IMPROBABLE DIFFERENTIAL CRYPTANALYSIS

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

BY

CİHANGİR TEZCAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
CRYPTOGRAPHY

JUNE 2014

Approval of the thesis:

IMPROBABLE DIFFERENTIAL CRYPTANALYSIS

submitted by **CİHANGİR TEZCAN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Department of Cryptography, Middle East Technical University** by,

Prof. Dr. Bülent Karasözen
Director, Graduate School of **Applied Mathematics** _____

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

Assoc. Prof. Dr. Ali Doğanaksoy
Supervisor, **Mathematics, METU** _____

Prof. Dr. Ersan Akyıldız
Co-supervisor, **Mathematics, METU** _____

Examining Committee Members:

Prof. Dr. Ferruh Özbudak
Mathematics, METU _____

Assoc. Prof. Ali Doğanaksoy
Mathematics, METU _____

Dr. Muhiddin Uğuz
Mathematics, METU _____

Prof. Dr. Ali Aydın Selçuk
Computer Engineering, TOBB ETU _____

Assist. Prof. Dr. Zülfükar Saygı
Mathematics, TOBB ETU _____

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: CİHANGİR TEZCAN

Signature :

ABSTRACT

IMPROBABLE DIFFERENTIAL CRYPTANALYSIS

TEZCAN, Cihangir

Ph.D., Department of Cryptography

Supervisor : Assoc. Prof. Dr. Ali Doğanaksoy

Co-Supervisor : Prof. Dr. Ersan Akyıldız

June 2014, 84 pages

We present a new statistical cryptanalytic technique that we call improbable differential cryptanalysis which uses a differential that is less probable when the correct key is used. We provide data complexity estimates for this kind of attacks and we also show a method to expand impossible differentials to improbable differentials. By using this expansion method, we cryptanalyze 13, 14, and 15-round CLEFIA for the key sizes of length 128, 192, and 256 bits, respectively. These are the best cryptanalytic results on CLEFIA up to this date.

We introduce a new criteria for evaluating S-boxes that we call undisturbed bits and attack PRESENT and SERPENT by exploiting their S-boxes. Without using undisturbed bits, the longest improbable differential attack we could find for PRESENT had a length of 7-rounds. However, we show that PRESENT has 6 undisturbed bits and by using them, we can construct 10-round improbable differentials and attack PRESENT reduced to 13 rounds. Similarly, without using undisturbed bits, the longest impossible differential we could find on SERPENT had a length of 3.5 rounds. However, we obtained four 5.5-round impossible differentials on SERPENT and provided a 7-round improbable differential attack. Hence, undisturbed bits should be avoided by S-box designers.

Moreover, we provide a second S-box property that we call differential factors. A key recovery attack may not capture the whole subkey corresponding to a S-box with a differential factor. This helps the attacker to guess less subkey bits and reduce the

time complexity of the attack. By using differential factors, we show that 10, 11, and 12-round differential-linear attacks of Dunkelman et al. on SERPENT can actually be performed with time complexities reduced by a factor of 4, 4, and 8, respectively. Furthermore, we slightly reduce the data complexity of these attacks by changing the differential with a more probable one but end up with an attack with higher time complexity.

Keywords : cryptanalysis, block cipher, improbable differential, undisturbed bit, differential factor

ÖZ

OLASI OLMAYAN DİFERANSİYEL KRİPTANALİZ

TEZCAN, Cihangir
Doktora, Kriptografi
Tez Yöneticisi : Doç. Dr. Ali Doğanaksoy
Ortak Tez Yöneticisi : Prof. Dr. Ersan Akyıldız

Haziran 2014, 84 sayfa

Doğru anahtar kullanıldığında daha az ihtimalle gözlemlenen diferansiyeller kullanan ve olası olmayan diferansiyel kriptanaliz ismini verdiğimiz yeni bir istatistiksel kriptanaliz tekniği sunuyoruz. Bu tür atakların veri karmaşıklığının yaklaşık olarak hesaplarını ve imkansız diferansiyelleri olası olmayan diferansiyellere genişleten bir metod sunuyoruz. Bu genişletme tekniğiyle CLEFIA şifresinin 13, 14 ve 15 döngüsüne ataklar sunuyoruz. Bu ataklar CLEFIA için bilinen en iyi ataklardır.

Değişim kutuları için rahatsız edilmemiş bit ismini verdiğimiz yeni bir özellik öneriyoruz ve PRESENT ve SERPENT şifrelerine bu özelliği kullanarak saldırıyoruz. Rahatsız edilmemiş bit kullanmadan PRESENT şifresinin en fazla 7 döngüsüne olası olmayan diferansiyel atak yapabildik. Ama bu şifredeki 6 rahatsız edilmemiş bit sayesinde 10 döngülük olası olmayan diferansiyel ile 13 döngüsünü kırabiliyoruz. Benzer şekilde rahatsız edilmemiş bit kullanmadan SERPENT şifresine en fazla 3.5 döngülük imkansız diferansiyel bulabildik. Ama rahatsız edilmemiş bitler sayesinde 5.5 döngülük imkansız diferansiyel elde ettik ve şifrenin 7 döngüsüne olası olmayan diferansiyel atak uygulayabildik. Bu nedenlerle değişim kutusu tasarımcılarının rahatsız edilmemiş bitlerden kaçınmalarını tavsiye ediyoruz.

Sunduğumuz diğer bir değişim kutusu özelliği de diferansiyel faktörler. Anahtar elde etme ataklarında diferansiyel faktörü olan değişim kutularına saldırırken, buradaki anahtar bitlerinin bazılarını elde etmek mümkün olmayabilir. Bu özellik saldırıyı gerçekleştirenin daha az anahtar bitini tahmin etmesine ve bu sayede atağın zaman

karmaşıklığının azalmasına neden olacaktır. Diferansiyel faktörleri kullanarak SERPENT şifresine Dunkelman ve diğerlerinin yaptığı 10, 11 ve 12 döngülük diferansiyel-lineer atakların zaman karmaşıklığının sırasıyla 4, 4 ve 8'de bire azaltılabileceğini gösteriyoruz. Ayrıca bu ataklardaki diferansiyeli değiştirerek de daha az veri karmaşıklığı ama daha fazla zaman karmaşıklığı gerektirecek şekilde bu atakları değiştiriyoruz.

Anahtar Kelimeler: kriptanaliz, blok şifre, olası olmayan diferansiyel kriptanaliz, rahatsız edilmemiş bit, diferansiyel faktör

To late Mercan Tezcan

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my thesis supervisor Assoc. Prof. Ali Doğanaksoy and co-supervisor Prof. Ersan Akyıldız for their support and guidance.

It is a great pleasure to thank Prof. Ali Aydın Selçuk for his support and motivation. This study would not be complete without his help.

I would like to thank my colleagues and co-authors Dr. Kerem Varıcı and Dr. Onur Özen for their support and motivation.

I am grateful to my parents Hayri Tezcan and Sırma Tezcan and my brother Cem Tezcan for always trusting and supporting me.

I want to thank my friends Seda Özdemir, Sinan Değer, Can Kartoğlu, Bulutay Güneş, and Murat Tolga Ertürk for their friendship and support.

I also would like to thank everyone at Institute of Applied Mathematics and METU Capoeira Society.

Finally, I would like to thank The Scientific and Technological Research Council of Turkey (TÜBİTAK) for supporting part of this research via TÜBİTAK 1001 project numbered 112E101 and titled ”Blok Şifrelerin Olası Olmayan Diferansiyel Kriptanalizi”.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACP	Adaptive Chosen Plaintexts
B	bytes
CP	Chosen Plaintexts
DDT	Difference Distribution Table
En	Reduced Round Encryptions
$\mathbb{F}$	Field
KP	Known Plaintexts
LAT	Linear Approximation Table
LT	Linear Transformation
MA	Memory Accesses
SPN	Substitution Permutation Network

CHAPTER 1

INTRODUCTION

Cryptology, the science of communication secrecy consists of two main components, *cryptography* and *cryptanalysis*. Cryptography is the science of designing secure ciphers and cryptanalysis is the science of analyzing the security of ciphers by trying to find weaknesses in the design. But nowadays, the words cryptology and cryptography are used interchangeably.

A cipher makes a message unreadable to anyone except those having the key by using an algorithm. More formally, let P denote the *message space*, which contains strings of symbols of a predetermined alphabet and C denote the *ciphertext space* which also contains strings of symbols of a predetermined alphabet.

An element p of P is called a *plaintext* and an element c of C is called a *ciphertext*. Let K denote the *key space* that contains strings of predetermined size. An element k of K is called a *key*. A one-to-one function E_e from P to C, which is uniquely determined by e is called an *encryption function*. One-to-one property is necessary since we want to reverse the process. A one-to-one function D_d from $E_e(P) \subset C$ to P, which is uniquely determined by d is called a *decryption function*.

A *cipher* or an *encryption scheme* contains an encryption function E_e and a decryption function D_d where $e, d \in K$ and d is uniquely determined for any e.

If e and d are equivalent or one of them can be easily obtained from the other in a cipher (by "easily", we mean "in logarithmic time"), that scheme is called a *symmetric-key scheme* or *symmetric-key encryption*. Two main symmetric-key encryption schemes are *block ciphers* and *stream ciphers*.

1.1 Block Ciphers

In a block cipher, the message p is divided into fixed length strings which are called *blocks* and the encryption function encrypts a single block at a time. Generally, the encryption is done by iterating the *round function* of the cipher for r many times where r is a predetermined integer.

Theory of block ciphers is well investigated and a lot of block ciphers are proposed.

Although most of these block ciphers have different designs, they can be roughly categorized as *Feistel Networks* and *Substitution-Permutation Networks (SPNs)*.

In Feistel networks, a round consists of dividing the input into two halves, applying the round function to one half using a subkey, exclusive-oring (XOR) the output of the round function with the other half and swapping the two halves. There is no need to do the swapping operation in the last round since it would not have any effect on the security of the cipher. These two halves are referred to as data lines and generalized Feistel ciphers contain more than two data lines. The CLEFIA cipher that is described in Section 1.1.3 is an example for a generalized Feistel structure with four data lines.

Encryption and decryption is identical in Feistel networks except for the order of the subkeys. A Feistel cipher is called *unbalanced* if the divided parts are not of equal size and this kind of constructions are investigated in [64].

SPN uses substitution boxes (S-boxes) and permutation boxes (P-boxes) where an $n \times m$ S-box substitutes n bits with m bits and a P-box permutes the bits. Generally in a SPN a round consists of XORing the input with a subkey, applying S-boxes and then P-boxes. The output of the last round is also XORed with a subkey. The ciphers PRESENT and SERPENT that are described in Sections 1.1.1 and 1.1.2 are examples for SPNs.

The Feistel network is named after Horst Feistel who did important research in this area and proposed Lucifer [70] cipher with Don Coppersmith which is a Feistel network. The Data Encryption Standard (DES) [33], which is a revised version of Lucifer algorithm, is designed by an IBM team in 1974 and it is adopted as national standard in 1977 by National Bureau of Standards (which is known as National Institute of Standards and Technology (NIST) today). DES is an example of a Feistel cipher.

Since the advances in technology result in faster central processing units, 56-bit key of DES was becoming vulnerable to brute force attacks in which the attacker tries every possible key. For this reason, on January 2, 1997, NIST announced a request for candidate algorithms for the Advanced Encryption Standard (AES) which would support 128, 192 and 256-bit keys. 15 algorithms were submitted to the competition and on October 2, 2000, NIST announced that the winner of the AES competition is Rijndael [32], which is designed by Daemen and Rijmen. AES is also an SPN.

1.1.1 PRESENT

PRESENT [39] was designed in 2007 by Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. It was adopted as an International Standard by ISO/IEC 29192-2:2012 [39] for lightweight cryptography, together with CLEFIA.

PRESENT is a 31-round SPN (Substitution Permutation Network) type block cipher with block size of 64 bits that supports 80 and 128-bit secret key. We represent k-bit keyed PRESENT as PRESENT-k. Round function of PRESENT, which is depicted in Figure 1.1, is same for both versions of PRESENT and consists of standard operations

such as subkey XOR, substitution and permutation: At the beginning of each round, 64-bit input of the round function is XORed with the subkey. Just after the subkey XOR, 16 identical 4×4-bit S-boxes $S(x)$ are used in parallel as a non-linear substitution layer and finally a permutation is performed so as to provide diffusion.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

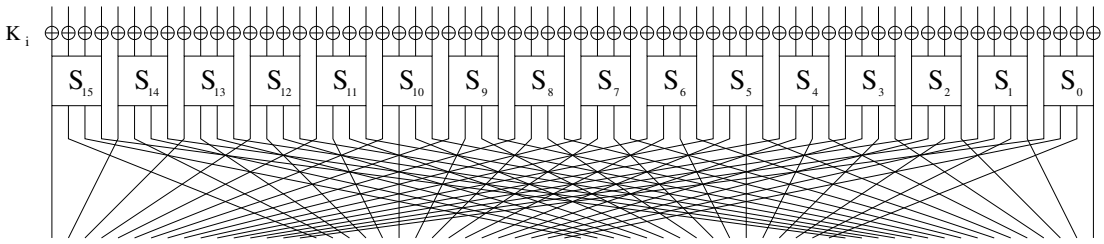

Figure 1.1: Round function of PRESENT

The subkeys for each round are derived from the user-provided secret key by the key scheduling algorithm. We provide only the details of the key scheduling algorithm of PRESENT-80 as it is the version we attack in this thesis: 80-bit secret key is stored in a key register K and represented as $k_{79}k_{78}\ldots k_0$. The subkeys K_i $(0 \leq i \leq 31)$ consist of 64 leftmost bits of the actual content of register K. After round key K_i is extracted, the key register K is rotated by 61 bit positions to the left, then S-box is applied to the left-most four bits of the key register and finally the round counter value, which is a different constant for each round, is XORed with bits $k_{19}k_{18}k_{17}k_{16}k_{15}$. Further details about the specification of PRESENT are provided in [23].

1.1.2 SERPENT

SERPENT [3] was designed by Anderson, Biham and Knudsen in 1998. It was submitted to AES contest and came second after Rijndael. It has a block size of 128 bits and accepts any key size of length 0 to 256 bits. It is a 32-round SPN, where each round consists of key mixing, a layer of S-boxes and a linear transformation.

The 128-bit input value before round i is denoted by $\hat{B}_i$, $i \in \{0, \ldots, 31\}$. Each $\hat{B}_i$ is composed of four 32-bit words X_0, X_1, X_2, X_3 where X_0 is the leftmost word.

Three round operations are specified as follows:

1. Key Mixing: At each round R_i, a 128-bit subkey K_i is XORed with the current intermediate data $\hat{B}_i$.

2. S-boxes: At each round, R_i uses a single S-box S_j, where $i \equiv j \pmod 8$ and $i \in \{0, \ldots, 31\}$, 32 times in parallel. In this thesis, we use the bitsliced version of SERPENT. For example, in the first round the first copy of S_0 takes the

least significant bits from X_0, X_1, X_2, X_3 and returns the output to the same bits. Thus, we obtain 32 4-bit slices referred as b_i's, where $i \in \{0, \ldots, 31\}$ and b_0 is the right most slice.

3. Linear Transformation: The four 32-bit words X_0, X_1, X_2, X_3 are linearly mixed by the following linear operations:

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

where $\lll$ denotes the left rotation operation and $\ll$ denotes the left shift operation.

32-round SERPENT cipher may be described by the following equations:

$$\hat{B}_0 := P \qquad \hat{B}_{i+1} := R_i(\hat{B}_i),\ i \in \{0, \ldots, 31\} \qquad C := \hat{B}_{32}$$

where

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

and $\hat{S}_i$ is the application of the S-box $S_{(i \ (mod\ 8))}$ 32 times in parallel, and LT is the linear transformation.

The key scheduling algorithm of SERPENT takes a 256-bit key as an input. If the key is shorter, then it is padded by a single bit of 1 and the remaining part is padded by bits of 0 up to 256 bits. By using an affine recurrence, the 256-bit key is used to construct 132 *prekeys* having length of 32 bits. The S-boxes are used to produce 32-bit keywords from prekeys. The round keys are obtained by combining these keywords.

1.1.3 CLEFIA

We use the notations provided in Table 1.1 in the following sections.

CLEFIA [68] was designed in 2007 by SONY Corporation and was adopted as an International Standard by ISO/IEC 29192-2:2012 [39] for lightweight cryptography, together with PRESENT.

Table 1.1: Notation

$a_{(b)}$	b denotes the bit length of a
$a\|b$	Concatenation of a and b
$[a, b]$	Vector representation of a and b
a^t	Transposition of a vector a
$a \oplus b$	Bitwise exclusive-OR (XOR) of a and b
$[x^{\{i,0\}}, x^{\{i,1\}}, x^{\{i,2\}}, x^{\{i,3\}}]$	i-th round output data
Δa	XOR difference for a

CLEFIA is a 128-bit block cipher having a generalized Feistel structure with four data lines. For the key lengths of 128, 192, and 256 bits, CLEFIA has 18, 22, and 26 rounds. Each round contains two parallel F functions, F_0 and F_1 and their structures are shown in Fig. 1.2 where S_0 and S_1 are 8×8-bit S-boxes. The two matrices M_0 and M_1 that are used in the F-functions are defined as follows.

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

The encryption function uses four 32-bit whitening keys (WK_0, WK_1, WK_2, WK_3) and $2r$ 32-bit round keys ($RK_0, \ldots, RK_{2r-1}$) where r is the number of rounds. We represent the bytes of a round key as $RK_i = RK_{i,0}|RK_{i,1}|RK_{i,2}|RK_{i,3}$. The encryption function ENC_r is shown in Fig. 1.3.

1.2 Cryptanalysis of Block Ciphers

One might choose to keep the encryption algorithm secret to increase the security. However in history, it is observed that secret algorithms obtained by reverse engineering, betrayal and espionage. Hence it is a good idea to assume that the security of the encryption algorithm should rely on the secrecy of the key, which is also know as the *Kerckhoffs' Principle*.

The most trivial way to attack a block cipher is to try every key in the key space. This is known as *exhaustive search* or *brute force attack*. This can be done by obtaining a few plaintexts and corresponding ciphertexts and encrypting these plaintexts by every possible key (this makes the attack a known-plaintext attack). If a key encrypts the plaintext to the previously known ciphertext, then that key becomes a candidate but sometimes it may not be the correct secret key. Such a case is called a *false alarm*. This is why we need to test that candidate key on more than one plaintext. A similar attack can be done by only decrypting the ciphertexts and checking whether the obtained plaintexts are something meaningful in the language that the plaintexts are suspected to be written. In that sense the attack becomes a ciphertext-only attack.

Note that exhaustive search is a generic method and it can be applied to any block

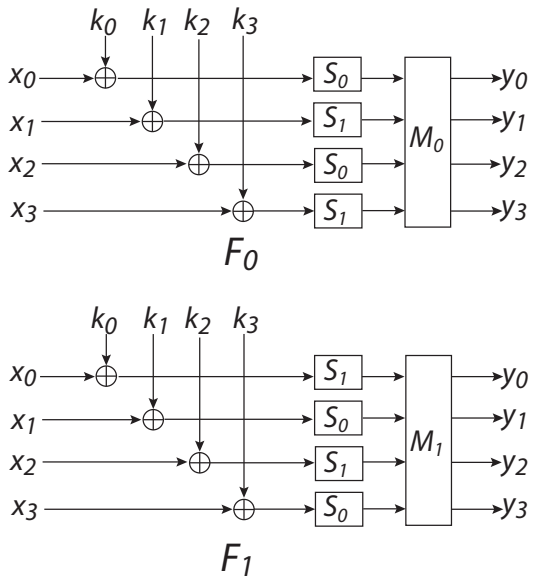

Figure 1.2: F_0 and F_1 functions

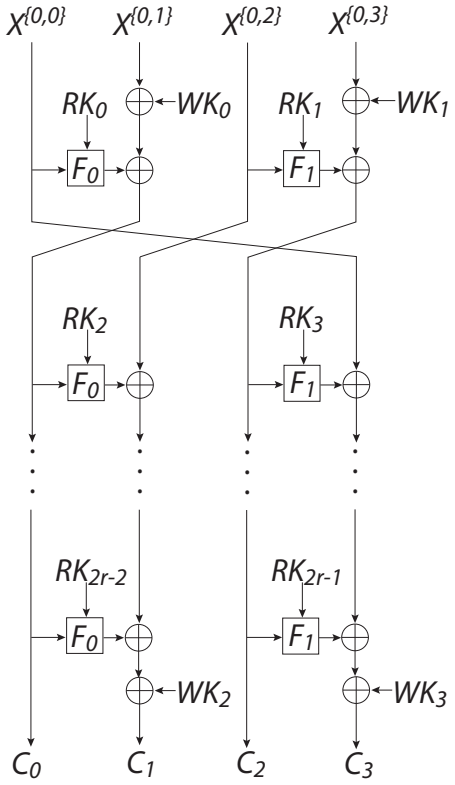

Figure 1.3: Encryption function

cipher. For this reason the key space is kept large in the design of block ciphers to avoid brute force attacks. The length of the key depends on the computational power of computers which depends on the current technology.

If the block size of a block cipher is b and if we know 2^b plaintexts and corresponding ciphertexts, this means that although we do not know the key, given a plaintext we can find the corresponding ciphertext or vice versa. This generic attack is known as *dictionary attack* or *table attack*.

A cipher is considered *broken* if an attack is given which finds the secret key faster than exhaustive search and uses less data than dictionary attack. If the attack is still infeasible, it is called a *theoretical attack* and otherwise, it is called a *practical attack*.

Most of the attacks in the literature requires additional knowledge about the system which determines the type of the attack:

- **Ciphertext-only attack (CO):** In this kind of attacks, the attacker has the knowledge of some ciphertexts which are encrypted by the unknown secret key.
- **Known-plaintext attack (KP):** In known-plaintext attacks, the attacker has the knowledge of some plaintexts and corresponding ciphertexts.
- **Chosen-plaintext (ciphertext) attack (CP):** In this scenario, the attacker has the knowledge of some plaintexts having some particular structure of her choice and corresponding ciphertexts. Similarly for chosen-ciphertext attack, the attacker has the knowledge of some ciphertexts having some particular structure of her choice and corresponding plaintexts.
- **Adaptive chosen-plaintext (ciphertext) attack (ACP):** This type of attack is similar to chosen plaintext attack. However this time, the chosen plaintexts depends on the results of the previous encryptions of plaintexts. Similarly for adaptive chosen-ciphertext attack, the chosen ciphertexts depends on the results of the previous decryptions of ciphertexts.

In the following sections we discuss some non-generic cryptanalysis techniques and the following definitions are necessary for these sections:

Definition 1.1. Let b be the block size of a block cipher. Then under a key, the encryption function becomes a bijective function from a set with 2^b elements to itself. In order to find a weakness for a cipher, the encryption function is compared with a random bijective function from and to the same set with 2^b elements. In the concept of cryptanalysis, such a function is referred to as a *random permutation*.

Definition 1.2. Statistical attacks on block ciphers make use of a property of the cipher so that an event occurs with a different probability than a random permutation. Such a property is called a *distinguisher* since it can be used to distinguish the cipher from a random permutation.

Definition 1.3. Distinguishing a cipher from a random permutation is referred to as a *distinguishing attack*.

Distinguishers or distinguishing attacks are important in two ways:

1. In some attack scenarios, the attacker may capture some input and output from the system but does not know which cipher is used or whether these outputs come from a cipher at all. In such a case, if the attacker has a distinguisher for a cipher and captured enough data, then they can use it to determine whether they belong to the cipher or not.
2. If an attacker captures some plaintexts and corresponding ciphertexts that are encrypted via a secret key, it may be possible for the attacker to extract some information about the secret key by extending the distinguisher, generally by adding a few rounds to the top or the bottom of the distinguisher. Hence, most attacks start with a distinguisher.

