefore, Equation 4.5 and Equation 4.6 cannot hold together.

4.4.2 Overview of the Attack

Let the block cipher $\mathbf{E}$ be described as $\mathbf{E} = \mathbf{E}_a \circ \mathbf{E}_0 \circ \mathbf{E}_1 \circ \mathbf{E}_b$, where $\mathbf{E}_0 \circ \mathbf{E}_1$ denotes the rounds for which the impossible boomerang distinguisher $(\alpha, \alpha') \nrightarrow (\delta, \delta')$ holds, $\mathbf{E}_a$ denotes the rounds before $\mathbf{E}_0$ and $\mathbf{E}_b$ denotes the rounds after $\mathbf{E}_1$. Assume that K_a and K_b are the guesses for the subkeys used in $\mathbf{E}_a$ and $\mathbf{E}_b$, respectively. Then, in a key recovery attack, the attacker can eliminate the wrong keys with the following algorithm:

1. Under the guess of K_a, partially encrypt the plaintexts and check whether the following equations are satisfied:

$$E_a^{K_a}(P_1) \oplus E_a^{K_a}(P_2) = \alpha \tag{4.7}$$

$$E_a^{K_a}(P_3) \oplus E_a^{K_a}(P_4) = \alpha' \tag{4.8}$$

2. If the above equations are satisfied, then request the ciphertexts of the chosen plaintexts under $\mathbf{E}$.

3. Partially decrypt these ciphertexts under the guess K_b and check the following equations:

$$(E_b^{K_b})^{-1}(C_1) \oplus (E_b^{K_b})^{-1}(C_2) = \delta \tag{4.9}$$

$$(E_b^{K_b})^{-1}(C_3) \oplus (E_b^{K_b})^{-1}(C_4) = \delta' \tag{4.10}$$

4. If these equations are satisfied, discard the guess (K_a,K_b).

CHAPTER 5

CONCLUSION

For many years, differential cryptanalysis and linear cryptanalysis have been the most fundamental and the most effective tools in block cipher cryptanalysis. For this reason, in the design of most modern block ciphers it is highly considered that the cipher should be provably secure against differential and linear cryptanalysis. Accordingly, ciphers' being resistant to these attacks stimulate cryptanalysts and also designers to develop the existing attack methods and explore new techniques. Thus, new cryptanalytic techniques is always desirable and considered necessary for a better evaluation of the security of a block cipher and also for the design of more secure ciphers.

This thesis covers different types of combined attacks based on differential and/or linear cryptanalysis, also illustrates these attacks by giving applications on some ciphers. First combined attack is the differential-linear attack. As an example, differential-linear cryptanalysis and enhanced differential-linear cryptanalysis of DES are given. As other versions of differential-linear cryptanalysis; differential-bilinear, higher order differential-linear, differential-nonlinear and square-nonlinear attacks are mentioned in Chapter 2. Impossible differential attack is another combined attack which is constructed by combining two differential distinguishers. Applications of impossible differential cryptanalysis on the block ciphers IDEA, CLEFIA and AES are presented in Chapter 3. Boomerang type attacks such as the original boomerang, amplified boomerang ad rectangle attacks which are another differential-differential type combined attack are described in Chapter 4 and examplified on the block cipher Serpent.

REFERENCES

[1] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, ISBN 0-8493-8523-7, Available Online at http://www.cacr.math.uwaterloo.ca/hac/, 1996.

[2] A. Kerckhoffs, *La Cryptographie Militaire*, Journal des sciences militaires, vol. IX, pp. 5-83, 1883.

[3] National Bureau of Standards, *Data Encryption Standard*, Federal Information Processing Standards Publications No.46, 1977.

[4] E. Biham, A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, Advances in Cryptology-CRYPTO'90, Lecture Notes in Computer Science, vol.537, pp. 2-21, Springer-Verlag, 1991.

[5] E. Biham, A. Shamir, *Differential Cryptanalysis of Data Encryption Standard*, Springer-Verlag, 1993.

[6] E. Biham, A. Shamir, *Differential Cryptanalysis of the Full 16-Round DES*, Advances in Cryptology-CRYPTO'92, Lecture Notes in Computer Science, vol.740, pp. 487-496, Springer-Verlag, 1993.

[7] F. Mirza, *Block Ciphers and Cryptanalysis*, 1998.

[8] M. Matsui, A. Yamagishi *A New Method for Known Plaintext Attack of FEAL Cipher*, Advances in Cryptology-EUROCRYPT'92, Lecture Notes in Computer Science, vol.658, pp. 81-91, Springer-Verlag, 1993.

[9] M. Matsui, *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology-EUROCRYPT'93, Lecture Notes in Computer Science, vol.765, pp. 386-397, 1994.