1.2.1 On Hiding a Plaintext Length

Note that in a block and stream ciphers, the length of the plaintext and the length of the corresponding ciphertext are the same. Although this does not introduce a weakness in general, in certain cases information may leak because of this. That is, if an eavesdropper listens to an encrypted communication and if they know what the communication is about, then they can guess what the plaintext is by looking at the length of the ciphertext. For example, when some forms on the internet are filled by a user, the data is sent encrypted. Since the eavesdropper knows the questions asked in the form, by looking at the lengths of the answers, they can guess what the answers are.

In order to hide the length of the plaintexts, some bits in random lengths can be appended to the end of the plaintexts. We investigated how much security we get for appending random length bits in [79]. This kind of investigation is beyond the scope of this thesis and will not be further considered.

1.3 Complexity

Attacks are compared with the amount of resources they require. These resources are defined as data complexity, time complexity and memory complexity:

- **Data Complexity:** The amount of plaintexts or ciphertexts that is required to perform the attack.
- **Memory Complexity:** The amount of storage required to perform the attack.
- **Time Complexity:** The amount of time required to perform the attack. Most of the time, it is measured by the number of encryptions or memory accesses.

In the case of exhaustive search, if the secret key is m bits, then the time complexity is 2^m encryptions. The attack requires a few plaintexts or ciphertexts and the storage

is required only for the values that are used in the encryption (or decryption) process. Hence the data and memory complexities are negligible.

In the case of dictionary attack, if we put every ciphertext and plaintext in a table, such an attack has 2^b data complexity, 2^b memory complexity and negligible time complexity.

1.4 Differential Cryptanalysis

Differential cryptanalysis [12] was discovered by Biham and Shamir in late 1980s and it is used to attack various block ciphers, stream ciphers and hash functions. Although it is claimed that agencies of some countries already knew this technique years before its discovery by Biham and Shamir, theirs is the first public announcement of this method. This technique breaks DES theoretically.

Differential cryptanalysis is a statistical chosen-plaintext attack and it considers differential relations between inputs and outputs for r consecutive rounds, for some integer r.

When two different inputs are encrypted with the secret key, the probability of the difference of the corresponding outputs to be β, for some β, is $p = 2^{-b}$ where b is the block size. If an α difference in input blocks results in β difference in the output blocks after r rounds of encryption with a probability p_0 higher than 2^{-b}, we call this relation an r-round differential *characteristic*. A differential characteristic with high probability is used to distinguish the correct subkeys from the wrong ones.

Today, differential cryptanalysis plays an important role in the design of blocks ciphers and designers make their algorithms resistant to this attack by giving an upper bound to the probability of r-round differentials [58].

In 1994, Knudsen discovered truncated differential cryptanalysis [41] which is an extension of differential cryptanalysis in which the differences are not fully specified.

1.5 Truncated Differential Cryptanalysis

In 1994, Knudsen discovered truncated differential cryptanalysis [41] which is an extension of differential cryptanalysis in which the differences are not fully specified. Because of this change, main difference between classical differential attacks and truncated differential attacks is the ratio of p and p_0. That is, most of the time we have $\frac{p_0}{p} \geq 4$ in differential attacks and $\frac{p_0}{p} \approx 1$ in truncated differential attacks.

1.6 Differential-Linear Cryptanalysis

In 1994, Langford and Hellman combined differential cryptanalysis with linear cryptanalysis and introduced differential-linear cryptanalysis [47]. They suggested using a truncated differential with probability 1 and concatenating a linear approximation with bias q (i.e. probability $1/2 + q$) where the output difference of the differential should contain zero differences in the places where input bits masked in the linear approximation. This way one can construct differential-linear distinguishers and the data complexity of the distinguisher is $O(q^{-4})$ chosen plaintexts. The exact number depends on the success probability and the number of possible subkeys.

Moreover, Biham, Dunkelman and Keller showed that it is possible to construct a differential-linear distinguisher where the differential holds with probability $p < 1$ and introduced enhanced differential-linear cryptanalysis [9]. They also showed that the attack is still applicable if the XOR of the masked bits of the differential is 1. In the enhanced method, the data complexity becomes $O(p^{-2}q^{-4})$ chosen plaintexts.

1.7 Impossible Differential Cryptanalysis

The cryptanalytic technique of impossible differential attack is discovered by Biham, Biryukov and Shamir and it is first presented at Rump Session of CRYPTO 1998 by Shamir [4]. Later on in [6], they presented this technique by giving an attack that breaks Skipjack [57] reduced from 32 to 31 rounds. They also used this technique to break reduced round version of IDEA [46] and Khufu [55] in [5]. Independently in 1998, in his proposal [40] for AES, Knudsen gave an attack to 6-round DEAL [40] which is similar to impossible differential cryptanalysis.

Impossible differential cryptanalysis uses an impossible differential which is a truncated differential that holds with probability 0. One way of obtaining such a differential is the *miss-in-the-middle* technique, that is the combination of two truncated differentials both of which hold with probability 1 and do not meet in the middle. That is, if a difference α becomes β after r_1 rounds of encryption and a difference δ becomes γ after r_2 rounds of decryption and if $\beta \neq \gamma$, we conclude that the difference α cannot become δ after $r_1 + r_2$ rounds of encryption. i.e. $\alpha \rightarrow \beta \neq \gamma \leftarrow \delta$.

An impossible differential obtained with a miss-in-the-middle technique works as a *sieve* in the procedure. If under a subkey that impossible differential holds, it means that the corresponding subkey is not the correct subkey and we eliminate it.

As in the case of differential cryptanalysis, impossible differential attacks are chosen plaintext attacks.

An impossible differential on r rounds of a block cipher can be used to distinguish a random permutation f from r-round version of that cipher. Assume an α difference cannot produce β diffrence after r rounds of encryption. If an input pair of f has the difference α and the corresponding output difference is β, then it is obvious that f

is not the r-round version of the cipher. If difference β is not observed, the number of pairs should be increased enough to be sure that f is not a random permutation. This number of pairs depends on the block size of the cipher and the structure of the differential.

For example, assume that the block size is n and the β difference has fixed k bits (hence $n-k$ bits of β can be anything). For a random pair, the probability of not observing the β difference is $1-2^{-k}$. Therefore if we use 2^p pairs and do not observe the β difference, the probability of incorrectly identifying a random permutation as the r-round version of the cipher becomes $(1-2^{-k})^{2^p}$. Hence, we must choose the value of p larger than k to make this probability close to 0. For small values of k, this probability can be calculated easily with a computer. For large values, the following approximation can be used:

$$\left(1-\frac{1}{2^k}\right)^{c\cdot 2^k} = \left(\left(1-\frac{1}{2^k}\right)^{2^k}\right)^c \approx \left(\frac{1}{e}\right)^c = \frac{1}{e^c}. \tag{1.1}$$

This approximation can be obtained by substituting -1 for x in the formal limit definition of exponential function which is

$$e^x = \lim_{n\to\infty}\left(1+\frac{x}{n}\right)^n. \tag{1.2}$$

1.8 Structures

When constructing plaintext pairs, the idea of *structures* is generally used in differential attacks. Assume that the α difference has x many 0 bits and y many undetermined bits and the block size n is $x+y$. Fixing the bits in the places where α has no difference is called a *structure*. Now we can construct 2^y different blocks which is in this structure and any two blocks of this structure has difference α. Hence the maximum number of pairs we can obtain from this structure is

$$\binom{2^y}{2} = \frac{2^y\cdot(2^y-1)}{2} = 2^{2y-1}-2^{y-1} \approx 2^{2y-1}. \tag{1.3}$$

Since we have 2^x different structures, approximately $2^{2y-1}\cdot 2^x = 2^{2y+x-1}$ different pairs can be constructed at most. Note that if we replace one or more of the zeros with ones in the difference α, the maximum number of pairs that can be obtained becomes exactly 2^{2y+x-1}.

1.9 Data Complexity and Success Probability Estimates

Statistical attacks on block ciphers make use of a property of the cipher so that an event occurs with a different probability than a random permutation and such a property is

called a distinguisher since it can be used to distinguish the cipher from a random permutation.

Moreover, if an attacker captures some plaintexts and corresponding ciphertexts that are encrypted via a secret key, it may be possible for the attacker to extract some information about the secret key by extending the distinguisher, generally by adding a few rounds to the top or the bottom of the distinguisher. Hence, most attacks start with a distinguisher.

For instance, differential cryptanalysis [41] considers characteristics or differentials which show that a particular output difference should be obtained with a relatively high probability when a particular input difference is used. Hence, when the correct key is used, the predicted differences occur more frequently. In a classical differential characteristic the differences are fully specified and in a truncated differential [41] only parts of the differences are specified.

On the other hand, impossible differential cryptanalysis [6] uses an impossible differential which shows that a particular difference cannot occur for the correct key (i.e. probability of this event is exactly zero). Therefore, if these differences are satisfied under a trial key, then it cannot be the correct one. Thus, the correct key can be obtained by eliminating all or most of the wrong keys.

The main idea behind the statistical attacks is to gather N plaintext-ciphertext pairs and check the occurrence of the distinguisher when the pairs are partially encrypted or decrypted under every possible subkey. Let p_0 denote the probability of observing the distinguisher when a pair is tried with the correct key and let p be the probability of satisfying the distinguisher for a random permutation. In order to perform the attack, we assume that the wrong keys act like a random permutation. That is, the probability of observing the distinguisher is the same, namely p, for every wrong subkey. This assumption is commonly referred to as the Wrong-Key Randomization Hypothesis.

Thus, if we keep the count of obtaining the distinguisher for every key, the number of hits a wrong subkey gets can be seen as a random variable of binomial distribution X with parameters N and p. Similarly, the number of hits the correct subkey gets can be seen as a random variable of binomial distribution X_0 with parameters N and p_0. It is possible to distinguish the cipher from a random permutation or distinguish the correct subkey from the wrong ones when the difference between the expected values of these two distributions $E = N \cdot p$ and $E_0 = N \cdot p_0$ is big enough. That is, we would like to have a threshold T somewhere between E and E_0 and expect the counter of the correct key to be on the side of E_0 together with none or a small number of counters corresponding to wrong keys. Thus, the keys with counters on the right side of the threshold are candidates for the correct key and the correct one can be obtained by exhaustively trying every possibility for the remaining key bits that are not attacked.

Definition 1.4. The case of the correct key having a counter on the wrong side of the threshold T is called *non-detection* and its probability is denoted by p_{nd}.

Definition 1.5. The case of a wrong key having a counter on the right side of the threshold T is called *false alarm* and its probability is denoted by p_{fa}.

Note that an attack is successful if the correct key's counter end up in the right side of the threshold. Thus, the success probability of an attack can be calculated as $1 - p_{nd}$. Moreover, for every false alarm, we need to do the same work that we do for the correct key. So having high p_{fa} increases the time complexity of the attack. Therefore, the attacker desires p_{nd} and p_{fa} to be very small. Increasing N decreases non-detection and false alarm probabilities because the difference between E and E_0 gets larger. However, increasing N also means increasing the data complexity of the attack. Since we repeat the attack process for every plaintext-ciphertext pair, increasing N also results in increased time complexity. Therefore, there should exist a value of N that provides the optimal time complexity.

In most of the statistical attacks the distinguisher is more probable for the cipher than a random permutation. In other words, we have $p_0 > p$. Impossible differential attack is an exception for this case since $p_0 = 0$ in impossible differential attacks. In this thesis, we will introduce a new cryptanalytic technique that we call improbable differential cryptanalysis in which we have $p_0 < p$. Thus, impossible differential cryptanalysis is just a special case of the improbable differential cryptanalysis. Since all the previous work on estimating data complexity and success probability is done for the case of $p_0 > p$, for the rest of this section we assume $p_0 > p$.

Since the counters we keep for the keys follow binomial distributions, we can calculate p_{nd} and p_{fa} as follows:

$$p_{nd} = \sum_{i=0}^{T} \binom{N}{i} p_0^i \cdot (1 - p_0)^{N-i} \tag{1.4}$$

$$p_{fa} = \sum_{i=T}^{N} \binom{N}{i} p^i \cdot (1 - p)^{N-i} \tag{1.5}$$

In order to find an optimal N for a desired success probability, one needs to compute these probabilities several times. However, for theoretical attacks N can be very large and p_0 and p can be very small. For instance, to attack a block cipher with a block size of b bits, N can be as large as 2^{2b-1} and p and p_0 can be as small as 2^{-b}. Therefore, it may be infeasible to compute these probabilities for many parameters. Thus, we require fast formulas to estimate p_{fa} and p_{nd}.

In [59], Biham and Shamir observed a strong relation between the signal-to-noise ratio and the success chance of an attack. Signal-to-noise ratio can be defined as follows:

Definition 1.6 ([66])**.** An important measure for the success of a differential attack is the proportion 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. This proportion is called the *signal-to-noise ratio*.

In most of the attacks, directly finding the correct key requires more data and time than eliminating most of the wrong keys. For this reason, we define advantage:

Definition 1.7 ([66])**.** If an attack on an m-bit key gets the correct value ranked among the top r out of 2^m possible candidates, we say the attack obtained an $(m - log(r))$-bit *advantage* over exhaustive search.

Although linear [52] and differential cryptanalysis [12] were introduced as early as the beginning of 1990s, they lacked a robust treatment of their success probability until Selçuk's analytical calculations [66]. Selçuk uses normal approximation of the binomial law to provide formulas of the success probability and he considers "success" as the case where the correct key is within a set of high-ranking candidates, which is referred as *advantage*. Selçuk's formulas for the differential cryptanalysis are as follows:

Theorem 1.1 ([66])**.** *Let p_s be the success probability that a differential attack on an m-bit key, with a characteristic of probability p_0 and signal-to-noise ratio S_N and with N plaintext-ciphertext pairs, delivers an a-bit or higher advantage. Assuming that the key counters are independent and that they are identically distributed for all wrong keys, we have, for sufficiently large m and N, and μ denoting $p_0 N$,*

$$p_s = \Phi\left(\frac{\sqrt{\mu S_N} - \Phi^{-1}(1 - 2^{-1})}{\sqrt{S_N + 1}}\right). \tag{1.6}$$

Above formula provides the success probability when the number of plaintext-ciphertext pairs N is fixed. It can be modified to obtain a formula that provides the required N to perform a differential attack with a fixed success probability p_s as follows:

Corollary 1.2 ([66])**.** *With the assumptions of Theorem 1.1,*

$$N = \frac{(\sqrt{S_N + 1}\Phi^{-1}(p_s) + \Phi^{-1}(1 - 2^{-a}))^2}{S_N} p_0^{-1} \tag{1.7}$$

plaintext-ciphertext pairs are needed in a differential attack to accomplish an a-bit advantage with a success probability p_s.

One main difference between classical differential attacks and truncated differential attacks is the ratio of p and p_0. That is, most of the time we have $\frac{p_0}{p} \geq 4$ in differential attacks and $\frac{p_0}{p} \approx 1$ in truncated differential attacks. This is one of the reasons why Selçuk's formulas are not applicable to truncated differential attacks. We also have $\frac{p_0}{p} \approx 1$ for the improbable differential attacks that we are going to introduce in Chapter 4 of this thesis and therefore, we cannot use these formulas for improbable differential cryptanalysis, too.

In [18, 20], Blondeau *et al.* provided accurate estimates of the data complexity and success probability for linear, differential and truncated differential attacks. We first recall the Kullback-Leibler divergence which is used in Blondeau *et al.*'s estimations.

Definition 1.8 (Kullback - Leibler divergence [29])**.** Let P and Q be two Bernoulli probability distributions of parameters p and q. The Kullback - Leibler divergence between P and Q is defined by

$$D(p||q) = p \ \ln\left(\frac{p}{q}\right) + (1 - p) \ \ln\left(\frac{1 - p}{1 - q}\right). \tag{1.8}$$

Instead of using a normal approximation, Blondeau *et al.* relied on the following simple and general approximation of the binomial distribution.

Theorem 1.3 ([2]). *Let p_0 and p be two real numbers such that $0 < p < p_0 < 1$ and let τ such that $p < \tau < p_0$. Let X_0 and X follow a binomial law of respective parameters (N, p_0) and (N, p). Then as $N \to \infty$,*

$$P(X \geq \tau N) \sim \frac{(1-p_0)\sqrt{\tau}}{(\tau-p)\sqrt{2\pi N(1-\tau)}} e^{-ND(\tau||p)}, \tag{1.9}$$

and

$$P(X_0 \leq \tau N) \sim \frac{p_0\sqrt{1-\tau}}{(p_0-\tau)\sqrt{2\pi N\tau}} e^{-ND(\tau||p_0)}. \tag{1.10}$$

In order to obtain the required number of plaintext-ciphertext pairs N and the threshold T for many cryptanalytic techniques, the following Blondeau-Gerard-Tillich algorithm takes p, p_0, p_{fa}, and p_{nd} as input and outputs N and τ, which is a relative threshold $\tau = \frac{T}{N}$. In this algorithm, the estimates for non-detection and false alarm error probabilities are denoted by $G_{nd}(N, \tau)$ and $G_{fa}(N, \tau)$.

Algorithm 1.1. ([18])

Input: p_0, p, p_{nd}, p_{fa}
Output: N, τ
$\tau_{min} := p$, $\tau_{max} := p_0$
repeat
  $\tau := \frac{\tau_{min}+\tau_{max}}{2}$
  Compute N_{nd} such that $\forall N > N_{nd}$, $G_{nd}(N, \tau) \leq p_{nd}$
  Compute N_{fa} such that $\forall N > N_{fa}, G_{fa}(N, \tau) \leq p_{fa}$
  if $N_{nd} > N_{fa}$ **then** $\tau_{min} = \tau$
  else $\tau_{max} = \tau$
until $N_{nd} = N_{fa}$
$N := N_{nd}$
Return N, τ

Blondeau *et al.* assumed $p_0 > p$ in [18] because improbable differential cryptanalysis was not introduced yet. Thus, this algorithm cannot be used directly for improbable differential cryptanalysis. After introducing improbable differential cryptanalysis in Chapter 4, we also modify these results to obtain accurate estimates of data complexity and success probability for improbable differential cryptanalysis.

1.10 S-box Analysis

S-boxes are commonly used as non-linear components for symmetric cryptosystems and hash functions. Properties of S-boxes provide resistance against many cryptanalytic techniques.

1.10.1 Differential Uniformity

Definition 1.9. For a mapping $S : F_2^n \rightarrow \mathbb{F}_2^m$, and all $\Delta_i \in \mathbb{F}_2^n$ and $\Delta_o \in \mathbb{F}_2^m$, let t be the number of elements x that satisfy $S(x \oplus \Delta_i) = S(x) \oplus \Delta_o$. Then $t/|2^n|$ is the differential probability of the characteristic $S(\Delta_i \rightarrow \Delta_o)$. The table that lists all t values for every $i, o \in X$ is called the *Difference Distribution Table (DDT)*.

The maximum value in a DDT, excluding the zero difference case, is called differential uniformity. S-box designers aim to minimize differential uniformity since differential cryptanalysis [12] uses characteristics with high differential probability.

1.10.2 Non-linear Uniformity

Definition 1.10. For a mapping $S : \mathbb{F}_2^n \rightarrow \mathbb{F}_2^m$, and all $a \in \mathbb{F}_2^n$ and $b \in \mathbb{F}_2^m$, let the numbers $L_f(a, b)$ be defined as

$$L_f(a, b) := |\#\{x \in F_2^n | a \cdot x = b \cdot S(x)\} - 2^{n-1}|$$

where $a \cdot b$ denotes the parity of the bit-wise product of a and b. Then S is called *non-linearly l-uniform* if $L_f(a, b) \leq l$ for all a and b with $b \neq 0$.

S-box designers aim to minimize the non-linear uniformity l since linear cryptanalysis [52] uses linear approximations with high bias.

1.10.3 Branch Number

Definition 1.11. [63] The branch number of an $n \times n$ S-box is

$$BN = \min_{a, b \neq a}(wt(a \oplus b) + wt(S(a) \oplus S(b))),$$

where $a, b \in X$ and $wt(a)$ is the Hamming weight of the bit vector a.

For a bijective S-box, the branch number is at least 2 and this property of S-boxes is closely related to algebraic [28] and cube attacks [34].

1.10.4 Number of Shares

S-boxes are also studied for their security against side-channel attacks. Side-channel attacks are based on the information leakage during the computation of the hardware implementation of a cryptographic algorithm. For instance, differential power analysis (DPA) [44] exploits the correlation between the instantaneous power consumption of a device and the intermediate results of a cryptographic algorithm. One countermeasure

against side-channel attacks is threshold implementation in which a variable is split into additive shares. Bilgin *et al.* analyzed the number of shares of S-boxes by categorizing all 3×3 and 4×4 S-boxes using affine equivalence classes and investigated the cost of this kind of protection in [14].

In Chapters 2 and 3, we are going to introduce two new S-box criteria that are related to differential, differential-linear, truncated differential, impossible differential, and improbable differential cryptanalysis.

1.11 Our Contribution and the Structure of the Thesis

In most of the differential cryptanalysis techniques, the distinguisher is more probable for the cipher than a random permutation. In other words, we have $p_0 > p$. The only exception is the impossible differential cryptanalysis in which we have $p_0 = 0$. In this thesis, we bridge the gap by introducing improbable differential cryptanalysis in which we have $p_0 < p$.

Moreover, we introduce two new S-box evaluation criteria:

1. **Undisturbed Bits:** Undisturbed bits are probability 1 truncated differentials for S-boxes and they can be used for constructing longer or better truncated, impossible, or improbable differentials.
2. **Differential Factors:** Differential factors are key differences for ciphers with key XOR before the S-box where the output difference of the S-box is invariant. An attacker cannot capture the whole key when there is a differential factor but this also reduces the work of the attacker.

This thesis is organized as follows: Undisturbed bits are introduced in Chapter 2 and S-boxes that are used in cryptographic algorithms that contain undisturbed bits are listed. Differential factors are introduced in Chapter 3 and S-boxes that are used in cryptographic algorithms that contain differential factors are listed. In Chapter 4, we introduce the improbable differential cryptanalysis and modify Blondeau *et al.*'s algorithm and formulas to have accurate estimates for the data complexity and success probability of improbable differential attacks. In Chapter 5, we provide 12 and 13-round improbable differential attacks on PRESENT using the undisturbed bits of its S-box. In Chapter 6, we provide 12, 13, and 14 round improbable differential attacks on CLEFIA which are the best known attacks on this cipher. In Chapter 7, we provide a 7-round improbable differential attack on SERPENT using the undisturbed bits of its S-boxes. Moreover, we correct the advantage and improve the time complexity of Dunkelman *et al.*'s differential-linear attacks by using the differential factors of SERPENT.

CHAPTER 2

UNDISTURBED BITS

We introduce *undisturbed bits* in this chapter. They are probability 1 truncated differentials for S-boxes and they can be used for constructing longer or better truncated, impossible, or improbable differentials. We published parts of this chapter in [77].

Definition 2.1. Depending on the design of an S-box, when a specific difference is given to the input (resp. output), difference of at least one of the output (resp. input) bits of the S-box may be guessed with probability 1. We call such bits *undisturbed*.

We start by showing that every invertible 3×3 S-box contains undisturbed bits.

Proposition 2.1. *Every bijective* 3×3 *S-box contains undisturbed bits.*

Proof. There are $8! = 40320$ different bijective 3×3 S-boxes. If we count the number of undisturbed bits of an S-box together with the undisturbed bits of its inverse, one can check that 17088 of them have 6, 10368 of them have 12, 6336 of them have 18, 3456 of them have 24, 1728 of them have 30, and 1344 of them have 42 undisturbed bits. □

In the proof, we also considered the undisturbed bits of the inverse of the S-boxes because in SPNs, the inverse of the S-box is used for decryption. Note that in [77], we tried to give a simple, non-computer aided proof for Proposition 2.1 by considering the all 4 equivalence classes of bijective 3×3 S-boxes. In that proof, we made the assumption that every 3×3 S-box of the same affine equivalence class has the same number of undisturbed bits. However, there are some exceptions to this assumption and they were overlooked in that proof.

PRINTcipher's S-box [43] and SEA's S-box [71] are 3×3 S-boxes that we observed in a cryptographic algorithm and they have 6 undisturbed bits. They are provided in Table 2.1. Although every possible 3×3 S-box contains undisturbed bits, note that a 4×4 S-box does not necessarily have undisturbed bits. For instance, we observed that six out of eight 4×4 S-boxes of the block cipher SERPENT [3] have 30 undisturbed bits in total and two of its S-boxes have no undisturbed bits. In our literature search we found 101 4×4 S-boxes that are used in block ciphers and hash functions and observed that 68 of them have 393 undisturbed bits in total. They are provided at the end of this chapter in Table 2.7. Note that some S-boxes are re-used by different algorithms. For

instance, S-box of PRESENT is also used in LED [38] and one of SERPENT's S-box S_2 is also used in HAMSI [86].

Moreover, we analyzed the 6×4 S-boxes of DES and 8×8 S-boxes of AES, Anubis, Aria, Camellia, CLEFIA, Crypton, E2, Grand Cru, Hierocrypt, ICEBERG, Khazad, Sarmal, SEED, and SMS4 and we did not observe any undisturbed bits. Finally, we tested the 5×5 and 6×6 S-boxes of FIDES and 7×7 and 9×9 S-boxes of KASUMI and MISTY and observed that FIDES's S-boxes and the 9×9 S-boxes of KASUMI and MISTY contain undisturbed bits. They are listed in Tables 2.2, 2.3, and2.4.

Table 2.1: Undisturbed Bits of 3×3 S-boxes

S-box	Input	Output
PRINTcipher [43]	1_x	??1
PRINTcipher [43]	2_x	?1?
PRINTcipher [43]	4_x	1??
PRINTcipher^{-1} [43]	1_x	??1
PRINTcipher^{-1} [43]	2_x	?1?
PRINTcipher^{-1} [43]	4_x	1??
SEA [71]	1_x	??1
SEA [71]	3_x	?1?
SEA [71]	4_x	1??
SEA^{-1} [71]	2_x	?1?
SEA^{-1} [71]	3_x	??1
SEA^{-1} [71]	4_x	1??

Table 2.2: Undisturbed Bits of 5×5 S-boxes

S-box	Input	Output
FIDES [13]	08_x	????1
FIDES [13]	$0B_x$	??1??
FIDES [13]	10_x	?1???
FIDES [13]	16_x	1????
FIDES [13]	$1A_x$	???1?

Undisturbed bits can be used to construct better truncated, impossible or improbable differentials and they should be avoided by S-box designers to provide more security against these kind of attacks. In order to support our claim, in this section we are going to obtain the longest impossible differentials for PRESENT and SERPENT ciphers using undisturbed bits. Undisturbed bits of the S-boxes of these ciphers are given in Table 2.7.

By using the undisturbed bits of PRESENT's S-box, we obtained a 6-round impossible differential as shown in Table 2.5 where a ? denotes an indeterminate value and x's in a row means that at least one of them is non-zero. Without using the undisturbed bits,

Table 2.3: Undisturbed Bits of 6×6 S-boxes

S-box	Input	Output
FIDES [13]	25_x	?1????
FIDES [13]	30_x	???1??
FIDES $^{-1}$ [13]	$2A_x$	????1?

Table 2.4: Undisturbed Bits of 9×9 S-boxes

S-box	Input	Output
KASUMI [1]	8_x	????????1
KASUMI [1]	40_x	???????1?
KASUMI [1]	2_x	??????1??
KASUMI [1]	20_x	?????1???
KASUMI [1]	10_x	????1????
KASUMI [1]	4_x	???1?????
KASUMI [1]	1_x	??1??????
KASUMI [1]	100_x	?1???????
KASUMI [1]	80_x	1????????
MISTY [53]	$1ff_x$	????????1
MISTY [53]	80_x	???????1?
MISTY [53]	100_x	??????1??
MISTY [53]	1_x	?????1???
MISTY [53]	2_x	????1????
MISTY [53]	4_x	???1?????
MISTY [53]	8_x	??1??????
MISTY [53]	20_x	?1???????
MISTY [53]	10_x	1????????

the longest impossible differential we could find for PRESENT had a length of 5-rounds with difference of the 60 out of 64 bits are fixed which means $p = 2^{-60}$. However, by using the 6 undisturbed bits of PRESENT, we can construct 5 and 6-round impossible differentials with only 13 and 39 bits fixed, respectively. In the Chapter 5, by further using the undisturbed bits, we construct a 5-round differential with probability $2^{-17.84}$ that can be combined with the 5-round impossible differential. Hence, by using the expansion method of Section 4.2, we construct a 10-round improbable differential and provide improbable differential attacks on PRESENT reduced to 12 and 13 rounds.

Without using undisturbed bits, the longest impossible differential we could find on SERPENT had a length of 3.5 rounds. However, we obtained four 5.5-round impossible differentials on SERPENT with the help of undisturbed bits and one of them is shown in detail in Table 2.6. Here S_i's are the differences after the S_i operations, LT represents the differences after the linear transformation and question marks represent indeterminate bit differences. The miss-in-the-middle is observed at the 13th bit of X_3

Table 2.5: A 6-Round Impossible Differential for PRESENT

Rounds	Differences in bits															
	x_{15}	x_{14}	x_{13}	x_{12}	x_{11}	x_{10}	x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
$X_{0,I}$	0000	0000	0000	0000	0000	0000	0000	0000	0000	1001	0000	0000	0000	1001	0000	0000
$X_{1,S}$	0000	0000	0000	0000	0000	0000	0000	0000	0000	???0	0000	0000	0000	???0	0000	0000
$X_{1,P}$	0000	0000	0?00	0?00	0000	0000	0?00	0?00	0000	0000	0?00	0?00	0000	0000	0000	0000
$X_{2,S}$	0000	0000	????	????	0000	0000	????	????	0000	0000	????	????	0000	0000	0000	0000
$X_{2,P}$	00??	00??	00??	0000	00??	00??	00??	0000	00??	00??	00??	0000	00??	00??	00??	0000
$X_{3,S}$	????	????	????	0000	????	????	????	0000	????	????	????	0000	????	????	????	0000
$X_{3,P}$	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0
$X_{3,P}$	????	????	????	????	????	????	????	????	????	????	????	????	$???x$	$???x$	$???x$	$???x$
$X_{4,S}$	????	????	????	????	????	????	????	????	????	????	????	????	$000x$	$000x$	$000x$	$000x$
$X_{4,P}$	????	????	????	0000	????	????	????	0000	????	????	????	0000	????	????	????	$xxxx$
$X_{5,S}$	????	????	????	0000	????	????	????	0000	????	????	????	0000	????	????	????	???1
$X_{5,P}$	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???1
$X_{6,S}$	0101	0101	0101	????	0101	0101	0101	????	0101	0101	0101	????	0101	0101	0101	0001
$X_{6,P}$	000?	000?	000?	0000	111?	111?	111?	1110	000?	000?	000?	0000	111?	111?	111?	1111

after round 2. Note that 5.5-round impossible differentials we have found are infeasible to mount an attack since $p = 2^{-128}$. Instead, by eliminating the last round of the 5.5-round impossible differential, we obtained a 4.5-round impossible differential with $p = 2^{-100}$ and used it to construct a 5.5-round improbable differential with probability $p' = 2^{-4}$. We are going to use this improbable differential in Chapter 7 to attack SERPENT reduced to 7 rounds.

Undisturbed bits are also used in [87] to obtain an 8-round impossible differential and show that RECTANGLE cipher is secure against impossible differential cryptanalysis. Moreover, in a different context, undisturbed bits are used in [72] to show that full PRESENT is secure against related-key differential cryptanalysis.

Table 2.7: Undisturbed Bits of 4×4 S-boxes

S-box	Input	Output
CLEFIA SS0^{-1} [68]	A_x	?0??
CLEFIA SS0^{-1} [68]	3_x,9_x	?1??
DES1 Row1^{-1} [59]	A_x	???1
DES1 Row1^{-1} [59]	2_x	??1?
DES1 Row1^{-1} [59]	8_x	1???
DES1 Row2^{-1} [59]	8_x	???1
DES1 Row3^{-1} [59]	5_x	???1
DES1 Row3^{-1} [59]	4_x	??1?
DES1 Row3^{-1} [59]	1_x	1???
DES1 Row4 [59]	F_x	?0?0
DES1 Row4 [59]	4_x,B_x	???1
DES1 Row4 [59]	2_x,D_x	?1??
DES1 Row4^{-1} [59]	8_x	???1
DES1 Row4^{-1} [59]	2_x	1???
DES2 Row1 [59]	7_x	?0??
DES2 Row1 [59]	8_x	??1?

Continued on next page

Table 2.7 – continued from previous page

S-box	Input	Output
DES2 Row1 [59]	1_x,6_x	?1??
DES2 Row1^{-1} [59]	A_x	???0
DES2 Row1^{-1} [59]	2_x,8_x	???1
DES2 Row2 [59]	E_x	???0
DES2 Row2 [59]	6_x,8_x	???1
DES2 Row2 [59]	4_x	1???
DES2 Row2^{-1} [59]	7_x	???0
DES2 Row2^{-1} [59]	F_x	??0?
DES2 Row2^{-1} [59]	1_x,6_x	???1
DES2 Row2^{-1} [59]	7_x,8_x	??1?
DES2 Row3 [59]	5_x	0???
DES2 Row3 [59]	6_x	?1?1
DES2 Row3 [59]	1_x,4_x	1???
DES2 Row3^{-1} [59]	A_x	0???
DES2 Row3^{-1} [59]	2_x,8_x	1???
DES2 Row4^{-1} [59]	2_x	???1
DES2 Row4^{-1} [59]	1_x	??1?
DES2 Row4^{-1} [59]	8_x	?1??
DES2 Row4^{-1} [59]	4_x	1???
DES3 Row1 [59]	E_x	???0
DES3 Row1 [59]	6_x,8_x	???1
DES3 Row1 [59]	7_x	1???
DES3 Row1^{-1} [59]	2_x	1???
DES3 Row2 [59]	E_x	???0
DES3 Row2 [59]	6_x,8_x	???1
DES3 Row2 [59]	4_x	??1?
DES3 Row2^{-1} [59]	F_x	??0?
DES3 Row2^{-1} [59]	2_x,D_x	??1?
DES3 Row2^{-1} [59]	8_x	1???
DES3 Row3 [59]	E_x	?0??
DES3 Row3 [59]	6_x,8_x	?1??
DES3 Row3 [59]	2_x	1???
DES3 Row3^{-1} [59]	6_x	???0
DES3 Row3^{-1} [59]	2_x,4_x	???1
DES3 Row4 [59]	3_x	0??1
DES3 Row4 [59]	1_x,2_x	1???
DES3 Row4^{-1} [59]	E_x	?0??
DES3 Row4^{-1} [59]	1_x	???1
DES3 Row4^{-1} [59]	6_x,8_x	?1??
DES4 Row1^{-1} [59]	8_x	??1?
DES4 Row1^{-1} [59]	2_x	1???
DES4 Row2^{-1} [59]	4_x	??1?
DES4 Row2^{-1} [59]	1_x	1???

Continued on next page

Table 2.7 – continued from previous page

S-box	Input	Output
DES4 Row3^{-1} [59]	2_x	??1?
DES4 Row3^{-1} [59]	8_x	1???
DES4 Row4^{-1} [59]	1_x	??1?
DES4 Row4^{-1} [59]	4_x	1???
DES5 Row1 [59]	5_x	?0??
DES5 Row1 [59]	1_x,4_x	?1??
DES5 Row1 [59]	C_x	1???
DES5 Row1^{-1} [59]	2_x	???1
DES5 Row1^{-1} [59]	3_x	??1?
DES5 Row2^{-1} [59]	4_x	1??1
DES5 Row3^{-1} [59]	6_x	??1?
DES5 Row3^{-1} [59]	4_x	1???
DES5 Row4 [59]	2_x	??1?
DES5 Row4 [59]	F_x	1???
DES5 Row4^{-1} [59]	8_x	1???
DES6 Row1^{-1} [59]	1_x	???1
DES6 Row2^{-1} [59]	8_x	???1
DES6 Row2^{-1} [59]	2_x	??1?
DES6 Row2^{-1} [59]	4_x	1???
DES6 Row3 [59]	2_x	??1?
DES6 Row3 [59]	6_x	?1??
DES6 Row3^{-1} [59]	1_x	???1
DES6 Row3^{-1} [59]	8_x	1???
DES6 Row4 [59]	E_x	???0
DES6 Row4 [59]	6_x,8_x	???1
DES6 Row4 [59]	1_x	?1??
DES6 Row4^{-1} [59]	6_x	0???
DES6 Row4^{-1} [59]	2_x,4_x	1???
DES7 Row1 [59]	7_x	?0??
DES7 Row1 [59]	4_x	??1?
DES7 Row1 [59]	1_x,6_x	?1??
DES7 Row1^{-1} [59]	D_x	??0?
DES7 Row1^{-1} [59]	8_x	???1
DES7 Row1^{-1} [59]	2_x,D_x	??1?
DES7 Row2 [59]	5_x	0???
DES7 Row2 [59]	9_x	??1?
DES7 Row2 [59]	1_x,4_x	1???
DES7 Row2^{-1} [59]	D_x	??0?
DES7 Row2^{-1} [59]	2_x	???1
DES7 Row2^{-1} [59]	5_x,8_x	??1?
DES7 Row3^{-1} [59]	4_x	???1
DES7 Row3^{-1} [59]	8_x	?1??
DES7 Row4 [59]	4_x	??11

Continued on next page

Table 2.7 – continued from previous page

S-box	Input	Output
DES7 Row4 [59]	1_x	?1??
DES7 Row4 [59]	6_x	1???
DES7 Row4^{-1} [59]	C_x	0???
DES7 Row4^{-1} [59]	4_x,8_x	1???
DES8 Row1 [59]	E_x	??0?
DES8 Row1 [59]	4_x	???1
DES8 Row1 [59]	6_x,8_x	??1?
DES8 Row1^{-1} [59]	F_x	??0?
DES8 Row1^{-1} [59]	9_x	0???
DES8 Row1^{-1} [59]	1_x	1?1?
DES8 Row1^{-1} [59]	E_x	??1?
DES8 Row1^{-1} [59]	8_x	1???
DES8 Row2 [59]	4_x	1???
DES8 Row2^{-1} [59]	F_x	?10?
DES8 Row2^{-1} [59]	2_x	???1
DES8 Row2^{-1} [59]	7_x,8_x	??1?
DES8 Row2^{-1} [59]	1_x	1???
DES8 Row3 [59]	2_x	??1?
DES8 Row3 [59]	1_x	?1??
DES8 Row3^{-1} [59]	5_x	0???
DES8 Row3^{-1} [59]	8_x	?1??
DES8 Row3^{-1} [59]	1_x,4_x	1???
DES8 Row4 [59]	A_x	?0??
DES8 Row4 [59]	2_x,8_x	?1??
DES8 Row4^{-1} [59]	E_x	?0??
DES8 Row4^{-1} [59]	3_x	???1
DES8 Row4^{-1} [59]	4_x,A_x	?1??
DES8 Row4^{-1} [59]	1_x	1???
GOST S4 [82]	8_x	1???
GOST S4 [82]	9_x	??1?
GOST S4^{-1} [82]	1_x	??1?
GOST S5^{-1} [82]	D_x	??1?
GOST S6 [82]	2_x	??1?
GOST S7 [82]	9_x	???1
GOST S7^{-1} [82]	2_x,8_x	?1??
GOST S7^{-1} [82]	3_x	1???
GOST S7^{-1} [82]	A_x	?0??
GOST S8^{-1} [82]	5_x	?1??
HB1 S1^{-1} [36]	1_x,E_x	???1
HB1 S1^{-1} [36]	F_x	???0
HB2 S3 [37]	2_x,D_x	1???
HB2 S3 [37]	F_x	0???
HB2 S4^{-1} [37]	1_x,E_x	???1

Continued on next page

Table 2.7 – continued from previous page

S-box	Input	Output
HB2 $S4^{-1}$ [37]	F_x	???0
LUCIFER $S0^{-1}$ [70]	3_x	???1
$LUFFA^{-1}$ [24]	1_x,3_x	???1
$LUFFA^{-1}$ [24]	2_x	???0
NOEKEON [31]	1_x	11??
NOEKEON [31]	8_x	0???
NOEKEON [31]	9_x	1???
NOEKEON [31]	A_x	?1??
NOEKEON [31]	B_x	?0??
$NOEKEON^{-1}$ [31]	1_x	11??
$NOEKEON^{-1}$ [31]	8_x	0???
$NOEKEON^{-1}$ [31]	9_x	1???
$NOEKEON^{-1}$ [31]	A_x	?1??
$NOEKEON^{-1}$ [31]	B_x	?0??
LBLOCK S0,S8 [85]	1_x, 2_x	???1
LBLOCK S0,S8 [85]	3_x	??10
LBLOCK S0,S8 [85]	8_x	??1?
LBLOCK S0,S8 [85]	B_x	??0?
LBLOCK $S0^{-1}$,$S8^{-1}$ [85]	1_x	?0??
LBLOCK $S0^{-1}$,$S8^{-1}$ [85]	4_x	01??
LBLOCK $S0^{-1}$,$S8^{-1}$ [85]	5_x	?1??
LBLOCK $S0^{-1}$,$S8^{-1}$ [85]	8_x, C_x	1???
LBLOCK S1,S9 [85]	1_x, 2_x	??1?
LBLOCK S1,S9 [85]	3_x	??01
LBLOCK S1,S9 [85]	8_x	???1
LBLOCK S1,S9 [85]	B_x	???0
LBLOCK $S1^{-1}$,$S9^{-1}$ [85]	2_x	?0??
LBLOCK $S1^{-1}$,$S9^{-1}$ [85]	4_x	01??
LBLOCK $S1^{-1}$,$S9^{-1}$ [85]	6_x	?1??
LBLOCK $S1^{-1}$,$S9^{-1}$ [85]	8_x, C_x	1???
LBLOCK S2 [85]	1_x, 2_x	??1?
LBLOCK S2 [85]	3_x	1?0?
LBLOCK S2 [85]	8_x	1???
LBLOCK S2 [85]	B_x	0???
LBLOCK $S2^{-1}$ [85]	1_x	01??
LBLOCK $S2^{-1}$ [85]	2_x	?0??
LBLOCK $S2^{-1}$ [85]	3_x	?1??
LBLOCK $S2^{-1}$ [85]	4_x, 5_x	1???
LBLOCK S3 [85]	1_x, 2_x	???1
LBLOCK S3 [85]	3_x	?1?0
LBLOCK S3 [85]	8_x	?1??
LBLOCK S3 [85]	B_x	?0??
LBLOCK $S3^{-1}$ [85]	1_x	?0??

Continued on next page

Table 2.7 – continued from previous page

S-box	Input	Output
LBLOCK S3^{-1} [85]	2_x, A_x	1???
LBLOCK S3^{-1} [85]	8_x	01??
LBLOCK S3^{-1} [85]	9_x	?1??
LBLOCK S4 [85]	1_x, 2_x	???1
LBLOCK S4 [85]	3_x	1??0
LBLOCK S4 [85]	8_x	1???
LBLOCK S4 [85]	B_x	0???
LBLOCK S4^{-1} [85]	1_x	?0??
LBLOCK S4^{-1} [85]	2_x	01??
LBLOCK S4^{-1} [85]	3_x	?1??
LBLOCK S4^{-1} [85]	4_x, 6_x	1???
LBLOCK S5 [85]	1_x, 2_x	???1
LBLOCK S5 [85]	3_x	?1?0
LBLOCK S5 [85]	8_x	?1??
LBLOCK S5 [85]	B_x	?0??
LBLOCK S5^{-1} [85]	1_x	?0??
LBLOCK S5^{-1} [85]	2_x	01??
LBLOCK S5^{-1} [85]	3_x	?1??
LBLOCK S5^{-1} [85]	8_x, C_x	1???
LBLOCK S6 [85]	1_x, 2_x	??1?
LBLOCK S6 [85]	3_x	??01
LBLOCK S6 [85]	8_x	???1
LBLOCK S6 [85]	B_x	???0
LBLOCK S6^{-1} [85]	2_x	?0??
LBLOCK S6^{-1} [85]	4_x	01??
LBLOCK S6^{-1} [85]	6_x	?1??
LBLOCK S6^{-1} [85]	8_x, C_x	1???
LBLOCK S7 [85]	1_x, 2_x	??1?
LBLOCK S7 [85]	3_x	??01
LBLOCK S7 [85]	8_x	???1
LBLOCK S7 [85]	B_x	???0
LBLOCK S7^{-1} [85]	2_x	?0??
LBLOCK S7^{-1} [85]	4_x	01??
LBLOCK S7^{-1} [85]	6_x	?1??
LBLOCK S7^{-1} [85]	8_x, C_x	1???
Piccolo [67]	1_x	10??
Piccolo [67]	2_x	0???
Piccolo [67]	3_x	1???
Piccolo [67]	8_x, 9_x	?1??
Piccolo^{-1} [67]	1_x,3_x	??1?
Piccolo^{-1} [67]	2_x	?10?
Piccolo^{-1} [67]	5_x	?0??
Piccolo^{-1} [67]	7_x	?1??

Continued on next page

Table 2.7 – continued from previous page

S-box	Input	Output
PRESENT [23]	9_x	???0
PRESENT [23]	1_x, 8_x	???1
PRESENT^{-1} [23]	5_x	???0
PRESENT^{-1} [23]	1_x, 4_x	???1
RECTANGLE [87]	2_x	?11?
RECTANGLE [87]	4_x	?1??
RECTANGLE [87]	6_x	?0??
RECTANGLE [87]	C_x	??1?
RECTANGLE [87]	E_x	??0?
RECTANGLE^{-1} [87]	1_x	??1?
RECTANGLE^{-1} [87]	4_x	??11
RECTANGLE^{-1} [87]	5_x	??0?
RECTANGLE^{-1} [87]	8_x	???1
RECTANGLE^{-1} [87]	C_x	???0
SERPENT S0 [3]	2_x, 4_x	1???
SERPENT S0 [3]	6_x	0???
SERPENT S0^{-1} [3]	4_x, 8_x	?1??
SERPENT S0^{-1} [3]	C_x	?0??
SERPENT S1 [3]	4_x, 8_x	?1??
SERPENT S1 [3]	C_x	?0??
SERPENT S1^{-1} [3]	1_x, 4_x	1???
SERPENT S1^{-1} [3]	5_x	0???
SERPENT S2 [3]	2_x, 8_x	???1
SERPENT S2 [3]	A_x	???0
SERPENT S2^{-1} [3]	1_x, C_x	???1
SERPENT S2^{-1} [3]	D_x	???0
SERPENT S4, S5 [3]	4_x, B_x	???1
SERPENT S4, S5 [3]	F_x	???0
SERPENT S6 [3]	2_x, 4_x	??1?
SERPENT S6 [3]	6_x	??0?
SERPENT S6^{-1} [3]	2_x, 8_x	??1?
SERPENT S6^{-1} [3]	A_x	??0?
SPONGENT [22]	1_x	???1
SPONGENT [22]	8_x	???1
SPONGENT [22]	9_x	???0
SPONGENT^{-1} [22]	7_x	1???
SPONGENT^{-1} [22]	8_x	1???
SPONGENT^{-1} [22]	F_x	0???
Twofish q0 t0 [65]	4_x	?1??
Twofish q0 t3^{-1} [65]	1_x,5_x	?1??
Twofish q0 t3^{-1} [65]	4_x	?0??
Twofish q1 t0 [65]	4_x	???1
Twofish q1 t1^{-1} [65]	2_x	?1??

Continued on next page

Table 2.7 – continued from previous page

S-box	Input	Output
Twofish q1 t3 [65]	7_x	?1??