[10] E. Biham, *On Matsui's Linear Cryptanalysis*, Advances in Cryptology-EUROCRYPT'94, Lecture Notes in Computer Science, vol.950, pp. 341, Springer-Berlin, 1995.

[11] S. K. Langford, M. E. Hellman, *Differential-Linear Cryptanalysis*, Advances in Cryptology-CRYPTO'94, Lecture Notes in Computer Science, vol.839, pp. 17-25, 1994.

[12] E. Biham, O. Dunkelman, N. Keller, *Enhanced Differential-Linear Cryptanalysis*, Advances in Cryptology- ASIACRYPT '02, Lecture Notes in Computer Science vol.2501, pp. 254-266, Springer-Verlag, 2002.

[13] A. Górska, K. Górski, Z. Kotulski, A. Paszkiewicz, J. Szczepański *New Experimental Results in Differential-Linear Cryptanalysis of Reduced Variants of DES*, 8^{th} International Conference on Advanced Computer Systems ACS'2001, vol.1, pp.333-346.

[14] N. T. Courtois, *Feistel Schemes and Bi-Linear Cryptanalysis(extended version)*, private communications, 2004.

[15] L.R. Knudsen, *Truncated and Higher Order Differentials*, Fast Software Encryption 2, Lecture Notes in Computer Science vol.1008, pp. 196-211, Springer-Verlag, 1995.

[16] A. Shimizu, S. Miyaguchi, *Fast Data Encipherment Algorithm FEAL*, Advances in Cryptology- EUROCRYPT '87, Lecture Notes in Computer Science vol.304, pp. 267-278, Springer-Verlag, 1988.

[17] S. Murphy, *The Cryptanalysis of FEAL-4 with 20 Chosen Plaintexts*, Journal of Cryptology-2(3): pp. 145-154, 1990.

[18] E. Biham, O. Dunkelman, N. Keller, *New Combined Attacks on Block Ciphers*, Advances in Cryptology- FSE '05, Lecture Notes in Computer Science vol.3557, pp. 126-144, Springer-Verlag, 2005.

[19] H. Handschuh, D. Naccache, *SHACAL*, Proceedings of first open NESSIE workshop, 2000, http://www.cosic.esat.kuleuven.be/nessie/ workshop/submissions.html.

[20] H. Handschuh, D. Naccache, *SHACAL*, NESSIE, 2001,
http://www.cosic.esat.kuleuven.be/nessie/tweaks.html.

[21] U.S. Department of Commerce, *Secure Hash Standard FIPS 180-1*, N.I.S.T., 1995.

[22] U.S. Department of Commerce, *Secure Hash Standard FIPS 180-2*, N.I.S.T., 2002.

[23] S. Hong, J. Kim, G. Kim, J. Sung, C. Lee, S. Lee, *Impossible Differential Attack on 30-Round SHACAL-2*, INDOCRYPT 2003, Lecture Notes in Computer Science vol.2904, pp. 97-106, Springer-Verlag, 2003.

[24] Y. Shin, J. Kim, G. Kim, S. Hong, S. Lee, *Differential-Linear Type Attacks on Reduced Rounds of SHACAL-2*, ACISP 2004, Lecture Notes in Computer Science vol.3108, pp. 110-122, Springer-Verlag, 2004.

[25] J. Daemen, L. Knudsen, V. Rijmen, *The Block Cipher Square*, FSE'97, Lecture Notes in Computer Science vol.1267, pp. 149-165, Springer-Verlag.

[26] E. Biham, A. Biryukov, A. Shamir, *Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials*, Advances in Cryptology- EUROCRYPT'99, Lecture Notes in Computer Science, vol.1592, pp. 12-23, Springer-Verlag, 1999.

[27] L.R. Knudsen, *DEAL-A 128-bit Block Cipher* Technical Report 151, Department of Informatics, University of Bergen Norway (1998).

[28] X. Lai, J.L. Massey, *A Proposal for a New Block Encryption Algorithm*, Advances in Cryptology-Proceedings of EUROCRYPT'90, Lecture Notes in Computer Science, vol.473, pp. 389-404, Springer-Verlag, 1991.

[29] E. Biham, A. Biryukov, A. Shamir, *Miss in the Middle Attacks on IDEA and Khufu*, 6^{th} International Workshop on Fast Software Encryption, Lecture Notes in Computer Science, vol.1636, pp. 124-138, Springer-Verlag, 1999.

[30] J. Daemen, V. Rijmen, *The Rijndael Block Cipher*, AES Proposal, 1999.

[31] NIST-National Institute of Standards and Technology, *Advanced Encryption Standard (AES)*, FIPS-197, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.