Table 2.6: A 5.5-Round Impossible Differential for SERPENT

Input	X_0: 0000	0000	0000	0000	0000	0000	0000	0000
	X_1: 0001	0000	0000	0000	0000	0000	0000	0000
	X_2: 0000	0010	0000	0000	0000	0000	0000	0000
	X_3: 0001	0000	0000	0000	0000	0000	0000	0000
LT	X_0: 0000	0000	0000	0000	0000	0000	0000	0000
	X_1: 0000	0000	0000	0000	0000	0000	0000	0000
	X_2: 0000	0000	0000	0100	0000	0000	0000	0000
	X_3: 0000	0000	0000	0000	0000	0000	0000	0000
S_1	X_0: 0000	0000	0000	0?00	0000	0000	0000	0000
	X_1: 0000	0000	0000	0?00	0000	0000	0000	0000
	X_2: 0000	0000	0000	0100	0000	0000	0000	0000
	X_3: 0000	0000	0000	0?00	0000	0000	0000	0000
LT	X_0: 0?00	100?	0000	0000	0000	0000	00??	0010
	X_1: 0000	0000	0100	?000	0000	0000	0000	000?
	X_2: 00?0	0000	0000	110?	?000	1000	0000	0000
	X_3: 0001	00?0	0000	0000	0000	0000	0000	0000
S_2	X_0: 0??1	?0??	0100	??0?	?000	?000	00??	00??
	X_1: 0???	?0??	0?00	??0?	?000	?000	00??	00??
	X_2: 0???	?0??	0?00	??0?	?000	?000	00??	00??
	X_3: 0???	?0??	0?00	??0?	?000	?000	00??	00??
LT	X_0: ????	????	????	????	????	????	????	????
	X_1: ????	0??0	??0?	????	??0?	?1??	????	0???
	X_2: ????	????	????	????	????	????	1???	????
	X_3: ?0??	????	????	1?0?	??**1**?	??0?	????	??0?
				IMPOSSIBLE				
LT	X_0: ????	????	????	????	??0?	????	????	????
	X_1: ????	????	????	????	??0?	????	????	????
	X_2: ????	????	????	????	??0?	????	????	????
	X_3: ????	????	????	????	??**0**?	????	????	????
S_3	X_0: ?000	??0?	0?00	?0??	??0?	???0	?0??	0000
	X_1: 0???	????	??0?	????	?00?	?00?	????	?0??
	X_2: 0?00	????	0???	?00?	??0?	00?0	00??	??0?
	X_3: ????	????	????	????	?00?	??0?	???0	?0?0
LT	X_0: 0???	?0?0	?000	00?0	000?	000?	00?0	0000
	X_1: 0???	?0?0	?000	00?0	000?	000?	00?0	0000
	X_2: 0???	?0?0	?000	00?0	000?	000?	00?0	0000
	X_3: 0???	?0?0	?000	00?0	000?	000?	00?0	0000
S_4	X_0: 0000	0000	0000	00?0	000?	0000	0000	0000
	X_1: 0??0	00?0	0000	0000	0000	000?	0000	0000
	X_2: 0000	?000	0000	0000	0000	0000	00?0	0000
	X_3: 0?0?	0000	?000	0000	0000	000?	0000	0000
LT	X_0: 0?00	0000	0000	0000	0000	0000	0000	0000
	X_1: 0?00	0000	0000	0000	0000	0000	0000	0000
	X_2: 0?00	0000	0000	0000	0000	0000	0000	0000
	X_3: 0?00	0000	0000	0000	0000	0000	0000	0000
S_5	X_0: 0000	0000	0000	0000	0000	0000	0000	0000
	X_1: 0100	0000	0000	0000	0000	0000	0000	0000
	X_2: 0000	0000	0000	0000	0000	0000	0000	0000
	X_3: 0000	0000	0000	0000	0000	0000	0000	0000
LT	X_0: 0000	0000	0000	0000	0000	0000	0001	0000
	X_1: 1000	0000	0000	0000	0000	0000	0000	0000
	X_2: 0000	0000	0000	0000	0000	0000	0000	0000
	X_3: 0000	0000	0000	0000	0000	0000	0000	0000

CHAPTER 3

DIFFERENTIAL FACTORS

We introduce *differential factors* in this chapter. Differential factors are key differences for ciphers with key XOR before the S-box where the output difference of the S-box is invariant. An attacker cannot capture the whole key when there is a differential factor but this also reduces the work of the attacker.

3.1 Differential Factors

A differential variant attack on an SPN cipher tries to capture the round keys corresponding to the S-boxes activated by the differential. However, output difference of the the S-box operation may be invariant when the round key is XORed with some specific value. Such a case would prevent the attacker from fully capturing the round key. This observation is similar to the *linear factors* of block ciphers but here we are focusing on the S-box instead of some rounds of the cipher and we focus on key differences instead of invariant key bits.

Definition 3.1 ([25])**.** A block cipher is said to have a *linear factor* if, for all plaintexts and keys, there is a fixed non-empty set of key bits whose simultaneous complementation leaves the XOR sum of a fixed non-empty set of ciphertext bits unchanged.

In order to have a similar property for S-boxes in the concept of differential cryptanalysis, we define the *differential factors* as follows:

Definition 3.2. Let S be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. For all $x, y \in \mathbb{F}_2^n$ that satisfy $S(x) \oplus S(y) = \mu$, if we also have $S(x \oplus \lambda) \oplus S(y \oplus \lambda) = \mu$, then we say that the S-box has a *differential factor* λ for the output difference μ. (i.e. μ remains invariant for λ).

When we introduced undisturbed bits in Chapter 2, we considered the undisturbed bits of the S-boxes together with the undisturbed bits of the inverse of the S-boxes because in SPNs, the inverse of the S-box is used for decryption. In the following theorem, we prove that the number of differential factors of an S-box is the same with the number of differential factors of its inverse. Moreover, it also provides the differential factors of the inverse S-box when we know the differential factors of the S-box. Hence, there is no need to check the differential factors of the inverse S-boxes.

Theorem 3.1. *If a bijective S-box S has a differential factor λ for an output difference μ, then S^{-1} has a differential factor μ for the output difference λ.*

Proof. Let us assume that S has a differential factor λ for an output difference μ. If $S^{-1}(c_1) \oplus S^{-1}(c_2) = \lambda$ for some c_1 and c_2, then we need to show that $S^{-1}(c_1 \oplus \mu) \oplus S^{-1}(c_2 \oplus \mu) = \lambda$.

Let $c_1 \oplus \mu = S(p_1)$ for some p_1, then we have $S(S^{-1}(c_1) \oplus \lambda) \oplus S(p_1 \oplus \lambda) = \mu$ since λ is a differential factor of S for μ. Thus, we have

$$\begin{aligned} S^{-1}(c_1 \oplus \mu) \oplus S^{-1}(c_2 \oplus \mu) &= S^{-1}(S(p_1)) \oplus S^{-1}(S(S^{-1}(c_1) \oplus \lambda) \oplus \mu) \\ &= p_1 \oplus S^{-1}(S(p_1 \oplus \lambda)) \\ &= p_1 \oplus p_1 \oplus \lambda \\ &= \lambda \end{aligned}$$

□

Theorem 3.2. *If λ_1 and λ_2 are differential factors for an output difference μ, then $\lambda_1 \oplus \lambda_2$ is also differential factor for the output difference μ. i.e. All differential factors λ_i for μ forms a vector space.*

Proof. We are going to use the following change of variables: $x' = x \oplus \lambda_1$ and $y' = y \oplus \lambda_1$. For all (x, y) pairs satisfying $S(x) \oplus S(y) = \mu$, we have $S(x \oplus \lambda_1) \oplus S(y \oplus \lambda_1) = \mu$ and $S(x \oplus \lambda_2) \oplus S(y \oplus \lambda_2) = \mu$. Thus, we have

$$S(x \oplus \lambda_1 \oplus \lambda_2) \oplus S(y \oplus \lambda_1 \oplus \lambda_2) = S(x' \oplus \lambda_2) \oplus S(y' \oplus \lambda_2) = \mu$$

□

3.1.1 Differential Factors and Cryptanalysis

We start by recalling the definition of advantage.

Definition 3.3 ([66])**.** If an attack on an m-bit key gets the correct value ranked among the top r out of 2^m possible candidates, we say the attack obtained an $(m - log(r))$-bit *advantage* over exhaustive search.

Theorem 3.3. *In a block cipher let an S-box S contains a differential factor λ for an output difference μ and the partial round key k is XORed with the input of S. If an input pair provides the output difference μ under a partial subkey k, then the same output difference is observed under the partial subkey $k \oplus \lambda$. Therefore, during a differential attack involving the guess of a partial subkey corresponding to the output difference μ, the advantage of the cryptanalyst is reduced by 1 bit and the time complexity of this key guess step is halved.*

Proof. In a differential attack for any key k, k and $k \oplus \lambda$ would get the same number of hits since λ is a differential factor. Hence the attacker cannot distinguish half of the guessed keys with the other half. Therefore during the key guessing step, the attacker does not need to guess half of the keys. Thus, the time complexity of this step is halved. □

Corollary 3.4. *During a differential attack involving the guess of a partial subkey corresponding to the output difference μ of an S-box that has a vector space of differential factors of dimension r for μ, then advantage of the cryptanalyst is reduced by r bits and the time complexity of the key guess step is reduced by a factor of 2^r.*

Proof. Follows directly from Theorem 3.2 and Theorem 3.3. □

Table 3.1: Differential Factors of SERPENT's S-boxes

S-box	Differential Factor	Output Difference
S_0	4_x	4_x
S_0	D_x	F_x
S_1	4_x	4_x
S_1	F_x	E_x
S_2	2_x	1_x
S_2	4_x	D_x
S_6	6_x	2_x
S_6	F_x	F_x

Biham *et al.*'s differential-linear attacks [11, 35] on SERPENT are good examples for the importance of differential factors. Because since differential factors were not publicly known before 2014, in these attacks the differential factors are overlooked. Thus, the time complexity and the number of captured subkey bits of these attacks are actually less than it is reported in [11, 35]. The differential-linear attacks of [11, 35] start at round 1 and the 3-round differential activates 5 S-boxes S_1 in this round. Two of the output differences of these activated S-boxes is 4_x and E_x and there are differential factors for these differences as shown in Table 3.1. The authors guess every possible 20 subkey bits corresponding to these S-boxes but by Theorem 3.3, the attacker can have only 18-bit advantage and there is no need to try half of the subkey bits corresponding to these two differential factors. Thus, the advantage of the differential-linear attacks on 10, 11, and 12 rounds of SERPENT are actually 38, 46, and 158 bits instead of 40, 48, and 160 bits, respectively. And again by Theorem 3.3, the same attacks can be performed with time complexities reduced by a factor of 4.

Moreover, the 12-round attack of [35] adds one more round to the top of the differential which affects every S-box at round 0 except the S-boxes 2, 3, 19, and 23 and guesses the 112 bits of the subkey corresponding to these affected S-boxes. However, by using the undisturbed bits of SERPENT, we observed that the output difference of the S-box 9 is exactly 4_x. Since 4_x has also a differential factor for S_0, the attacker's advantage

Table 3.2: Differential Factors of 8×8 S-boxes

S-box	λ	μ
CRYPTON S0, S1 [48]	10_x	10_x
CRYPTON S0, S1 [48]	20_x	20_x
CRYPTON S0, S1 [48]	30_x	30_x
CRYPTON S0, S1 [48]	40_x	40_x
CRYPTON S0, S1 [48]	50_x	50_x
CRYPTON S0, S1 [48]	60_x	60_x
CRYPTON S0, S1 [48]	70_x	70_x
CRYPTON S0, S1 [48]	80_x	80_x
CRYPTON S0, S1 [48]	90_x	90_x
CRYPTON S0, S1 [48]	$A0_x$	$A0_x$
CRYPTON S0, S1 [48]	$B0_x$	$B0_x$
CRYPTON S0, S1 [48]	$C0_x$	$C0_x$
CRYPTON S0, S1 [48]	$D0_x$	$D0_x$
CRYPTON S0, S1 [48]	$E0_x$	$E0_x$
CRYPTON S0, S1 [48]	$F0_x$	$F0_x$

reduces to 158 bits and the time complexity of the attack further reduced by a factor of 2.

We are going to provide these observations on SERPENT's differential factors in more detail in Chapter 7 where we improve the attacks of [11, 35].

The only 8×8 S-boxes we could find with differential factors are the two S-boxes of the initial version of the CRYPTON cipher [48]. They contain 15 differential factors each and they are provided in Table 3.2. These S-boxes are replaced in the revised version of the CRYPTON cipher [49] and the new S-boxes do not contain any differential factors.

Table 3.3: Differential Factors of 4×4 S-boxes

S-box	λ	μ
DES1 Row3 [59]	F_x	2_x
DES1 Row3 [59]	F_x	8_x
DES1 Row3 [59]	F_x	A_x
DES2 Row1 [59]	6_x	A_x
DES2 Row2 [59]	2_x	7_x
DES2 Row2 [59]	4_x	7_x
DES2 Row2 [59]	6_x	7_x
DES2 Row3 [59]	1_x	A_x
DES2 Row3 [59]	6_x	A_x
DES2 Row3 [59]	7_x	A_x
DES3 Row3 [59]	2_x	6_x

Continued on next page

Table 3.3 – continued from previous page

S-box	λ	μ
DES3 Row3 [59]	8_x	6_x
DES3 Row3 [59]	A_x	6_x
DES3 Row4 [59]	3_x	1_x
DES3 Row4 [59]	3_x	6_x
DES3 Row4 [59]	3_x	7_x
DES3 Row4 [59]	3_x	8_x
DES3 Row4 [59]	3_x	9_x
DES3 Row4 [59]	1_x	E_x
DES3 Row4 [59]	2_x	E_x
DES3 Row4 [59]	3_x	E_x
DES3 Row4 [59]	3_x	F_x
DES5 Row4 [59]	2_x	F_x
DES6 Row1 [59]	9_x	D_x
DES6 Row2 [59]	B_x	4_x
DES6 Row4 [59]	6_x	6_x
DES7 Row2 [59]	4_x	D_x
DES7 Row2 [59]	9_x	D_x
DES7 Row2 [59]	D_x	D_x
DES7 Row4 [59]	4_x	3_x
DES7 Row4 [59]	1_x	C_x
DES7 Row4 [59]	4_x	C_x
DES7 Row4 [59]	5_x	C_x
DES7 Row4 [59]	4_x	F_x
DES8 Row2 [59]	6_x	7_x
DES8 Row2 [59]	B_x	8_x
GOST S1 [82]	5_x	3_x
GOST S4 [82]	D_x	5_x
GOST S6 [82]	9_x	B_x
GOST S8 [82]	7_x	5_x
GOST S8 [82]	E_x	6_x
LBLOCK S0, S8 [85]	B_x	1_x
LBLOCK S0, S8 [85]	3_x	4_x
LBLOCK S1, S6, S7, S9 [85]	B_x	2_x
LBLOCK S1, S6, S7, S9 [85]	3_x	4_x
LBLOCK S2 [85]	3_x	1_x
LBLOCK S2 [85]	B_x	2_x
LBLOCK S3 [85]	B_x	1_x
LBLOCK S3 [85]	3_x	8_x
LBLOCK S4, S5 [85]	B_x	1_x
LBLOCK S4, S5 [85]	3_x	2_x
LUFFA [24]	4_x	1_x
LUFFA [24]	2_x	2_x
NOEKEON [31]	1_x	1_x

Continued on next page

Table 3.3 – continued from previous page

S-box	λ	μ
NOEKEON [31]	B_x	B_x
PRESENT [23]	B_x	8_x
Piccolo [67]	1_x	2_x
Piccolo [67]	2_x	5_x
RECTANGLE [87]	2_x	4_x
RECTANGLE [87]	E_x	C_x
SARMAL S2 [83]	F_x	4_x
SARMAL S2 [83]	A_x	9_x
SERPENT S0 [3]	4_x	4_x
SERPENT S0 [3]	D_x	F_x
SERPENT S1 [3]	4_x	4_x
SERPENT S1 [3]	F_x	E_x
SERPENT S2 [3]	2_x	1_x
SERPENT S2 [3]	4_x	D_x
SERPENT S6 [3]	6_x	2_x
SERPENT S6 [3]	F_x	F_x
SPONGENT [22]	F_x	9_x
SPONGENT [22]	1_x	F_x
Twofish q0 t1 [65]	6_x	9_x
Twofish q1 t2 [65]	5_x	B_x

3.1.2 Relating Differential Factors to Other Properties of S-boxes

Since we are considering non-zero μ and λ, a 3×3 S-box can contain at most $7\cdot7 = 49$ differential factors. In such a case, an S-box provides no security at all. In Chapter 2, we showed that every bijective 3×3 S-box contains an undisturbed bit (actually at least 6 undisturbed bits). However, this is not the case for differential factors. Among the $8! = 40320$ different bijective 3×3 S-boxes, we observed that 10752 of them do not contain any differential factor. Moreover, 18816 of them contain 9, 9408 of them contain 25, and 1344 of them contain 49 differential factors.

We further observed that the 3×3 S-boxes that do not have any differential factor also have 6 undisturbed bits, which is the smallest number of undisturbed bits a 3×3 S-box can have. Thus, for the case of 3×3 S-boxes, it is enough to check differential factors. Thus, in general, it looks like it is harder to get differential factors than undisturbed bits.

In our literature search we found 102 unique 4×4 S-boxes that are used in block ciphers and hash functions and observed that 40 of them have 74 differential factors in total, without counting the differential factors of their inverses. These are the S-boxes of DES [59], GOST [82], LBLOCK [85], LUFFA [24], NOEKEON [31], Piccolo [67], PRESENT [23], RECTANGLE [87], SARMAL [83], SERPENT [3], SPONGENT [22] and Twofish [65] and they are provided in Table 3.3. During this analysis, we observed that the existence of differential factors for an S-box is closely related to the number

of nonzero entries in the columns of the DDT table. For instance, for a 4×4 S-box the maximum value in the DDT table, which is called differential uniformity, cannot be made less than 4 and since we prefer these kind of S-boxes for cryptographic purposes, we observed the following phenomenon:

Conjecture 3.1. For a 4×4 S-box S with a differential uniformity of 4, S has a differential factor for the output difference μ if and only if the μ-th column of the DDT table of S consists of only zeros and fours.

3.2 Equivalent Definitions with only One Variable

When defining differential factors in Section 3.1, we used two variables x and y since they are directly linked to the input pairs in differential cryptanalysis. One can observe that the same definition and theorems of Section 3.1 for bijective S-boxes can be given by using a single variable. We provide them as follows.

Definition 3.4. S has a *differential factor* λ for the output difference μ if

$$S^{-1}(S(x) \oplus \mu) \oplus \lambda = S^{-1}(S(x \oplus \lambda) \oplus \mu)$$

for all x.

Proposition 3.5. *Definition 3.2 is equivalent to Definition 3.4.*

Proof. Since $S(x) \oplus S(y) = \mu$, we have $y = S^{-1}(S(x) \oplus \mu)$. Similarly, $y \oplus \lambda = S^{-1}(S(x \oplus \lambda) \oplus \mu)$ since $S(x \oplus \lambda) \oplus S(y \oplus \lambda) = \mu$. XORing both equations gives $\lambda = S^{-1}(S(x) \oplus \mu) \oplus S^{-1}(S(x \oplus \lambda) \oplus \mu)$ and we are done. □

Definition 3.5. S has a *differential factor* λ for the output difference μ if

$$S(S^{-1}(x) \oplus \lambda) \oplus \mu = S(S^{-1}(x \oplus \mu) \oplus \lambda)$$

for all x.

Proposition 3.6. *Definition 3.2 is equivalent to Definition 3.5.*

Proof. Let $y = S(x)$. Then the Definition 3.4 becomes

$$S^{-1}(y \oplus \mu) \oplus \lambda = S^{-1}(S(S^{-1}(y) \oplus \lambda) \oplus \mu)$$

for all y. Applying the S operation on both sides of the equation gives

$$S(S^{-1}(y \oplus \mu) \oplus \lambda) = S(S^{-1}(y) \oplus \lambda) \oplus \mu$$

for all y and we are done. □

Thus, Propositions 3.5 and 3.6 prove the Theorem 3.1.

Proposition 3.7. *If λ_1 and λ_2 are differential factors for an output difference μ, then $\lambda_1 \oplus \lambda_2$ is also differential factor for the output difference μ. i.e. All differential factors λ_i for μ forms a vector space.*

Proof. We have

$$S^{-1}(S(x) \oplus \mu) \oplus \lambda_1 = S^{-1}(S(x \oplus \lambda_1) \oplus \mu)$$

for all x, by Definition 3.4. And we have

$$S^{-1}(S(x \oplus \lambda_1) \oplus \mu) \oplus \lambda_2 = S^{-1}(S(x \oplus \lambda_1 + \lambda_2) \oplus \mu)$$

since λ_2 is a differential factor. Thus, we get

$$S^{-1}(S(x) \oplus \mu) \oplus \lambda_2 \oplus \lambda_2 = S^{-1}(S(x \oplus \lambda_1 \oplus \lambda_2) \oplus \mu)$$

for all x and we are done. □

CHAPTER 4

IMPROBABLE DIFFERENTIAL CRYPTANALYSIS

We introduce *improbable differential cryptanalysis* in this chapter. It is a differential cryptanalysis technique that bridges the gap between (truncated) differential cryptanalysis and impossible differential cryptanalysis. We first came up with the idea of improbable differential cryptanalysis in 2008 but we first published it in 2010 in [74]. We published parts of this chapter in [74, 76, 78].

4.1 Introduction

Improbable differential attack is a statistical differential attack in which a given differential of a cipher is less probable than a random permutation. Hence, we aim to find a differential with α input difference and β output difference so that these differences are observed with probability p_0 for the cipher and with probability p for a random permutation where $p_0 < p$. An improbable differential is defined as a differential that does not have the output difference β with a probability p', when the input difference is α. Thus, p' denotes the total probability of differentials having α input difference with an output difference other than β. Hence for the cipher, probability of observing the α and β differences (i.e. satisfying the improbable differential) becomes $p_0 = p \cdot (1 - p')$. Note that p_0 may be larger than $p \cdot (1 - p')$ if there are differentials having α input difference and β output difference. Hence the attacker should check the existence of such differentials.

Note that the impossible differential attacks can be seen as a special case of improbable differential attacks where the probability p' is taken as 1.

An improbable differential can be obtained by using a miss-in-the-middle [6] like technique which we call the almost miss in the middle technique. Let α difference becomes δ with probability p_1 after r_1 rounds of encryption and β difference becomes γ after r_2 rounds of decryption as shown in Fig. 4.1. With the assumption that these two events are independent, if δ is different than γ, then α difference does not become β with probability $p' = p_1 \cdot p_2$ after $r_1 + r_2$ rounds of encryption. Note that p_1 and p_2 equal to 1 in the miss-in-the-middle technique. Furthermore, we define an expansion method for constructing an improbable differential from an impossible differential in Section 4.2.

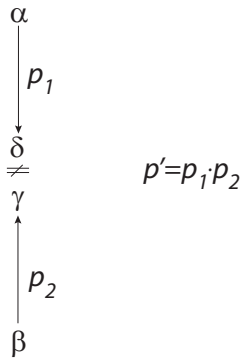

Figure 4.1: Almost miss in the middle technique

4.2 Expansion Technique: Improbable Differentials from Impossible Differentials

An improbable differential can be obtained by combining a differential (or two) with an impossible differential in order to obtain improbable differentials covering more rounds. Let $\delta \nrightarrow \gamma$ be an impossible differential and $\alpha \rightarrow \delta$ and $\gamma \leftarrow \beta$ be two differentials with probabilities p_1 and p_2, respectively. Then we can construct improbable differentials $\alpha \nrightarrow \gamma$, $\delta \nrightarrow \beta$ and $\alpha \nrightarrow \beta$ with probabilities p' equal to p_1, p_2 and $p_1 \cdot p_2$ as shown in Fig. 4.2.

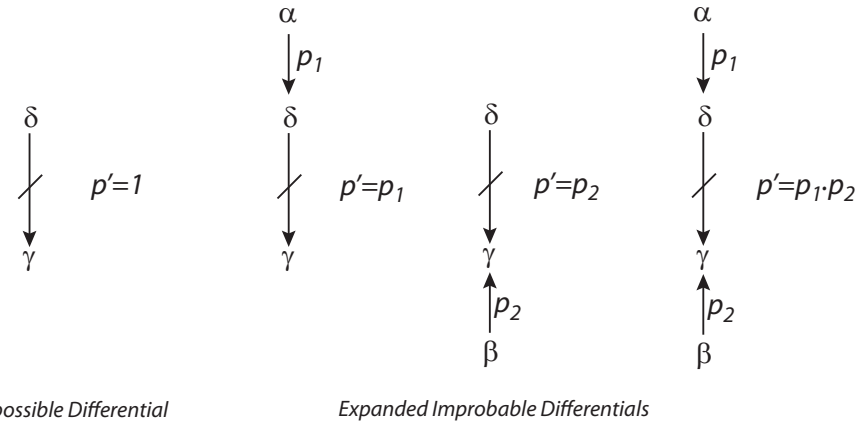

Figure 4.2: Expansion of an impossible differential to improbable differentials

This expansion method can be used to construct improbable differentials to distinguish more rounds of the cipher from a random permutation; or an impossible differential attack can be turned into an improbable differential attack on more rounds of the cipher when suitable differentials $\alpha \rightarrow \delta$ or $\gamma \leftarrow \beta$ exist. However, such a conversion might require more data to obtain the correct key and hence result in higher data and time complexity. If the size of the guessed key decreases in the converted improbable differential attack, so does the memory complexity. The guessed subkeys can be represented by one bit of an array in impossible differential attacks. However, we need to

keep counters for the subkeys in improbable differential attacks and hence the memory complexity is higher when the same number of subkeys are guessed.

4.3 On the Expansion Technique

In the proposal of the improbable differential cryptanalysis [74], we cautioned that the equation $p_0 = (1 - p')p$ for an improbable differential may not be accurate if there are high-probability truncated differentials with the same input and output differences, and that the actual probabilities should be checked. However, it may not always be possible to computationally perform such a verification. Recently, Blondeau [17] has questioned the validity of the previous improbable differential attacks on PRESENT and CLEFIA [77, 74].

For instance in [77], we provide two improbable differentials Δ_1 and Δ_2 for PRESENT where ? stands for any non-zero difference

$$\Delta_5 : 0000000000001001_x \rightarrow_{9r} 555?555?555?5551_x, p = 2^{-48}$$
$$\Delta_6 : 0000000000001001_x \rightarrow_{10r} ???0???0???0???1_x, p = 2^{-16}$$

Due to experimental results, we claimed that Δ_5 is not an improbable differential due to the existence of high-probability truncated differentials but Δ_6 is a correct improbable differential, and we performed the attack using Δ_6. In [17], by experimentally observing that the probability of the first 8 rounds of Δ_5 is $p_E = 2^{-12.97}$ while $p = 2^{-13}$, Blondeau shows that an improbable differential attack using Δ_5 cannot succeed and concludes that it may be impossible to obtain improbable differentials for SPN ciphers using the expansion technique due to the failure of Δ_5. However, in [78], Tezcan and Temizel show that the highly parallel structure of graphics processing units (GPUs) are suitable for the verification of differentials and, by performing $2^{44.5}$ reduced-round PRESENT encryption in less than 8 hours using a single Tesla k20 GPU, they show that the theoretically obtained p_0 for Δ_6 is actually correct. Although Blondeau claims in [17] that Δ_5 was used to attack PRESENT in [75], it does not appear there since [75] is only an abstract in a book of abstracts. Apparently Blondeau confuses an earlier version of the paper [77] with [75], which is available on the web.

Moreover, we observed that by removing the last rounds, Δ_6 is also valid when it is reduced to 9 or 8 rounds and Δ_5 is not valid even when it is reduced to 7 or 6 rounds. Since the validity of Δ_6 reduced to 8 rounds and Δ_5 reduced to 6 rounds can be checked using just a few hundred plaintext pairs, this way we can easily check the validity of the assumptions made when constructing improbable differentials. Thus, it can be seen that Δ_5 fails due to the existence of 6-round high-probability differentials for PRESENT. We provide the experimental results on Δ_6 in Table 4.1.

Note that this approach does not apply to the improbable differentials of CLEFIA reduced to 8 or 7 rounds since the 9-round impossible differentials

$$\Delta_1 : [0_{(32)}, 0_{(32)}, 0_{(32)}, [X, 0, 0, 0]_{(32)}] \nrightarrow_{9r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [0, Y, 0, 0]_{(32)}], p = 2^{-120}$$

and

$$\Delta_2 : [0_{(32)}, 0_{(32)}, 0_{(32)}, [0, 0, X, 0]_{(32)}] \nrightarrow_{9r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [0, Y, 0, 0]_{(32)}], p = 2^{-120}$$

are obtained using the properties of the M_0 matrix of the eighth round, and the differentials obtained by removing the last rounds of Δ_1 and Δ_2 are no longer impossible. This also favors the validity of the improbable differentials $\Delta_3|\Delta_1$ and $\Delta_4|\Delta_2$ since they can only fail if there exist 9-round high-probability differentials with the same input and output differences.

$$\Delta_3 : [[\psi, 0, 0, 0]_{(32)}, \zeta_{(32)}, 0_{(32)}, 0_{(32)}] \rightarrow_{1r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [\psi, 0, 0, 0]_{(32)}]$$
$$\Delta_4 : [[0, 0, \psi, 0]_{(32)}, \zeta'_{(32)}, 0_{(32)}, 0_{(32)}] \rightarrow_{1r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [0, 0, \psi, 0]_{(32)}]$$

Table 4.1: Comparison of the theoretically and experimentally calculated values of p_0 of Δ_6 over 8, 9, and 10 rounds. The results indicate that the experimental values agree with and in fact are better than the theoretically calculated values.

Length	p	p_0	p_E
8 rounds	$2^{-1.192645}$	$2^{-1.192651}$	$2^{-1.192661}$
9 rounds	$2^{-12.192645}$	$2^{-12.192651}$	$2^{-12.193691}$
10 rounds	2^{-13}	$2^{-13.000006}$	$2^{-13.001046}$

Verification of the 10-round improbable differentials used in the attacks on CLEFIA are beyond our computational power. For this reason Blondeau constructs a toy Feistel cipher with six data lines of 4-bit words and experiments on an 11-round improbable differential Δ_7 with $p = 2^{-20}$ and $p_0 = 2^{-20.42}$ [17].

$$\Delta_7 : [X, Y, 0, 0, 0, 0] \rightarrow_{11r} [0, 0, 0, Z, 0, 0], p = 2^{-20}$$

The experimental result[1] of $p_E = 2^{-20.28}$ being greater than p_0 is valid only for this toy cipher, and the reason for this difference is the small block size and the slow diffusion of the toy cipher. It can easily be verified that the theoretical probabilities of similar improbable differentials match the practical ones in similar toy ciphers with larger block size or better diffusion. For instance, by adding two more data lines to Blondeau's toy cipher (i.e., the same cipher with eight data lines of 4-bit words), we can experiment on a 15-round improbable differential Δ_8 with $p = 2^{-28}$ and $p_0 = 2^{-28.42}$ which is obtained in a similar fashion.

$$\Delta_8 : [X, Y, 0, 0, 0, 0, 0, 0] \rightarrow_{15r} [0, 0, 0, 0, 0, Z, 0, 0], p = 2^{-28}$$

This time, the experimental probability becomes $p_E = 2^{-28.52}$ which is even better than the theoretical one.

[1] Note that this differential is valid only when $S(a) \oplus S(a + X) = Y$, where X and Y differences are chosen according to the DDT of the S-box used. However, in [12] some of the experimental results were given for arbitrary X and Y regardless of such consistency with the DDT. Obviously such results cannot be used to judge the accuracy of the differential being studied.

4.3.1 Expansions with Two Differentials

There is no easy way to theoretically compute the probability of an improbable differential obtained by expanding an impossible differential with two differentials. We observed that if these differentials were considered as independent statistical events, then the probability would be $(1 - p_1 p_2) \cdot p$. Hence the probability of the improbable differential should be between $(1 - p_1 p_2) \cdot p$ and p. But the attack would fail if it is closer to p than expected. There are no improbable differential attacks in the literature that use expansions with two differentials and we were not able to construct one that can be checked with our computational power. However, the uniformity assumption that is made in the single differential expansion appears twice in this case and the third assumption is the independency of the events. For this reason we suggest that only the experimentally verified expansions with two differentials should be used in cryptanalysis.

4.3.2 On Constructing Expansions

An intuition for constructing improbable differentials using the expansion technique is, first obtain the longest impossible differential and then combine it with a suitable differential with high probability. The main parameters p and p' here come from the impossible differential and the expansion differential combined with it, respectively. Therefore, knowing how p and p' affect the attack complexity can help the cryptanalyst to obtain the optimal improbable differentials, instead of always beginning with the longest impossible differential available.

In [18, 20], aside from providing accurate estimates for data complexity and success probability for various statistical cryptanalysis techniques, Blondeau *et al.* also observed that the behaviour of the number of pairs required to perform an attack is dominated by $D(p_0||p)^{-1}$. Note that a similar observation was made by Daemen *et al.* for differential power analysis [30].

We first recall the Kullback-Leibler divergence which plays an important role in these estimates.

Definition 4.1 (Kullback-Leibler divergence [29])**.** Let P and Q be two Bernoulli probability distributions of parameters p and q. The Kullback - Leibler divergence between P and Q is defined by

$$D(p||q) = p \ \ln\left(\frac{p}{q}\right) + (1-p) \ \ln\left(\frac{1-p}{1-q}\right). \tag{4.1}$$

We can observe the following result on the Kullback-Leibler divergence for the improbable differentials:

Proposition 4.1. *An improbable differential attack with $p' \gg p_0$ has data complexity $O((p')^2 \cdot p)^{-1}$.*

Proof. We use the fact that $\ln(1-p) \approx -p$ for small p.

$$
\begin{array}{lcll}
D(p_0||p) & = & p_0 \ln(\frac{p_0}{p}) + (1-p_0)\ln(\frac{1-p_0}{1-p}) & \\
& = & p_0 \ln(1-p') + (1-p_0)\ln(\frac{1-p_0}{1-p}) & (\text{since } \frac{p_0}{p} = 1-p') \\
& \approx & p_0(-p') + (1-p_0)(p-p_0) & (\text{since } \ln(1-p) \approx -p) \\
& \approx & p_0(-\frac{p-p_0}{p}) + (1-p_0)(p-p_0) & (\text{since } p' = \frac{p-p_0}{p}) \\
& \approx & (p-p_0)(-\frac{p_0}{p} + 1 - p_0) & \\
& \approx & (p-p_0)(-1+p'+1-p_0) & (\text{since } \frac{p_0}{p} = 1-p') \\
& \approx & (p' \cdot p)(p'-p_0) & (\text{since } p - p_0 = p'p) \\
& \approx & (p')^2 p - p' \cdot p \cdot p_0 & \\
& \approx & (p')^2 \cdot p & (\text{since } p' \gg p_0)
\end{array}
$$

□

Thus, we conclude that an improbable differential attack that uses the expansion technique has data complexity $O((p')^2 \cdot p)^{-1}$. This results can be experimentally verified by running the Algorithm 4.1,which is going to be defined in the following Section, with different inputs.

4.4 Data Complexity and Success Probability

Since p_0 is less than p, our aim is to use N plaintext pairs and count the hits that every guessed subkey gets and expect that the counter for the correct subkey to be less than a threshold T. Number of hits a wrong subkey gets can be seen as a random variable of a binomial distribution with parameters N, p (and a random variable of a binomial distribution with parameters N, p_0 for the correct subkey). We denote the *non-detection* error probability with p_{nd} which is the probability of the counter for the correct subkey to be higher than T. And we denote the *false alarm* error probability with p_{fa} which is the probability of the counter for a wrong subkey to be less than or equal to T. Therefore, the success probability of an improbable differential attack is $1 - p_{nd}$.

Accurate estimates of the data complexity and success probability for many statistical attacks are provided by Blondeau *et al.* in [18, 20] and these estimates can be used for improbable differential attacks with some modifications. Unlike improbable differential cryptanalysis, in most of the statistical attacks $p_0 > p$ and this assumption is made throughout [18]. Hence, we need to modify the approximations N', N'' and N_∞ of the number of required samples N that are given in [18] for the $p_0 < p$ case in order to use them for improbable differential attacks.

We modify Algorithm 1 of [18] for the $p_0 < p$ case which computes the exact number of required samples N and corresponding relative threshold $\tau := \frac{T}{N}$ to reach error probabilities less than (p_{nd}, p_{fa}). The estimates for non-detection and false alarm error probabilities are denoted by $G_{nd}(N, \tau)$ and $G_{fa}(N, \tau)$.

Algorithm 4.1. (from [18], modified for the $p_0 < p$ Case)

Input : p_0, p, p_{nd}, p_{fa}
Output : N, τ
$\tau_{min} := p_0, \tau_{max} := p$
repeat
$\tau := \frac{\tau_{min}+\tau_{max}}{2}$
Compute N_{nd} such that $\forall N > N_{nd}, G_{nd}(N, \tau) \leq p_{nd}$
Compute N_{fa} such that $\forall N > N_{fa}, G_{fa}(N, \tau) \leq p_{fa}$
if $N_{nd} > N_{fa}$ **then** $\tau_{min} = \tau$
else $\tau_{max} = \tau$
until $N_{nd} = N_{fa}$
$N := N_{nd}$
Return N, τ

N_{nd} and N_{fa} can be calculated by a dichotomic search and the following Equations 4.2 and 4.3 can be used for the estimates $G_{nd}(N, \tau)$ and $G_{fa}(N, \tau)$, respectively. The number of samples obtained from the algorithm with these estimates is denoted by N_∞.

Theorem 4.2 ([2]). *Let p_0 and p be two real numbers such that $0 < p_0 < p < 1$ and let τ such that $p_0 < \tau < p$. Let Σ_0 and Σ_k follow a binomial law of respective parameters (N, p_0) and (N, p). Then as $N \to \infty$,*

$$P(\Sigma_0 \geq \tau N) \sim \frac{(1-p_0)\sqrt{\tau}}{(\tau - p_0)\sqrt{2\pi N(1-\tau)}} e^{-ND(\tau||p_0)}, \tag{4.2}$$

and

$$P(\Sigma_k \leq \tau N) \sim \frac{p\sqrt{1-\tau}}{(p-\tau)\sqrt{2\pi N\tau}} e^{-ND(\tau||p)}. \tag{4.3}$$

A simple approximation N' of N is defined in [18] when the relative threshold is chosen as $\tau = p_0$ which makes non-detection error probability p_{nd} of order $1/2$. We define N' for the $p_0 < p$ case as in [16]:

Proposition 4.3. *For a relative threshold $\tau = p_0$, a good approximation of the required number of pairs N to distinguish between the correctly keyed permutation and an incorrectly keyed permutation with false alarm probability less than or equal to p_{fa} is*

$$N' = -\frac{1}{D(p_0||p)} \left[\ln \left(\frac{\nu \cdot p_{fa}}{\sqrt{D(p_0||p)}} \right) + 0.5 \ln(-\ln(\nu \cdot p_{fa})) \right] \tag{4.4}$$

where

$$\nu = \frac{(p-p_0)\sqrt{2\pi p_0}}{p\sqrt{(1-p_0)}}.$$

In [18] a good approximation of N' which is also valid for the $p_0 < p$ case is defined as follows

$$N'' = -\frac{\ln(2\sqrt{\pi} p_{fa})}{D(p_0||p)}. \tag{4.5}$$

CHAPTER 5

ATTACKS ON PRESENT

In this chapter, we show how undisturbed bits can be used to attack ciphers and present 12 and 13-round improbable differential attacks on PRESENT that exploit undisturbed bits of PRESENT's S-box. We published parts of this chapter in [77].

5.1 10-Round Improbable Differential

In Chapter 2 we obtained a 6-round impossible differential by using the undisturbed bits of PRESENT's S-box. Without using the undisturbed bits, the longest impossible differential we could find for PRESENT had a length of 5-rounds with difference of the 60 out of 64 bits are fixed which means $p = 2^{-60}$. However, by using the 6 undisturbed bits of PRESENT, we can construct 5 and 6-round impossible differentials with only 13 and 39 bits fixed, respectively. In this section, by further using the undisturbed bits, we construct a 5-round differential with probability $2^{-17.84}$ that can be combined with the 5-round impossible differential. Hence, by using the expansion method, we construct a 10-round improbable differential and provide improbable differential attacks on PRESENT reduced to 12 and 13 rounds.

We use $X_i = x_{15}, \ldots, x_0$ to denote the intermediate differences of the round i with x_0 being the least significant nibble and $0 \leq i \leq 31$. Moreover, we use P, S, I to the denote output of the permutation layer, output of the substitution layer, and input of a round, respectively.

Using the undisturbed bits of PRESENT that are provided in Table 2.7, we construct a 5-round impossible differential that is provided in Table 5.1.

Instead of combining our 6-round impossible differential with a 4-round differential, we combined our 5-round impossible differential with a 5-round differential. This is because, we have $p = 2^{-13}$ and $p = 2^{-39}$ for our 5 and 6-round impossible differentials, respectively and the smaller probability $p = 2^{-39}$ results in higher data and time complexities. That choice also prevents us from performing a 13-round attack on PRESENT. This change is due to the observation that an improbable differential attack that uses the expansion technique has data complexity $O((p')^2 \cdot p)^{-1}$, which is obtained in Section 4.3.2.

Table 5.1: A 5-Round Impossible Differential for PRESENT

	Differences in bits															
Rounds	x_{15}	x_{14}	x_{13}	x_{12}	x_{11}	x_{10}	x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
$X_{7,I}$	0000	0000	0000	0000	0000	0000	0000	0000	0000	1001	0000	0000	0000	1001	0000	0000
$X_{8,S}$	0000	0000	0000	0000	0000	0000	0000	0000	0000	???0	0000	0000	0000	???0	0000	0000
$X_{8,P}$	0000	0000	0?00	0?00	0000	0000	0?00	0?00	0000	0000	0?00	0?00	0000	0000	0000	0000
$X_{9,S}$	0000	0000	????	????	0000	0000	????	????	0000	0000	????	????	0000	0000	0000	0000
$X_{9,P}$	00??	00??	00??	0000	00??	00??	00??	0000	00??	00??	00??	0000	00??	00??	00??	0000
$X_{10,S}$	????	????	????	0000	????	????	????	0000	????	????	????	0000	????	????	????	0000
$X_{10,P}$	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0
$X_{10,P}$	????	????	????	????	????	????	????	????	????	????	????	????	$???x$	$???x$	$???x$	???1
$X_{11,S}$	????	????	????	????	????	????	????	????	????	????	????	????	$000x$	$000x$	$000x$	0001
$X_{11,P}$	????	????	????	0000	????	????	????	0000	????	????	????	0000	????	????	????	$xxx1$
$X_{12,S}$	????	????	????	0000	????	????	????	0000	????	????	????	0000	????	????	????	0001
$X_{12,P}$	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???0	???1

We extend the 5-round impossible differential given in Table 5.1 to a 10-round improbable differential by combining it with a 5-round characteristic which is given in Table 5.2.

Table 5.2: A 5-Round Differential Characteristic for PRESENT

Rounds	Differences	Probability
$X_{2,I}$	$x_2 = 1_x, x_0 = 1_x$	1
$X_{3,S}$	$x_2 = 9_x, x_0 = 9_x$	2^{-4}
$X_{3,P}$	$x_{12} = 5_x, x_0 = 5_x$	1
$X_{4,S}$	$x_{12} = 1_x, x_0 = 1_x$	2^{-6}
$X_{4,P}$	$x_3 = 1_x, x_0 = 1_x$	1
$X_{5,S}$	$x_3 = 9_x, x_0 = 9_x$	2^{-4}
$X_{5,P}$	$x_{12} = 9_x, x_0 = 9_x$	1
$X_{6,S}$	$x_{12} = 4_x, x_0 = 4_x$	2^{-4}
$X_{6,P}$	$x_{11} = 1_x, x_8 = 1_x$	1
$X_{7,S}$	$x_{11} = 3_x, x_8 = 3_x$	2^{-4}
$X_{7,P}$	$x_6 = 9_x, x_2 = 9_x$	1

The total probability of the 5-round characteristic is 2^{-22} and hence for our 10-round improbable differential we have $p' \geq 2^{-22}$. However, a closer look at the characteristic shows that this probability is much higher. Because in Table 5.1 we take input differences of x_6 and x_2 as 9_x at $X_{7,I}$ only to show that the impossible differential can be combined with the characteristic. If we take the input difference of i S-boxes to be 9_x where $1 \leq i \leq 16$, the impossible differential still holds. Thus, the first round of the impossible differential is satisfied if the S-boxes x_{12} and x_0 at $X_{6,S}$ have the same output differences and the affected S-boxes at $X_{7,S}$ have the same output differences. By using the difference distribution table of PRESENT's S-box provided in Table 5.3, one can see that the probability of this event is

$$2^{-6} + 2^{-7} + 2^{-9} + 2^{-10} = 0.025634765625.$$

Thus, the probability of our 10-round improbable differential becomes

$$p' \geq 0.025634765625 \cdot 2^{-14} \approx 2^{-19.29}.$$

Table 5.3: Difference Distribution Table of PRESENT's S-box

	0_x	1_x	2_x	3_x	4_x	5_x	6_x	7_x	8_x	9_x	A_x	B_x	C_x	D_x	E_x	F_x
0_x	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1_x	0	0	0	4	0	0	0	4	0	4	0	0	0	4	0	0
2_x	0	0	0	2	0	4	2	0	0	0	2	0	2	2	2	0
3_x	0	2	0	2	2	0	4	2	0	0	2	2	0	0	0	0
4_x	0	0	0	0	0	4	2	2	0	2	2	0	2	0	2	0
5_x	0	2	0	0	2	0	0	0	0	2	2	2	4	2	0	0
6_x	0	0	2	0	0	0	2	0	2	0	0	4	2	0	0	4
7_x	0	4	2	0	0	0	2	0	2	0	0	0	2	0	0	4
8_x	0	0	0	2	0	0	0	2	0	2	0	4	0	2	0	4
9_x	0	0	2	0	4	0	2	0	2	0	0	0	2	0	4	0
A_x	0	0	2	2	0	4	0	0	2	0	2	0	0	2	2	0
B_x	0	2	0	0	2	0	0	0	4	2	2	2	0	2	0	0
C_x	0	0	2	0	0	4	0	2	2	2	2	0	0	0	2	0
D_x	0	2	4	2	2	0	0	2	0	0	2	2	0	0	0	0
E_x	0	0	2	2	0	0	2	2	2	2	0	0	2	2	0	0
F_x	0	4	0	0	4	0	0	0	0	0	0	0	0	0	4	4

One way of obtaining the precise value of p' is to search every possible 10-round differential with the desired input and output differences. Instead, we experimentally calculated p' as $2^{-17.84}$ (for 100 randomly selected keys, we used 2^{32} pairs in each experiment and the mean value was $2^{-17.84}$ with a standard deviation of $2^{-22.40}$).

Note that the 5-round impossible differential and the 5-round differential are valid for any consecutive 5 rounds of PRESENT because the rounds are almost identical. However, the reason for starting the impossible differential at the 7th round and the differential at the 2nd round of PRESENT is that we wanted to attack the first 12 and 13 rounds of PRESENT. This is not the case for our improbable differential attack on SERPENT that is provided in Chapter 7.

5.2 Attack on PRESENT-80-12

We add two rounds above our 10-round improbable differential and attack 12 rounds of PRESENT-80. The 12-round improbable differential attack is summarized in Table 5.4.

The attack procedure is as follows:

1. Choose 2^n structures of 2^{16} plaintexts each where x_3, x_2, x_1, and x_0 take all possible values and other bits are fixed. Such a structure of plaintexts propose

Table 5.4: 12-Round Improbable Differential Attack

Rounds	Differences	Probability
$X_{0,I}$	x_3, x_2, x_1 =???? except $1_x, 6_x, 8_x$, and E_x, $x_0 = 2_x, 4_x, A_x$, or C_x	-
$X_{1,S}$	$x_3 = 0?0?, x_2 = 0?0?, x_1 = 0?0?, x_0 = 0101$	$3^{-3} \cdot 2^{-2}$
$X_{1,P}$	x_8 =???1, x_0 =???1	1
$X_{2,S}$	$x_8 = 1_x, x_0 = 1_x$	2^{-6}
$X_{2,P}$	$x_2 = 1_x, x_0 = 1_x$	1
$X_{3,S}$	$x_2 = 9_x, x_0 = 9_x$	2^{-4}
$\vdots$	$\vdots$	$\vdots$
$X_{12,P}$	???0 ???0 ???0 ???0 ???0 ???0 ???0 ???0 ???0 ???0 ???0 ???0 ???0 ???0 ???0 ???1	

$3^3 \cdot 2^{23}$ pairs where x_3, x_2, x_1 have any difference except 1_x, 6_x, 8_x, and E_x and x_0 has difference 2_x, 4_x, A_x, or C_x. Hence we gather $N = 3^3 \cdot 2^{n+23}$ many plaintext pairs.