[32] E. Biham, N. Keller, *Cryptanalysis of Reduced Variants of Rijndael*, 3rd AES Conference, 2000.

[33] J.H. Cheon, M. Kim, K. Kim, J-Y. Lee, S. Kang, *Improved Impossible Differential Cryptanalysis of Rijndael and Crypton*, in: ICISC 2001, LNCS, vol.2288, pp. 39-49, Springer-Verlag.

[34] R.C.-W. Phan, M.U. Siddiqi, *Generalized Impossible Differentials of Advanced Encryption Standard*, Electronics Letters 37 (14)(2001), pp. 896-898.

[35] R.C.-W. Phan, *Classes of Impossible Differentials of Advanced Encryption Standard*, Electronics Letters 38 (11)(2002), pp. 508-510.

[36] R.C.-W. Phan, *Impossible Differential Cryptanalysis of 7-Round Advanced Encryption Standard(AES)*, Information Processing Letters, vol.91, n.1, pp. 33-38, 2004.

[37] W. Zhang, W. Wu, D. Feng, *New Results on Impossible Differential Cryptanalysis of Reduced AES*, ICISC 2007, LNCS, vol.4817, pp. 239-250, Springer-Verlag.

[38] B. Bahrak, M.R. Aref, *Impossible Differential Attack on Seven-Round AES-128*, IET.Inf.Secur., 2008, vol.2, no.2, pp. 28-32.

[39] J.Chen, Y. Hu, Y. Zhang, *Impossible Differential Cryptanalysis of Advanced Encryption Standard*, Science in China Series F:Information Sciences, vol.50, no.3, pp.342-350, Springer, 2007.

[40] J. N. J*ú*nior, *Cryptanalysis and Design of Block Ciphers* Ph.D. Thesis, 2003.

[41] O. Dunkelman, *Techniques for Cryptanalysis of Block Ciphers* Ph.D. Thesis, 2006.

[42] J. Lu, *Cryptanalysis of Block Ciphers* Ph.D. Thesis, 2008.

[43] J. Kim, *Combined Differential, Linear and Related Key Attacks on Block Ciphers and MAC Algorithms* Ph.D. Thesis, 2008.

[44] A. Darbuka, *Related-Key Attacks on Block Ciphers* M.Sc. Thesis, 2009.

[45] D. Çelik, *Basic Cryptanalysis Methods on Block Ciphers* M.Sc. Thesis, to be completed.

[46] J. Lu, O. Dunkelman, N. Keller, J. Kim, *New Impossible Differential Attacks on AES*, INDOCRYPT'08, LNCS 5365, pp.279-293, Springer-Verlag, 2008.

[47] T.Shirai, K. Shibutani, T. Akishita, S. Moriai, T. Iwata, *The 128-Bit Blockcipher CLEFIA (Extended Abstract)*, FSE'07, LNCS 4593, pp.181-195, Springer-Verlag, 2007.

[48] Sony Corporation, *The 128-Bit Blockcipher CLEFIA, Security and Performance Evaluations, Revision 1.0*, 2007.

[49] W. Wang, X. Wang, *Improved Impossible Differential Cryptanalysis of CLEFIA*, http://eprint.iacr.org/2007/466

[50] Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Saito, T. Suzaki, H. Kubo, *Impossible Differential Cryptanalysis of CLEFIA*, FSE'08, LNCS 5086, pp.398-411, Springer-Verlag, 2008.

[51] B. Sun, R. Li, M. Wang, P. Li, C. Li, *Impossible Differential Cryptanalysis of CLEFIA*,

[52] D. Wagner, *The Boomerang Attack*, FSE'99, LNCS 1636, pp.156-170, Springer, 1999.

[53] R. Anderson, E. Biham, L.R. Knudsen, *Serpent: A Proposal for the Advanced Encryption Standard*, NIST AES Proposal, 1998.

[54] T. Kohno, J. Kelsey, B. Schneier, *Preliminary Cryptanalysis of Reduced-Round Serpent*, Third AES Candidate Conference, 2000.

[55] J. Kelsey, T. Kohno, B. Schneier, *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent*, FSE 2000, LNCS 1978, pp.75-93, Springer-Verlag, 2001.

[56] E. Biham, O. Dunkelman, N. Keller, *The Rectangle Attack-Rectangling the Serpent*, EUROCRYPT 2001, LNCS 2045, pp.340-357, Springer-Verlag, 2001.

[57] E. Biham, O. Dunkelman, N. Keller, *New Results on Boomerang and Rectangle Attacks*, FSE 2002, LNCS 2365, pp.1-16, Springer-Verlag, 2002.