2. Obtain all the ciphertexts and choose only the ciphertexts pairs where x_{12}, x_8, and x_4 have zero difference, x_0 has difference L or x_{12}, x_4, and x_0 have zero difference, x_8 has difference L just before the final permutation of the last round, where $L \in \{1_x, 2_x, 4_x, 8_x\}$. Hence, there is a 13-bit filtering condition over the ciphertext pairs and therefore, $3^3 \cdot 2^{n+10}$ pairs remain.

   Actually, taking L as non-zero is enough for the impossible differential to hold. However, we experimentally verified that for the input difference of the improbable differential there are differentials which lead to weight one differences at x_0 of $X_{11,P}$ with high probability. Hence, we chose L as a difference of weight one and used the fact that PRESENT's S-box does not map weight one differences to weight one differences.

3. Guess 16 bits of the key $k_{31}, k_{30}, \ldots, k_{16}$ and partially encrypt every plaintext pair. Keep only the pairs where $x_0 = 5_x$ and x_3, x_2, x_1 are of the form $0?0?$ after the substitution. This filtering condition has probability $3^{-3} \cdot 2^{-2}$ and 2^{n+8} pairs remain.

4. Keep only the pairs where x_8 and x_0 have the difference 3_x, 5_x, 7_x, B_x, D_x or F_x after the permutation. Hence $\frac{6}{8}\frac{6}{8} \cdot 2^{n+8}$ pairs remain.

5. Guess 8 bits of the key $k_{70}, k_{69}, k_{68}, k_{67}, k_{38}, k_{37}, k_{36}, k_{35}$ and partially encrypt every pair at $X_{1,P}$. Increase the counter of the corresponding key by 1 if a pair satisfies $x_8 = 1_x$ and $x_0 = 1_x$ at $X_{2,S}$. We expect $\frac{1}{36}\frac{6}{8}\frac{6}{8} \cdot 2^{n+8} = 2^{n+2}$ many pairs to satisfy this property for a guessed key.

6. For every recorded 24-bit subkey with counter less than the threshold T, remaining bits of the key can be obtained by repeating the attack by replacing the 5-round differential with a different one with similar probability.

The attack is on 24 bits of the key and we expect $p_{fa} \cdot 2^{24}$ many subkeys to get hits less than or equivalent to threshold T. The probability of satisfying the improbable

differential for a wrong subkey is

$$p = \frac{1}{36}\frac{6}{8}\frac{6}{8} \cdot 3^{-3} \cdot 2^{-2} \cdot 2^{-13} = 3^{-3} \cdot 2^{-21}.$$

We use Algorithm 4.1 from Section 4.4 to estimate required number of pairs N to attack with a given success probability. To minimize the time complexity, we select $p_{nd} = 0.001$ and $p_{fa} = 2^{-25}$ as the input of the Algorithm 4.1 and we obtain $N = 2^{67.62}$ and $T = 4015639374946 \leq 2^{42}$. Thus, the data complexity of the attack is $2^{67.62-23+16} \cdot 3^{-3} \approx 2^{55.87}$ chosen plaintexts and memory complexity is about 2^{24} 42-bit counters. Time complexity of the attack is dominated by step 3. At this step, instead of trying every possible 16-bit subkey for a pair, we can create a table to store pairs of 16-bit inputs with the desired difference that provide the desired output difference (size of such a table is negligible when compared to the size of the counters). Hence, the values of the keys can be obtained by one table look-up for every pair. Since we repeat this step for $2^{54.62}$ pairs, time complexity of the attack is $2^{54.62}$ memory accesses.

5.3 Attack on PRESENT-80-13

The 12-round improbable attack can be extended to a 13-round attack by putting one more round with 2^{-4} probability at the top of our 5-round characteristic. We select $p_{nd} = 0.001$ and $p_{fa} = 2^{-25}$ and the data complexity becomes $2^{63.86}$ chosen plaintexts, memory complexity is about 2^{24} 50-bit counters and the time complexity is $2^{62.62}$ memory accesses. Data complexity of these improbable differential attacks can be reduced by reducing p_{nd} or increasing p_{fa}. Attacks on PRESENT-80 are summarized in Table 5.5.

Table 5.5: Summary of the Attacks on PRESENT-80

#Rounds	Attack Type	Data	Time	Memory	Success	Reference
12	Improbable Differential	$2^{55.87}$ CP	$2^{54.62}$ MA	$42 \cdot 2^{24}$ bits	99.9%	Sect. 5.2
13	Improbable Differential	$2^{63.85}$ CP	$2^{62.62}$ MA	$50 \cdot 2^{24}$ bits	99.9%	Sect. 5.3
16	Differential	2^{64} CP	2^{64} MA	$6 \cdot 2^{32}$ bits	99.99%	[84]
18	Multiple Differential	2^{64} CP	$2^{71.72}$ En	2^{32} blocks	94%	[19]
24	Linear (for weak keys)	$2^{63.5}$ KP	2^{40} En	2^{40} blocks	95%	[60]
24	Statistical Saturation (theoretical)	2^{57} CP	2^{57} En	2^{32} counters	n/a	[27]
26	Multiple Linear	2^{64} KP	2^{72} En	2^{31} blocks	95%	[26]

Note that the same improbable differential attacks can be applied to PRESENT-128 with similar complexities.

CHAPTER 6

ATTACKS ON CLEFIA

In this chapter, we present 10-round improbable differentials and introduce an improbable differential attack on 13-round CLEFIA with key length of 128 bits. We also introduce improbable differential attacks on 14 and 15-round CLEFIA for key lengths 196 and 256 bits. Moreover, we provide a practical improbable differential attack on 6-round CLEFIA . In these attacks our aim is to derive the round keys and we do not consider the key scheduling part as done in [69, 80, 81].

These attacks were published in 2010 in [74] and they were the best attacks on CLEFIA. In Section 6.6 of this chapter, we provide the improved improbable differential attacks on CLEFIA-128, CLEFIA-192, and CLEFIA-256 and to the best of our knowledge, these are the best attacks on this cipher. These attacks show the power and importance of the improbable differential cryptanalysis.

6.1 10-round Improbable Differentials

We will use the following two 9-round impossible differentials that are introduced in [80],

$$\Delta_1 : [0_{(32)}, 0_{(32)}, 0_{(32)}, [X, 0, 0, 0]_{(32)}] \nrightarrow_{9r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [0, Y, 0, 0]_{(32)}]$$
$$\Delta_2 : [0_{(32)}, 0_{(32)}, 0_{(32)}, [0, 0, X, 0]_{(32)}] \nrightarrow_{9r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [0, Y, 0, 0]_{(32)}]$$

where $X_{(8)}$ and $Y_{(8)}$ are non-zero differences. We obtain 10-round improbable differentials by adding the following one-round differentials to the top of these 9-round impossible differentials,

$$\Delta_3 : [[\psi, 0, 0, 0]_{(32)}, \zeta_{(32)}, 0_{(32)}, 0_{(32)}] \rightarrow_{1r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [\psi, 0, 0, 0]_{(32)}]$$
$$\Delta_4 : [[0, 0, \psi, 0]_{(32)}, \zeta'_{(32)}, 0_{(32)}, 0_{(32)}] \rightarrow_{1r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [0, 0, \psi, 0]_{(32)}]$$

which hold when the output difference of the F_0 function is ζ (resp. ζ') when the input difference is $[\psi, 0, 0, 0]$ (resp. $[0, 0, \psi, 0]$). We choose ψ and corresponding ζ and ζ' depending on the DDT of S_0 in order to increase the probability of the differential. One can observe that the values 10, 8, 6 and 4 appear 9, 119, 848 and 5037 times in the

DDT of S_0, respectively. So we have $9 + 119 + 848 + 5037 = 6013$ nonzero entries. When ψ, ζ and ζ' are chosen according to these differences, the average probability of the 10-round improbable differentials becomes

$$p' = \frac{9 \cdot 10 + 119 \cdot 8 + 848 \cdot 6 + 5037 \cdot 4}{256 \cdot 6013} \approx 2^{-5.87}.$$

6.2 Improbable Differential Attack on 13-Round CLEFIA

We put one additional round on the plaintext side and two additional rounds on the ciphertext side of the 10-round improbable differentials to attack first 13 rounds of CLEFIA that captures RK_1, $RK_{23,1} \oplus WK_{2,1}$, RK_{24}, and RK_{25}.

We place the whitening key WK_2 at the XOR with the 11th-round output word $x^{\{11,2\}}$ and XOR with RK_{23}. Moreover, we place the whitening key WK_1 at the XOR with the first round output word $x^{\{1,2\}}$, as shown in Fig. 6.1. These movements are equivalent transformations.

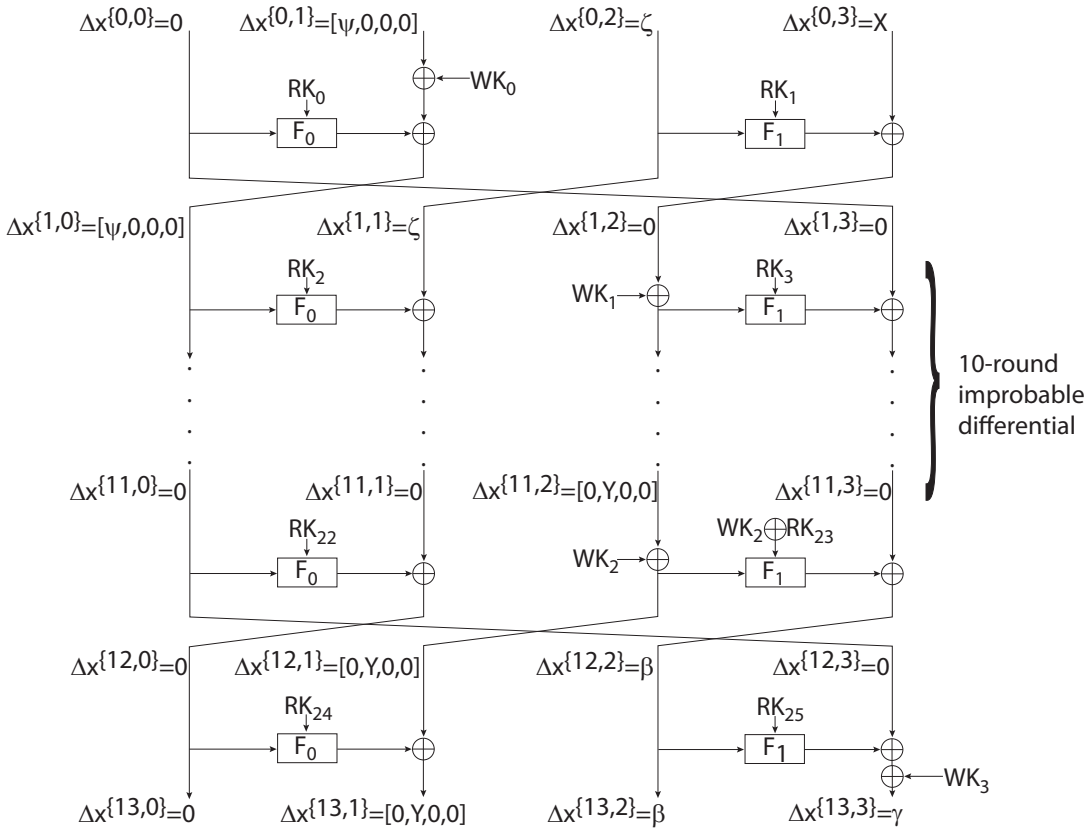

Figure 6.1: Improbable differential attack on 13-round CLEFIA

6.2.1 Data Collection

For a single choice of ψ and corresponding ζ values, we choose 2^K structures of plaintexts where the first word $x^{\{0,0\}}$ and the second, third and fourth bytes of the second word $x^{\{0,1\}}$ are fixed (similarly, we fix the first, second and fourth bytes of the second word $x^{\{0,1\}}$ for a choice of ψ and ζ'). We construct pairs where the first byte (resp. third byte) of the second word $x^{\{0,1\}}$ has the difference ψ, the third word $x^{\{0,2\}}$ has the difference ζ (resp. ζ') and the fourth word $x^{\{0,3\}}$ has the same difference with the output difference of F_1, which is obtained from the guessed round key RK_1, when the input difference of F_1 is ζ (resp. ζ'). Such a structure proposes $2 \cdot 6013 \cdot 2^{71}$ pairs.

We keep the ciphertext pairs having the difference $[0, [0, Y, 0, 0], \beta, \gamma]$ where γ is nonzero and β represents every 255 difference value that can be obtained from the multiplication of M_1 with $[0, Y, 0, 0]^t$. Such a difference in the ciphertext pairs is observed with a probability of

$$\frac{1}{2^{32}} \cdot \frac{255}{2^{32}} \cdot \frac{255}{2^{32}} \cdot \frac{2^{32}-1}{2^{32}} \approx 2^{-80}.$$

Therefore, $6013 \cdot 2^{K-8}$ pairs remain.

6.2.2 Key Recovery

We keep counters for $RK_{23,1} \oplus WK_{2,1}|RK_{24}|RK_{25}$ for every guess of RK_1 and increase the corresponding counter when the improbable differential is obtained with a guessed key. Keys satisfying the improbable differential are obtained by differential table look-ups indexed on the input and the output differences of the 12th-round F_1 and 13th-round F_1. The probability of satisfying the improbable differential for a wrong key is $p = 2^{-40}$ from the average probabilities 2^{-8} and 2^{-32} for the 12th and 13th-round F_1 functions respectively. Therefore, the probability of obtaining the improbable differential for the correct key is $p_0 = p \cdot (1 - p') \approx 2^{-40.02}$.

During the attack we try to obtain the 104-bit round key, namely RK_1, $RK_{23,1} \oplus WK_{2,1}$, RK_{24}, RK_{25} and for the correct key to get the least number of hits, false alarm probability p_{fa} must be less than 2^{-104}. Feeding the Algorithm 4.1 with the inputs p, p_0, $p_{fa} = 2^{-105}$, and $p_{nd} = 0.01$ shows that when the threshold T is $673474 < 2^{20}$, $N \approx 2^{59.38}$ pairs are needed for the correct key to remain below the threshold and all of the wrong ones to remain above it with a success probability of 99%.

6.2.3 Attack Complexity

With the 2^{80} ciphertext filtering conditions, we need $2^{80} \cdot 2^{59.38} = 2^{139.38}$ pairs to perform the attack. Since we have 6013 choices for ψ, we need $2^K \approx 2^{54.83}$ structures so that $6013 \cdot 2^{72+K} = 2^{139.38}$. Hence, the data complexity of the attack is $2^{126.83}$ chosen plaintexts.

For every guess of RK_1 and RK_{24} and for every choice of ψ, we perform $2^{59.38}$ F-function computations which is

$$2^{64} \cdot 2^{59.38} \cdot \frac{1}{2} \cdot \frac{1}{13} \approx 2^{118.68}$$

encryptions. However, the time complexity is $2^{126.83}$ encryptions for obtaining the ciphertexts.

The memory complexity of the attack comes from the 20-bit counters kept for the 104-bit round keys $RK_1|RK_{23,1} \oplus WK_{2,1}|RK_{24}|RK_{25}$, which require $20 \cdot 2^{104} \approx 2^{108.32}$ bits.

6.3 Improbable Differential Attack on 14-Round CLEFIA

We expand our 13-round attack by one round on the ciphertext side to break 14-round CLEFIA for the key length of 192 or 256 bits. This attack captures 168 bits of the round keys, namely RK_1, $RK_{23,1}$, $RK_{24} \oplus WK_3$, $RK_{25} \oplus WK_2$, RK_{26}, and RK_{27}.

We move the whitening keys WK_1, WK_2, and WK_3 in the same way as in the 13-round attack.

6.3.1 Data Collection

We generate pairs in the same way as in the 13-round attack and we want 13th-round output difference to be $[[0, Y, 0, 0], \beta, \gamma, 0]$ to perform the attack. Consequently, we keep the ciphertext pairs satisfying the difference $[[0, Y, 0, 0], \beta', \gamma, \delta]$ where γ and δ are non-zero and β' is the XOR of β with the 255 possible values that can be obtained from the multiplication of M_0 with $[0, Y, 0, 0]^t$. Such a difference in ciphertext pairs is observed with a probability of

$$\frac{255}{2^{32}} \cdot \frac{255 \cdot 255}{2^{32}} \cdot \frac{2^{32}-1}{2^{32}} \cdot \frac{2^{32}-1}{2^{32}} \approx 2^{-40}.$$

Therefore, $6013 \cdot 2^{K+32}$ pairs remain.

6.3.2 Key Recovery

We guess the second byte of RK_{24} and check if the second word of the output of 13th-round has difference β. The probability of this event is 2^{-8} and therefore, $6013 \cdot 2^{K+24}$ pairs remain. In order to check whether the 72-bit key $RK_{23,1}|RK_{25} \oplus WK_2|RK_{27}$ satisfies the improbable differential, we use differential tables indexed on the input and output differences of the 12th-round, 13th-round and 14th-round F_1 functions. The input values of these F_1 functions are obtained by the guesses of $RK_{24} \oplus WK_3$ and

the first, third and fourth bytes of RK_{26}. The input of the 13th-round F_0 is obtained from RK_{27} candidates.

The probability of a candidate key to satisfy the improbable differential using three F_1 differential tables is $p = 2^{-72}$ from the average probabilities 2^{-8}, 2^{-32} and 2^{-32} for the 12th, 13th and 14th-round F_1 functions, respectively. Feeding the Algorithm 4.1 with the inputs p, p_0, $p_{fa} = 2^{-169}$, and $p_{nd} = 0.01$ shows that when the threshold T is $1022026 < 2^{20}$, $N \approx 2^{91.98}$ pairs are needed for the correct key to remain below the threshold and all of the wrong ones to remain above it with a success probability of 99%.

Keeping a key table for the attacked 168 key bits would require a memory that exceeds 2^{128} blocks where a block is 128 bits long. For this reason, we keep all of the $2^{127.43}$ plaintexts in a table, then guess RK_1 and choose the plaintext pairs for the attack.

6.3.3 Complexity

We need $2^{91.98+40+8} = 2^{139.98}$ pairs in total to perform the attack. Since we have 6013 choices for ψ, we need $2^K \approx 2^{55.43}$ structures so that $6013 \cdot 2^{72+K} = 2^{139.98}$. Hence, the attack has data complexity of $2^{127.43}$ chosen plaintexts.

For every guess of RK_1, $RK_{24} \oplus WK_3$, and RK_{26}, we perform $2^{91.98}$ F-function computations which is

$$2^{96} \cdot 2^{91.98} \cdot \frac{1}{2} \cdot \frac{1}{14} \approx 2^{183.17}$$

encryptions.

We keep 20-bit counters for the 72-bit keys $RK_{23,1}|RK_{25} \oplus WK_2|RK_{27}$ but the memory complexity is dominated by the ciphertext table of $2^{127.43}$ blocks.

6.4 Improbable Differential Attack on 15-Round CLEFIA

We expand the 14-round improbable differential attack by one round on the ciphertext side to attack 15-round CLEFIA in which we exhaustively search for the 15th-round keys RK_{28} and RK_{29}. Our aim is to obtain the value of the 232-bit round key, namely RK_1, $RK_{23,1}$, RK_{24}, RK_{25}, $RK_{26} \oplus WK_3$, $RK_{27} \oplus WK_2$, RK_{28} and RK_{29}.

We move the whitening keys WK_1, WK_2, and WK_3 in the same way as in the 14-round attack.

For the inputs $p = 2^{-72}$, p_0, $p_{fa} = 2^{-233}$, and $p_{nd} = 0.01$, Algorithm 4.1 produces the outputs $N \approx 2^{92.40}$ and $T = 1361613 < 2^{21}$. Hence, the data complexity of the attack is $2^{127.85}$ chosen plaintexts and the memory complexity is $2^{127.85}$ blocks.

The time complexity of the attack comes from $2^{92.40}$ F-function computations for RK_1,

RK_{24}, $RK_{26} \oplus WK_3$ guesses and the exhaustive search of RK_{28} and RK_{29}, which is

$$2^{92.40} \cdot 2^{96} \cdot 2^{64} \cdot \frac{1}{2} \cdot \frac{1}{15} \approx 2^{247.49}$$

encryptions.

6.5 Practical Improbable Differential Attack on 6-Round CLEFIA

From the 9-round impossible differential that was used in Section 6.1, one can easily obtain the following 4-round impossible differential

$$[0_{(32)}, 0_{(32)}, 0_{(32)}, [\psi, 0, 0, 0]_{(32)}] \nrightarrow_{4r} [?_{(32)}, ?_{(32)}, ?_{(32)}, \psi'_{(32)}]$$

where ψ' is any difference other than ψ. We obtain a 5-round improbable differential by adding the following 1-round differential

$$[[\psi, 0, 0, 0]_{(32)}, \zeta_{(32)}, 0_{(32)}, 0_{(32)}] \rightarrow_{1r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [\psi, 0, 0, 0]_{(32)}]$$

to the top of the 4-round impossible differential. The differential holds with probability $p' = \frac{10}{256}$ if we choose $\psi = 08000000_x$ and $\zeta = 7EFCE519_x$.

In order to attack 6-round CLEFIA, we prepare plaintext pairs with the difference $[[\psi, 0, 0, 0]_{(32)}, \zeta_{(32)}, 0_{(32)}, 0_{(32)}]$. Then we guess RK_{11} and increase the counter for the guessed RK_{11} if $X_{5,3}$ has the difference $\psi'_{(32)}$. We expect the correct RK_{11} to have the smallest counter.

Feeding the Algorithm 4.1 with the inputs $p = 1 - 2^{-32}$, $p_0 = p \cdot (1 - \frac{10}{256})$, $p_{fa} = 2^{-33}$, and $p_{nd} = 0.01$ shows that when the threshold T is 184, $N = 185$ pairs are needed for the correct RK_{11} to remain below the threshold and all of the wrong ones to remain above it with a success probability of 99%.

6.5.1 Summary of Attacks

Results of these improbable differential attacks and the impossible differential attacks of [80] on CLEFIA are summarized in Table 6.1.

6.6 Improved Improbable Differential Attacks on CLEFIA-128, CLEFIA-192, and CLEFIA-256

In previous section of this Chapter, we provided the improbable differential attacks on CLEFIA as it was published in [74]. The main problem with these attacks is the high

Table 6.1: Results of the impossible differential attacks of [80] and improbable differential attacks on CLEFIA

#Rounds	Attack	Key Size	Data	Time	Memory	Success	Reference
12	Impossible	128, 192, 256	$2^{118.9}$ CP	2^{119} En	2^{73} blocks	-	[80]
13	Improbable	128, 192, 256	$2^{126.83}$ CP	$2^{126.83}$ En	$2^{101.32}$ blocks	99%	Sect. 6.2
13	Impossible	192, 256	$2^{119.8}$ CP	2^{146} En	2^{120} blocks	-	[80]
14	Improbable	192, 256	$2^{127.43}$ CP	$2^{183.17}$ En	$2^{127.43}$ blocks	99%	Sect. 6.3
14	Impossible	256	$2^{120.3}$ CP	2^{212} En	2^{121} blocks	-	[80]
15	Improbable	256	$2^{127.85}$ CP	$2^{247.49}$ En	$2^{127.85}$ blocks	99%	Sect. 6.4

data complexity. In this section we improve these attacks by modifying the impossible differential and characteristic that are combined using the expansion technique. For the attack on CLEFIA-128, we also use a weakness in the key schedule discovered by Zhang *et al.* [88].

6.6.1 Modifying the Impossible Differential

Instead of Δ_1 and Δ_2, we are going to use the following two 9-round impossible differentials of [81],

$$\Delta_1' : [0_{(32)}, 0_{(32)}, 0_{(32)}, [X, 0, 0, 0]_{(32)}] \nrightarrow_{9r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [Y, 0, Z, 0]_{(32)}], p = 2^{-112}$$
$$\Delta_2' : [0_{(32)}, 0_{(32)}, 0_{(32)}, [0, 0, X, 0]_{(32)}] \nrightarrow_{9r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [Y, 0, Z, 0]_{(32)}], p = 2^{-112}$$

where $X_{(8)}$, $Y_{(8)}$, and $Z_{(8)}$ are non-zero differences. This change reduces p by a factor of 2^8. Thus, the required number of pairs and therefore the data complexity of the attacks reduce by a factor of 2^8. Since the key guess step is repeated for every pair, time complexity of this step reduces by a factor of 2^8, too. Due to the change in the output difference of the impossible differential, this time we keep a differential table for $RK_{23,0} \oplus WK_{2,0} | RK_{23,2} \oplus WK_{2,2} | RK_{25}$ instead of $RK_{23,1} \oplus WK_{2,1} | RK_{25}$, which has negligible effect on the memory complexity. However, since we are guessing 8 more bits, we decrease our false alarm probability p_{fa} and this results in a slight increase in data and time complexity, namely around a factor of $2^{0.04}$.

This improvement is valid for our 13, 14, and 15 round attacks and the main improvement in the complexities comes from this step. The improved improbable differential attack on CLEFIA reduced to 13 rounds is shown in Fig. 6.2.

6.6.2 Modifying the Expansion

Similar to what was described in Section 6.1, we obtain 10-round improbable differentials by adding the following one-round differentials to the top of these 9-round impossible differentials,

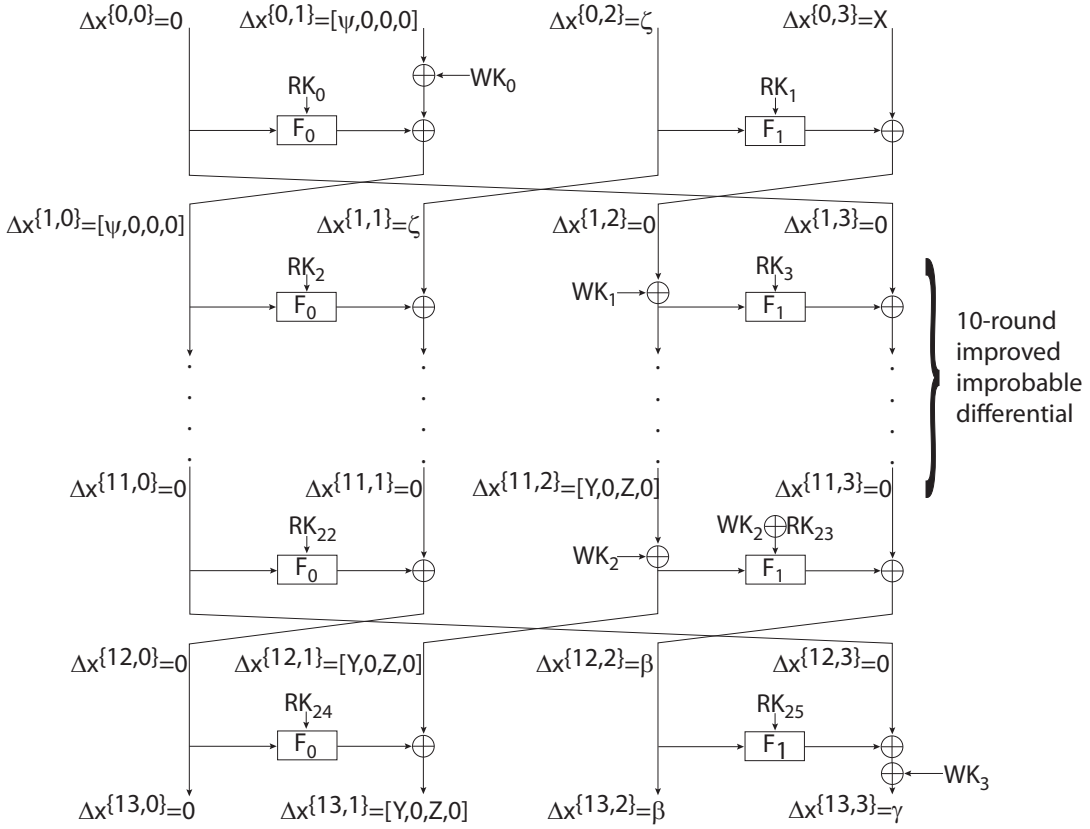

Figure 6.2: Improved improbable differential attack on 13-round CLEFIA

$$\Delta_3 : [[\psi, 0, 0, 0]_{(32)}, \zeta_{(32)}, 0_{(32)}, 0_{(32)}] \rightarrow_{1r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [\psi, 0, 0, 0]_{(32)}]$$
$$\Delta_4 : [[0, 0, \psi, 0]_{(32)}, \zeta'_{(32)}, 0_{(32)}, 0_{(32)}] \rightarrow_{1r} [0_{(32)}, 0_{(32)}, 0_{(32)}, [0, 0, \psi, 0]_{(32)}]$$

which hold when the output difference of the F_0 function is ζ (resp. ζ') when the input difference is $[\psi, 0, 0, 0]$ (resp. $[0, 0, \psi, 0]$). We choose ψ and corresponding ζ and ζ' depending on the DDT of S_0. One can observe that the values 10, 8, 6, 4, and 2 appear 9, 119, 848, 5037, and 19501 times in the DDT of S_0, respectively. So we have $9 + 119 + 848 + 5037 + 19501 = 25514$ nonzero entries. In the original attacks, only the differences observed for 10, 8, 6, and 4 times in DDT are considered to have higher p' which was around $2^{-5.87}$. If we also take into account the differences that have the value 2 in the DDT then the probability p' decreases to

$$p' = \frac{9 \cdot 10 + 119 \cdot 8 + 848 \cdot 6 + 5037 \cdot 4 + 19501 \cdot 2}{256 \cdot 25514} \approx 2^{-6.64}.$$

Although this change decreases p' and therefore increases the number of pairs required to perform the attack by a factor of $2^{1.55}$, due to the new ψ and ζ values we can generate $2^{2.09}$ more pairs. Thus, this change also decreases the data and time complexity of the attacks by a factor of $2^{2.09-1.55} = 2^{0.54}$. The intuition for this improvement comes from Proposition 4.1.

This improvement is valid for our 13, 14, and 15-round attacks.

6.6.3 Key Schedule Weakness of CLEFIA-128

In [88], authors claimed an attack on 14-round CLEFIA-128 without the whitening keys but CLEFIA design team pointed out a flaw in their attack. Although the proposed attack was not better than any generic attack, authors' idea of attacking the intermediate key L instead of the master key K showed a weakness in the key schedule of CLEFIA-128 for the first time. This weakness in the key schedule was later used in [51] and [73] to extend the 12-round impossible differential attacks to 13-round impossible differential attacks.

If we represent the bits of the 128-bit intermediate key L as $l_{0\sim127}$, then the relation between L and RK_1 and RK_{24} can be shown as follows: $RK_1 = l_{32\sim63} \oplus c$ and $RK_{24} = l_{42\sim63}|l_{121\sim127}|l_{114\sim116} \oplus c'$ for some constants c and c'.

Thus, the guessed round keys RK_1 and RK_{24} have 22 bits in common. Therefore, we do not need to guess these common bits twice in our 13-round attack on CLEFIA-128. Hence, the attack is on 74 bits of L and the 16 bits $RK_{23,0} \oplus WK_{2,0}|RK_{23,2} \oplus WK_{2,2}$, instead of 112 bits of round keys. When attacking 112 bits of the round key in the original attack, false alarm probability was chosen as $p_{fa} = 2^{-113}$ to have the correct round key below the threshold T and all wrong round keys above it. However, when attacking 90 bits, we can choose $p_{fa} = 2^{-30}$ to have 30 bits of advantage and then we obtain the whole 128-bit L by exhaustive search.

Table 6.2: Relation between round keys and intermediate key L for CLEFIA-128 (common bits are shown in bold)

	Corresponding Bits of Intermediate Key L															
RK_1	32	33	34	35	36	37	38	39	40	41	**42**	**43**	**44**	**45**	**46**	**47**
	48	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**
RK_{24}	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**
	58	**59**	**60**	**61**	**62**	**63**	121	122	123	124	125	126	127	114	115	116
RK_{25}	117	118	119	120	107	108	109	110	111	112	113	100	101	102	103	104
	105	106	93	94	95	96	97	98	99	86	87	88	89	90	91	92

CLEFIA has different key schedules for CLEFIA-192 and CLEFIA-256 and the key schedule weakness of CLEFIA-128 is not valid for CLEFIA-192 and CLEFIA-256.

The input and output values of Algorithm 4.1 for these improved attacks are listed in Table 6.3.

With these N and T values, the resulting data, time and memory complexities of the improved attacks are explained in the following subsections and summarized in Table 6.4.

Table 6.3: List of Algorithm 4.1 input and outputs of the improbable differential attacks on CLEFIA

#Rounds	p	p'	p_{fa}	p_{nd}	N	T	Reference
13	2^{-120}	$2^{-5.87}$	2^{-105}	0.01	$2^{139.38}$	673474	Sect.6.2
13	2^{-112}	$2^{-6.64}$	2^{-113}	0.01	$2^{133.03}$	2112318	Sect.6.6
13	2^{-112}	$2^{-6.64}$	2^{-30}	0.01	$2^{131.42}$	676646	Sect.6.6
14	2^{-120}	$2^{-5.87}$	2^{-169}	0.01	$2^{139.98}$	1022026	Sect.6.3
14	2^{-112}	$2^{-6.64}$	2^{-177}	0.01	$2^{133.59}$	3134325	Sect.6.6
15	2^{-120}	$2^{-5.87}$	2^{-233}	0.01	$2^{140.40}$	1361613	Sect.6.4
15	2^{-112}	$2^{-6.64}$	2^{-241}	0.01	$2^{133.99}$	4131860	Sect.6.6

6.6.4 Complexity of the Improved Attacks

6.6.4.1 13-round Attack on CLEFIA-128

For the 13-round attack on CLEFIA-128, we use all three of the above improvements. For the inputs $p = 2^{-112}$, p_0, $p_{fa} = 2^{-30}$, and $p_{nd} = 0.01$, Algorithm 4.1 produces the outputs $N \approx 2^{131.42}$ and $T = 676646 < 2^{20}$. Thus, with the 2^{64} ciphertext filtering conditions, we need $2^{64} \cdot 2^{67.42} = 2^{131.42}$ pairs to perform the attack. Since we have 25514 choices for ψ, we need $2^K \approx 2^{44.78}$ structures so that $25514 \cdot 2^{72+K} = 2^{131.42}$. Hence, the data complexity of the attack is $2^{116.78}$ chosen plaintexts.

For every guess of RK_1 and RK_{24} and for every choice of ψ, we perform $2^{67.42}$ F function computations which is

$$2^{64-22} \cdot 2^{67.42} \cdot \frac{1}{2} \cdot \frac{1}{13} \approx 2^{104.72}$$

encryptions. However, we need to perform $2^{116.78}$ encryptions to obtain plaintext-ciphertext pairs and $2^{90-30+38+16} = 2^{114}$ encryptions for exhaustive search of the full key. Thus, the expected time complexity of the attack is $2^{116.78}+2^{114}+2^{104.72} \approx 2^{116.98}$ encryptions.

The memory required for storing the 20-bit counters kept for the 90-bit round keys $RK_1|RK_{23,0} \oplus WK_{2,0}|RK_{23,2} \oplus WK_{2,2}|RK_{24}|RK_{25}$ is $20 \cdot 2^{90} \approx 2^{94.32}$ bits.

If we only use the first two improvements and do not use the key schedule weakness, then the attack becomes valid for all key sizes. But this time we need to choose $p_{fa} = 2^{-113}$ so that only the correct round key remains below the threshold T. This time the Algorithm 4.1 outputs become $N = 2^{133.03}$ and $T = 2112318 < 2^{22}$, and we get data and time complexities as $2^{118.39}$ with a memory complexity of $22 \cdot 2^{112} \approx 2^{116.46}$ bits.

6.6.4.2 14-round Attack on CLEFIA-192 and CLEFIA-256

For the inputs $p = 2^{-112}$, p_0, $p_{fa} = 2^{-177}$, and $p_{nd} = 0.01$, Algorithm 4.1 produces the outputs $N \approx 2^{133.59}$ and $T = 3134325 < 2^{22}$. Hence, the data complexity of the attack is $2^{118.95}$ chosen plaintexts and the memory complexity is $2^{118.95}$ blocks.

For every guess of RK_1, $RK_{24} \oplus WK_3$, and RK_{26}, we perform $2^{91.98}$ F function computations which is

$$2^{96} \cdot 2^{85.49} \cdot \frac{1}{2} \cdot \frac{1}{14} \approx 2^{177.68}$$

encryptions.

6.6.4.3 15-round Attack on CLEFIA-256

For the inputs $p = 2^{-112}$, p_0, $p_{fa} = 2^{-241}$, and $p_{nd} = 0.01$, Algorithm 4.1 produces the outputs $N \approx 2^{133.99}$ and $T = 4131860 < 2^{22}$. Hence, the data complexity of the attack is $2^{119.35}$ chosen plaintexts and the memory complexity is $2^{119.35}$ blocks.

The time complexity of the attack comes from $2^{85.99}$ F function computations for RK_1, RK_{24}, $RK_{26} \oplus WK_3$ guesses and the exhaustive search of RK_{28} and RK_{29}, which is

$$2^{85.99} \cdot 2^{96} \cdot 2 \cdot 2^{64} \cdot \frac{1}{2} \cdot \frac{1}{15} \approx 2^{242.08}$$

encryptions.

Table 6.4: Comparison of our attack with the previous attacks on CLEFIA. Our attack is among the deepest penetrating attacks on all key sizes of CLEFIA. Furthermore, it has the best data and time complexities on all versions.

#Rounds	Attack	Key Size	Data	Time	Memory	Success	Reference
12	Impossible	All	$2^{118.9}$ CP	2^{119} En	2^{73} blocks	-	[80]
12	Impossible	All	2^{108} CP	2^{108} En	2^{99} blocks	-	[81]
13	**Improbable**	**All**	$\mathbf{2^{126.83}}$ **CP**	$\mathbf{2^{126.83}}$ **En**	$\mathbf{2^{101.32}}$ **blocks**	99%	**Sect. 6.2**
13	Impossible	128	$2^{119.4}$ CP	$2^{125.52}$ En	$2^{119.4}$ blocks	-	[73]
13	Impossible	128	$2^{117.8}$ CP	$2^{121.2}$ En	$2^{86.8}$ blocks	-	[51]
13	**Improbable**	**All**	$2^{118.39}$ **CP**	$\mathbf{2^{118.39}}$ **En**	$\mathbf{2^{109.46}}$ **blocks**	99%	**Sect. 6.6**
13	**Improbable**	**128**	$\mathbf{2^{116.78}}$ **CP**	$\mathbf{2^{116.98}}$ **En**	$\mathbf{2^{87.32}}$ **blocks**	99%	**Sect. 6.6**
13	Impossible	192, 256	$2^{119.8}$ CP	2^{146} En	2^{120} blocks	-	[80]
14	**Improbable** [1]	**192, 256**	$\mathbf{2^{127.43}}$ **CP**	$\mathbf{2^{183.17}}$ **En**	$\mathbf{2^{127.43}}$ **blocks**	99%	**Sect.6.3**
14	Multidim. ZC	192, 256	$2^{127.5}$ KP	$2^{180.2}$ En	2^{111} blocks	-	[21]
14	**Improbable**	**192, 256**	$\mathbf{2^{118.95}}$ **CP**	$\mathbf{2^{177.68}}$ **En**	$\mathbf{2^{118.95}}$ **blocks**	99%	**Sect. 6.6**
14	Impossible	256	$2^{120.3}$ CP	2^{212} En	2^{121} blocks	-	[80]
15	**Improbable** [2]	**256**	$\mathbf{2^{127.85}}$ **CP**	$\mathbf{2^{247.49}}$ **En**	$\mathbf{2^{127.85}}$ **blocks**	99%	**Sect. 6.4**
15	Multidim. ZC	256	$2^{127.5}$ KP	$2^{244.08}$ En	2^{111} blocks	-	[21]
15	**Improbable**	**256**	$\mathbf{2^{119.35}}$ **CP**	$\mathbf{2^{242.08}}$ **En**	$\mathbf{2^{119.35}}$ **blocks**	99%	**Sect. 6.6**

[1] Due to a calculation error, in [74] the data and memory complexities of this attack was reported as $2^{126.98}$ instead of $2^{127.43}$

[2] Due to a calculation error, in [74] the data and memory complexities of this attack was reported as $2^{127.40}$ instead of $2^{127.85}$

CHAPTER 7

ATTACKS ON SERPENT

In this chapter, we present 7-round improbable differential attack on SERPENT that exploit undisturbed bits of SERPENT's S-boxes. Moreover, we show the importance of differential factors by correcting the advantage and improving the time complexity of Dunkelman *et al.*'s differential-linear attacks on SERPENT by using the differential factors of SERPENT's S-boxes.

7.1 Improbable Differential Attacks on SERPENT

7.1.1 5.5-Round Impossible Differential

Without using undisturbed bits, the longest impossible differential we could find on SERPENT had a length of 3.5 rounds. However, we obtained four 5.5-round impossible differentials on SERPENT with the help of undisturbed bits and one of them is shown in detail in Table 2.6. Here S_i's are the differences after the S_i operations, LT represents the differences after the linear transformation and question marks represent indeterminate bit differences. The miss-in-the-middle is observed at the 13th bit of X_3 after round 2. Note that 5.5-round impossible differentials we have found are infeasible to mount an attack since $p = 2^{-128}$. Instead, by eliminating the last round of the 5.5-round impossible differential, we obtained a 4.5-round impossible differential with $p = 2^{-100}$ and used it to construct a 5.5-round improbable differential with probability $p' = 2^{-4}$. We use this improbable differential to attack SERPENT reduced to 7 rounds. This change is due to the observation that an improbable differential attack that uses the expansion technique has data complexity $O((p')^2 \cdot p)^{-1}$, which is obtained in Section 4.3.2. The 4.5-round impossible differential that we use in our attack is shown in detail in Table 7.1.

7.1.2 7-Round Improbable Differential Attack

1. Choose 2^n structures of 2^{24} plaintexts each where bitslices b_0, b_3, b_6, b_7, b_{20} and b_{23} take values L_0, L_3, L_6, L_7, L_{20} and L_{23}, respectively and other bits are fixed. Here

Table 7.1: A 4.5-Round Impossible Differential for SERPENT

Input	X_0: 0000	0000	0000	0000	0000	0000	0000	0000
	X_1: 0001	0000	0000	0000	0000	0000	0000	0000
	X_2: 0000	0010	0000	0000	0000	0000	0000	0000
	X_3: 0001	0000	0000	0000	0000	0000	0000	0000
LT	X_0: 0000	0000	0000	0000	0000	0000	0000	0000
	X_1: 0000	0000	0000	0000	0000	0000	0000	0000
	X_2: 0000	0000	0000	0100	0000	0000	0000	0000
	X_3: 0000	0000	0000	0000	0000	0000	0000	0000
S_1	X_0: 0000	0000	0000	0?00	0000	0000	0000	0000
	X_1: 0000	0000	0000	0?00	0000	0000	0000	0000
	X_2: 0000	0000	0000	0100	0000	0000	0000	0000
	X_3: 0000	0000	0000	0?00	0000	0000	0000	0000
LT	X_0: 0?00	100?	0000	0000	0000	0000	00??	0010
	X_1: 0000	0000	0100	?000	0000	0000	0000	000?
	X_2: 00?0	0000	0000	110?	?000	1000	0000	0000
	X_3: 0001	00?0	0000	0000	0000	0000	0000	0000
S_2	X_0: 0??1	?0??	0100	??0?	?000	?000	00??	00??
	X_1: 0???	?0??	0?00	??0?	?000	?000	00??	00??
	X_2: 0???	?0??	0?00	??0?	?000	?000	00??	00??
	X_3: 0???	?0??	0?00	??0?	?000	?000	00??	00??
LT	X_0: ????	????	????	????	????	????	????	????
	X_1: ????	0??0	??0?	????	??0?	?1??	????	0???
	X_2: ????	????	????	????	????	????	1???	????
	X_3: ?0??	????	????	1?0?	??**1**?	??0?	????	??0?
	Impossible							
LT	X_0: ????	????	????	????	??0?	????	????	????
	X_1: ????	????	????	????	??0?	????	????	????
	X_2: ????	????	????	????	??0?	????	????	????
	X_3: ????	????	????	????	??**0**?	????	????	????
S_3	X_0: ??00	????	????	????	??0?	????	????	?0?0
	X_1: ????	????	????	????	??0?	????	????	????
	X_2: ????	????	????	??0?	??0?	?0??	????	????
	X_3: ????	????	????	????	??0?	????	????	????
LT	X_0: 0???	?0??	???0	?0?0	000?	?00?	?0??	0???
	X_1: 0???	?0??	???0	?0?0	000?	?00?	?0??	0???
	X_2: 0???	?0??	???0	?0?0	000?	?00?	?0??	0???
	X_3: 0???	?0??	???0	?0?0	000?	?00?	?0??	0???
S_4	X_0: 0000	?00?	?0?0	00?0	000?	?00?	0000	0000
	X_1: 0???	?0??	0??0	0000	0000	?00?	?0??	0???
	X_2: 0??0	?0??	0??0	?000	0000	000?	00??	0??0
	X_3: 0???	?0??	??00	00?0	0000	?00?	?0??	00??
LT	X_0: 0??0	0?00	0000	0000	00?0	0??0	?000	?000
	X_1: 0?00	00?0	0000	0000	0000	0000	0000	0000
	X_2: 0??0	??00	??0?	0??0	??0?	0000	0000	00?0
	X_3: 0?00	000?	0000	0000	0000	000?	00??	0000

$$
\begin{aligned}
L_0 \in &\ \{3_x, 5_x, 6_x, B_x, D_x, E_x\} \\
L_3 \in &\ \{2_x, 4_x, 7_x, 9_x, A_x, C_x\} \\
L_6 \in &\ \{1_x, 3_x, 6_x, 8_x, C_x, E_x, F_x\} \\
L_7 \in &\ \{3_x, 6_x, 7_x, 9_x, B_x, C_x\} \\
L_{20} \in &\ \{3_x, 5_x, 6_x, B_x, D_x, E_x\} \\
L_{23} \in &\ \{3_x, 6_x, A_x, B_x, C_x, D_x, F_x\}
\end{aligned}
$$

By using this set it is possible to obtain $2^{27} \cdot 3^4 \cdot 7^2$ pairs where the corresponding bitslices have the desired difference. Hence, we gather a total of $N = 2^{n+27} \cdot 3^4 \cdot 7^2$ plaintext pairs.

2. Request all ciphertexts and select only the ones where the ciphertext pairs having zero difference at bitslices b_0, b_2, b_6, b_{11}, b_{16}, b_{19}, b_{21}, b_{28}, b_{31}. Since there is a 36-bit filtering condition, $2^{n-9} \cdot 3^4 \cdot 7^2$ pairs remain.

3. Guess 24 bits of the subkey that correspond to the bitslices b_0, b_3, b_6, b_7, b_{20} and b_{23} before feeding the input to S_7. Partially encrypt every plaintext pair and eliminate all the pairs except the ones that have differences $b_0 = 0100$, $b_3 = 1110$, $b_6 = 1010$, $b_7 = 0001$, $b_{20} = 0010$, $b_{23} = 1000$. Hence, 2^{n-13} pairs remain.

4. Guess 88 bits of the subkey that correspond to the bitslices b_1, b_{3-5}, b_{7-10}, b_{12-15}, b_{17-18}, b_{20}, b_{22-27}, b_{29} before feeding the output to S_5. Partially decrypt every ciphertext pair and increase the counter of the subkey when the pairs have the difference shown in Table 7.2. There are 64-bit filtering conditions and hence 2^{n-77} pairs remain.

The attack is on 112 bits of the subkey and we expect $p_{fa} \cdot 2^{112}$ many subkeys to get hits less than or equivalent to threshold T. The probability of satisfying the improbable differential for a wrong subkey is

$$p = 2^{-104} \cdot 3^{-4} \cdot 7^{-2} \approx 2^{-117.95}$$

and we have $p' = 2^{-4}$. To capture exactly the 112 bits of the subkey with a success probability of 99.9%, we select $p_{nd} = 0.001$ and $p_{fa} = 2^{-113}$ as the input of the Algorithm 4.1 and we obtain $N = 2^{131.80}$ and $T = 55914 \leq 2^{16}$. Thus, the data complexity of the attack is

$$2^{131.80} \cdot 2^{-3} \cdot 3^{-4} \cdot 7^{-2} \approx 2^{116.85}$$

chosen plaintexts and memory complexity is about 2^{112} 16-bit counters. During the subkey guess steps, we can guess 4 bits at a time and eliminate wrong pairs by checking the result of the corresponding S-box operations. Thus, the time complexity of Step 2 is

$$6 \cdot \frac{1}{7}\frac{1}{16} \cdot 2 \cdot 2^4 \cdot 2^{95.79} \approx 2^{96.57}$$

7-round SERPENT encryptions. Similarly, time complexity of Step 3 is $2^{117.57}$ 7-round SERPENT encryptions.

Table 7.2: A 7-Round Improbable Differential Attack

Input	X_0: 0000	0000	?00?	0000	0000	0000	??00	?00?
	X_1: 0000	0000	?00?	0000	0000	0000	??00	?00?
	X_2: 0000	0000	?00?	0000	0000	0000	??00	?00?
	X_3: 0000	0000	?00?	0000	0000	0000	??00	?00?
S_7	X_0: 0000	0000	0000	0000	0000	0000	1000	0000
	X_1: 0000	0000	0001	0000	0000	0000	0100	1000
	X_2: 0000	0000	0000	0000	0000	0000	0000	1001
	X_3: 0000	0000	1000	0000	0000	0000	0100	1000
LT	X_0: 0000	0010	0000	0000	0000	0000	0000	0000
	X_1: 0000	0000	0000	0000	0000	0000	0000	0000
	X_2: 0001	0010	0000	0000	0000	0000	0000	0000
	X_3: 0000	0000	0000	0000	0000	0000	0000	0000
S_0	X_0: 0000	0000	0000	0000	0000	0000	0000	0000
	X_1: 0001	0000	0000	0000	0000	0000	0000	0000
	X_2: 0000	0010	0000	0000	0000	0000	0000	0000
	X_3: 0001	0000	0000	0000	0000	0000	0000	0000
	4.5-Round Impossible Differential							
LT	X_0: 0??0	0?00	0000	0000	00?0	0??0	?000	?000
	X_1: 0?00	00?0	0000	0000	0000	0000	0000	0000
	X_2: 0??0	??00	??0?	0??0	??0?	0000	0000	00?0
	X_3: 0?00	000?	0000	0000	0000	000?	00??	0000
S_5	X_0: 0??0	????	??0?	0??0	????	0???	?0??	?0?0
	X_1: 0??0	????	??0?	0??0	????	0???	?0??	?0?0
	X_2: 0??0	????	??0?	0??0	????	0???	?0??	?0?0
	X_3: 0??0	????	??0?	0??0	????	0???	?0??	?0?0

$\downarrow$ $p' = 2^{-4}$

7.2 Improved Differential-Linear Attacks on SERPENT

In [11] a differential-linear attack on 11-round SERPENT-192 and SERPENT-256 is presented. The attack combines the 3-round differential

$$\Delta : 00000000000000000000000040050000 \rightarrow 0??00?000?00000000?00?0??0??0?0$$

that has a probability of $p = 2^{-7}$ with the 6-round linear approximation

$$\Lambda : 20060040000001001000000000000000 \rightarrow 00001000000000005000010000100001$$

of [7] that has bias $q = 2^{-27}$.

The first attack on 10-round SERPENT-128 is also presented in [11] which is obtained by removing the last round of this linear approximation. The data and time complexities of these attacks are reduced in [35] by using the following improvements:

1. Better analysis of the bias of the differential-linear approximation,

2. Better analysis of the success probability,

3. Changing the output mask.

Moreover in [35], these reduced complexities are used to extend the 11-round attack and obtain the first 12-round attack on SERPENT-256. In this section we further improve these differential-linear attacks by using the differential factors of SERPENT's S-boxes S_0 and S_1.

7.2.1 Differential Factors of SERPENT

Table 7.3: Differential Factors of SERPENT's S-boxes

S-box	Differential Factor	Output Difference
S_0	4_x	4_x
S_0	D_x	F_x
S_1	4_x	4_x
S_1	F_x	E_x
S_2	2_x	1_x
S_2	4_x	D_x
S_6	6_x	2_x
S_6	F_x	F_x

The differential-linear attacks of [11, 35] start at round 1 and the 3-round differential activates 5 S-boxes in this round. Two of the output differences of these activated S-boxes are 4_x and E_x which have differential factors as shown in Table 3.3. The authors guess every possible 20 subkey bits corresponding to these five S-boxes but the attacker can only obtain 18-bit advantage for this subkey due to Theorem 3.3 and there is no need to try half of the subkeys corresponding to these two S-boxes having differential factors. Thus, the advantage of the differential-linear attacks on 10, 11, and 12 rounds of SERPENT are actually 38, 46, and 158 bits instead of 40, 48, and 160 bits, respectively. And again by Theorem 3.3, the same attacks can be performed with time complexities reduced by a factor of 4.

Moreover, the 12-round attack of [35] adds one more round to the top of the differential which affects every S-box at round 0 except the S-boxes 2, 3, 19, and 23 and guesses the 112 bits of the subkey corresponding to these active S-boxes. However, by using the undisturbed bits of SERPENT, we observed that the output difference of the S-box 8 is exactly 4_x. Since $\mu = 4_x$ also has a differential factor for S_0, the attacker's advantage reduces to 157 bits and the time complexity of the attack further reduces by a factor of 2. Table 7.4 summarizes this 12-round attack and highlights the differential factors and the undisturbed bits that are used to reduce the time complexity.

We also observed that by replacing the 3-round differential with a more probable one,

Table 7.4: 12-round differential-linear attack of [35]. Output differences μ that contain differential factors, which are 4_x and E_x for S_1 and 4_x for S_0, are shown in bold. Undisturbed bits are shown in italic.

Input	X_0:	????	????	0???	0???	????	????	????	00??
	X_1:	????	????	0???	0???	????	????	????	00??
	X_2:	????	????	0???	0???	????	???*1*	????	00??
	X_3:	????	????	0???	0???	????	????	????	00??
S_0	X_0:	??0?	00?0	0000	0?00	00?0	000**0**	00??	00??
	X_1:	??0?	????	00?0	0???	0???	???**0**	0?00	0000
	X_2:	000?	00??	0??0	0?00	??00	?00**1**	0?00	0000
	X_3:	?0??	?0??	00??	0???	??0?	0??**0**	?001	0000
LT	X_0:	?000	0000	0000	0??0	0?00	?000	0000	0000
	X_1:	?000	0000	0000	0??0	0?00	?000	0000	0000
	X_2:	?000	0000	0000	0??0	0?00	?000	0000	0000
	X_3:	?000	0000	0000	0*1*?0	0?00	*1*000	0000	0000
S_1	X_0:	0000	0000	0000	0100	0**0**00	**0**000	0000	0000
	X_1:	1000	0000	0000	0010	0**1**00	**0**000	0000	0000
	X_2:	0000	0000	0000	0000	0**1**00	**1**000	0000	0000
	X_3:	0000	0000	0000	0010	0**1**00	**0**000	0000	0000
LT	X_0:	0000	0000	0000	0000	0000	0000	0001	0000
	X_1:	0000	0000	0000	0000	0000	0000	0000	0000
	X_2:	0000	0000	0000	0000	0000	0000	1001	0000
	X_3:	0000	0000	0000	0000	0000	0000	0000	0000
9-Round Differential-Linear Characteristic $\Delta \circ \Lambda$									
Last Round									

we can perform these attacks with less data complexity and capture four more subkey bits with a time complexity increased by a factor of $2^{3.35}$. These modified attacks are provided in the following Section 7.2.2.

7.2.2 3-Round Differentials with Higher Probability

The rounds of the 3-round differential used in the differential-linear attacks of [11, 35] have probabilities 2^{-5}, 2^{-1}, and 1 but the authors observed experimentally that this differential has probability 2^{-7} instead of 2^{-6}. We observed that there are 3-round differentials of SERPENT with probability 2^{-5} that can be combined with the same linear approximations. The rounds of these differential have probabilities 2^{-5}, 1, and 1 and for this reason, the theoretical and practical probabilities of these differentials are the same. These differentials activate six S-boxes at the first round of the attack instead of five. So replacing the original differential with these result in capturing 4

more subkey bits but time complexity of the attacks also increase by a factor of 2^4.

Since the data complexity of a differential-linear attack is of $O(p^{-2}q^{-4})$ and replacing the differential result in $p = 2^{-5}$ instead of 2^{-7}, one would expect the modified attacks to have data and time complexities reduced by a factor of 2^4. However, experiment results show that the gain in the modified attacks is at most a factor of $(2^{-0.32})^2$. This is because the transition between the original differential and the linear approximation is far better than expected. For instance, when the original 3-round differential is combined with a 1-round linear approximation of bias 2^{-5}, Dunkelman *et al.* experimentally verified that the 4-round differential-linear path has bias $2^{-13.75}$, instead of $2 \cdot 2^{-7} \cdot (2^{-5})^2 = 2^{-16}$. We performed similar experiments on five different 3-round differential with probability 2^{-5} using 2^{34} pairs and the results are summarized in Table 7.5.

Table 7.5: 4-Round Biases for 3-Round Differentials with Probability 2^{-5} and 1-round Linear Approximation with Bias 2^{-5}

#	Input Difference $X_0\ X_1\ X_2\ X_3$ (in Hexadecimal)	#Active S-boxes	Bias	Standard Deviation
1	40000000 00000000 40000002 00000000	6	$2^{-13,49}$	$2^{-18.03}$
2	00000000 40000000 40000002 00000000	6	$2^{-13,43}$	$2^{-18.11}$
3	00000000 40000000 00000002 40000000	6	$2^{-13,56}$	$2^{-18.07}$
4	00000000 40000000 40000002 00000002	6	$2^{-13,43}$	$2^{-18.19}$
5	00000002 00000000 00000012 00000000	6	$2^{-14,65}$	$2^{-18.00}$

We replace the original differential with the second one from Table 7.5 and obtain new 10, and 11 round differential-linear attacks. This change provides a 4-round bias of $2^{-13,43}$ instead of $2^{-13.75}$. Thus the data and time complexity gain in the modified attack is a factor of $(2^{-0.32})^2$. This differential activates six S-boxes instead of five so we capture four more subkey bits and the time complexity is multiplied by 2^4. One of the output differences of these activated S-boxes is again 4_x and thus we have one differential factor. Since the rest of our improved attacks are almost identical to the attacks of [35], we refer the interested reader to [35]. We summarize the complexities of the attacks on SERPENT in Table 7.6.

Table 7.6: Summary of attacks on SERPENT. Note that it is claimed in [54] that the multidimensional linear attacks of [56] may not work as claimed depending on the linear hull effect. If the claims are correct, then our use of differential factors in the attacks of [35] becomes the best attacks for this cipher.

#Rounds	Attack Type	Key Size	Data	Time	Memory	Advantage	Success	Reference
6	Meet-in-the-middle	256	512 KP	2^{247} En	2^{246} B	-	-	[45]
6	Differential	All	2^{83} CP	2^{90} En	2^{40} B	-	-	[45]
6	Differential	All	2^{71} CP	2^{103} En	2^{75} B	-	-	[45]
6	Differential	192, 256	2^{41} CP	2^{163} En	2^{45} B	124	-	[45]
7	Differential	256	2^{122} CP	2^{248} En	2^{126} B	128	-	[45]
7	**Improbable**	**All**	$\mathbf{2^{116.85}}$ **CP**	$\mathbf{2^{117.57}}$ **En**	$\mathbf{2^{113}}$ **B**	**112**	**99.9%**	**Sect. 7.1**
7	Differential	All	2^{84} CP	2^{85} MA	2^{56} B	-	-	[8]
10	Rectangle	192, 256	$2^{126.3}$ CP	$2^{173.8}$ MA	$2^{131.8}$ B	80	-	[10]
10	Boomerang	192, 256	$2^{126.3}$ AC	$2^{173.8}$ MA	2^{89} B	80	-	[10]
10	Differential-Linear	All	$2^{101.2}$ CP	$2^{115.2}$ En	2^{40} B	40	84%	[35]
10	**Differential-Linear**	**All**	$\mathbf{2^{101.2}}$ **CP**	$\mathbf{2^{113.2}}$ **En**	$\mathbf{2^{40}}$ **B**	**38**	**84%**	**Sect. 7.2.1**
10	**Differential-Linear**	**All**	$\mathbf{2^{100.55}}$ **CP**	$\mathbf{2^{116.55}}$ **En**	$\mathbf{2^{40}}$ **B**	**42**	**84%**	**Sect. 7.2.2**
11	Linear	256	2^{118} KP	2^{214} MA	2^{85} B	140	78.5%	[7]
11	Multidimensional Linear [1]	All	2^{116} KP	$2^{107.5}$ En	2^{108} B	48	78.5%	[56]
11	Multidimensional Linear [2]	All	2^{118} KP	$2^{109.5}$ En	2^{104} B	44	78.5%	[56]
11	Nonlinear	192, 256	$2^{120.36}$ KP	$2^{139.63}$ MA	$2^{133.17}$ B	118	78.5%	[54]
11	Filtered Nonlinear	192, 256	$2^{114.55}$ KP	$2^{155.76}$ MA	$2^{146.59}$ B	132	78.5%	[54]
11	Differential-Linear	192, 256	$2^{121.8}$ CP	$2^{135.7}$ MA	2^{76} B	48	84%	[35]
11	**Differential-Linear**	**192, 256**	$\mathbf{2^{121.8}}$ **CP**	$\mathbf{2^{133.7}}$ **MA**	$\mathbf{2^{76}}$ **B**	**46**	**84%**	**Sect. 7.2.1**
11	**Differential-Linear**	**192, 256**	$\mathbf{2^{121.15}}$ **CP**	$\mathbf{2^{137.05}}$ **MA**	$\mathbf{2^{76}}$ **B**	**50**	**84%**	**Sect. 7.2.2**
12	Multidimensional Linear [3]	256	2^{116} KP	$2^{237.5}$ En	2^{125} B	174	78.5%	[56]
12	Differential-Linear	256	$2^{123.5}$ CP	$2^{249.4}$ En	$2^{128.5}$ B	160	84%	[35]
12	**Differential-Linear**	**256**	$\mathbf{2^{123.5}}$ **CP**	$\mathbf{2^{246.4}}$ **En**	$\mathbf{2^{128.5}}$ **B**	**157**	**84%**	**Sect. 7.2.1**

[1] In [54], it is claimed that the correct data complexity of this attack is $2^{125.81}$ KP and the time complexity is $2^{101.44}$ En $+2^{114.13}$ MA.

[2] In [54], it is claimed that the correct data complexity of this attack is $2^{127.78}$ KP and the time complexity is $2^{97.41}$ En $+2^{110.10}$ MA.

[3] In [54], it is claimed that the correct data complexity of this attack is $\geq 2^{125.81}$ KP and the time complexity is $2^{229.44}$ En $+2^{242.13}$ MA.

Table 7.7: 11-Round differential-linear attack with a 3-round differential of probability 2^{-5}. Output differences $\mu = 4_x$ and $\mu = E_x$ that contain differential factors for S_1 are shown in bold. Undisturbed bits are shown in italic.

Input	X_0:	0??0	0000	0000	00?0	0000	?00?	00?0	0000
	X_1:	0??0	0000	0000	00?0	0000	?00?	00?0	0000
	X_2:	0??0	0000	0000	00?0	0000	?00?	00?0	0000
	X_3:	0??0	0000	0000	00*1*0	0000	?00?	00*1*0	0000
S_1	X_0:	0000	0000	0000	0010	0000	000**0**	00**0**0	0000
	X_1:	0110	0000	0000	0000	0000	100**1**	00**0**0	0000
	X_2:	0000	0000	0000	0000	0000	000**1**	00**1**0	0000
	X_3:	0000	0000	0000	0000	0000	100**1**	00**0**0	0000
LT	X_0:	0000	0000	0000	0000	0000	0000	0000	0000
	X_1:	0100	0000	0000	0000	0000	0000	0000	0000
	X_2:	0100	0000	0000	0000	0000	0000	0000	0010
	X_3:	0000	0000	0000	0000	0000	0000	0000	0000
S_2	X_0:	0000	0000	0000	0000	0000	0000	0000	0000
	X_1:	0000	0000	0000	0000	0000	0000	0000	0010
	X_2:	0100	0000	0000	0000	0000	0000	0000	0000
	X_3:	0000	0000	0000	0000	0000	0000	0000	0010
LT	X_0:	0000	0000	0000	0000	0000	0000	0000	0000
	X_1:	0000	0000	0000	0000	0000	0000	0000	0000
	X_2:	0000	0000	1000	0000	0000	0000	0000	0000
	X_3:	0000	0000	0000	0000	0000	0000	0000	0000
S_3	X_0:	0000	0000	?000	0000	0000	0000	0000	0000
	X_1:	0000	0000	?000	0000	0000	0000	0000	0000
	X_2:	0000	0000	?000	0000	0000	0000	0000	0000
	X_3:	0000	0000	?000	0000	0000	0000	0000	0000
LT	X_0:	00?0	0000	0000	?000	0000	0??0	0?00	?00?
	X_1:	0000	?00?	0000	0000	0000	0000	00?0	0000
	X_2:	0000	0000	?0??	000?	0000	0000	000?	0?00
	X_3:	0?00	0000	0000	0000	0?00	0000	0000	00?0
S_4	X_0:	0??0	?00?	?0??	?00?	0?00	0??0	0???	????
	X_1:	0??0	?00?	?0??	?00?	0?00	0??0	0???	????
	X_2:	0??0	?00?	?0??	?00?	0?00	0??0	0???	????
	X_3:	0??0	?00?	?0??	?00?	0?00	0??0	0???	????
6-Round Linear Approximation Λ									
Last Round									

$p = 2^{-5}$

REFERENCES

[1] 3rd Generation Partnership Project, Specification of the 3GPP Confidentiality and Integrity Algorithms - Document 2: KASUMI Specification (Release 11), Technical Report 3GPP TS 35.202 V11.0.0 (2012-09), September 2012.

[2] R. Arratia and L. Gordon, Tutorial on large deviations for the binomial distribution, in *Bulletin of Mathematical Biology 51*, pp. 125–131, 1989.

[3] E. Biham, R. J. Anderson, and L. R. Knudsen, Serpent: A new block cipher proposal, in S. Vaudenay, editor, *FSE*, volume 1372 of *Lecture Notes in Computer Science*, pp. 222–238, Springer, 1998, ISBN 3-540-64265-X.

[4] E. Biham, A. Biryukov, and A. Shamir, Impossible differential attacks, in *Rump Session of CRYPTO 1998*.

[5] E. Biham, A. Biryukov, and A. Shamir, Miss in the middle attacks on IDEA and Khufu, in Knudsen [42], pp. 124–138.

[6] E. Biham, A. Biryukov, and A. Shamir, Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials, J. Cryptology, 18(4), pp. 291–311, 2005.

[7] E. Biham, O. Dunkelman, and N. Keller, Linear cryptanalysis of reduced round Serpent, in M. Matsui, editor, *FSE*, volume 2355 of *Lecture Notes in Computer Science*, pp. 16–27, Springer, 2001, ISBN 3-540-43869-6.

[8] E. Biham, O. Dunkelman, and N. Keller, The rectangle attack - rectangling the Serpent, in B. Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pp. 340–357, Springer, 2001, ISBN 3-540-42070-3.

[9] E. Biham, O. Dunkelman, and N. Keller, Enhancing differential-linear cryptanalysis, in Zheng [89], pp. 254–266.

[10] E. Biham, O. Dunkelman, and N. Keller, New results on boomerang and rectangle attacks, in J. Daemen and V. Rijmen, editors, *FSE*, volume 2365 of *Lecture Notes in Computer Science*, pp. 1–16, Springer, 2002, ISBN 3-540-44009-7.

[11] E. Biham, O. Dunkelman, and N. Keller, Differential-linear cryptanalysis of Serpent, in T. Johansson, editor, *FSE*, volume 2887 of *Lecture Notes in Computer Science*, pp. 9–21, Springer, 2003, ISBN 3-540-20449-0.

[12] E. Biham and A. Shamir, Differential Cryptanalysis of DES-like Cryptosystems, J. Cryptology, 4(1), pp. 3–72, 1991.

[13] B. Bilgin, A. Bogdanov, M. Knezevic, F. Mendel, and Q. Wang, Fides: Lightweight authenticated cipher with side-channel resistance for constrained

hardware, in G. Bertoni and J.-S. Coron, editors, *CHES*, volume 8086 of *Lecture Notes in Computer Science*, pp. 142–158, Springer, 2013, ISBN 978-3-642-40348-4.

[14] B. Bilgin, S. Nikova, V. Nikov, V. Rijmen, and G. Stütz, Threshold implementations of all 3x3 and 4x4 S-boxes, in E. Prouff and P. Schaumont, editors, *CHES*, volume 7428 of *Lecture Notes in Computer Science*, pp. 76–91, Springer, 2012, ISBN 978-3-642-33026-1.

[15] A. Biryukov, editor, *Fast Software Encryption, 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007, Revised Selected Papers*, volume 4593 of *Lecture Notes in Computer Science*, Springer, 2007, ISBN 978-3-540-74617-1.

[16] C. Blondeau, Private communication (2009).

[17] C. Blondeau, Improbable differential from impossible differential: On the validity of the model, in G. Paul and S. Vaudenay, editors, *INDOCRYPT*, volume 8250 of *Lecture Notes in Computer Science*, pp. 149–160, Springer, 2013, ISBN 978-3-319-03514-7.

[18] C. Blondeau and B. Gérard, On the data complexity of statistical attacks against block ciphers, in A. Kholosha, E. Rosnes, and M.Parker, editors, *Workshop on Coding and Cryptography - WCC 2009*, pp. 469–488, Ullensvang, Norway, May 2009.

[19] C. Blondeau and B. Gérard, Multiple differential cryptanalysis: Theory and practice, in A. Joux, editor, *FSE*, volume 6733 of *Lecture Notes in Computer Science*, pp. 35–54, Springer, 2011, ISBN 978-3-642-21701-2.

[20] C. Blondeau, B. Gérard, and J.-P. Tillich, Accurate estimates of the data complexity and success probability for various cryptanalyses, Des. Codes Cryptography, 59(1-3), pp. 3–34, 2011.

[21] A. Bogdanov, H. Geng, M. Wang, L. Wen, and B. Collard, Zero-correlation linear cryptanalysis with FFT and improved attacks on ISO standards Camellia and Clefia, in T. Lange, K. Lauter, and P. Lisonek, editors, *Selected Areas in Cryptography*, volume 8282 of *Lecture Notes in Computer Science*, pp. 306–323, Springer, 2013, ISBN 978-3-662-43413-0.

[22] A. Bogdanov, M. Knezevic, G. Leander, D. Toz, K. Varici, and I. Verbauwhede, spongent: A lightweight hash function, in Preneel and Takagi [62], pp. 312–325.

[23] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, PRESENT: An ultra-lightweight block cipher, in P. Paillier and I. Verbauwhede, editors, *CHES*, volume 4727 of *Lecture Notes in Computer Science*, pp. 450–466, Springer, 2007, ISBN 978-3-540-74734-5.

[24] C. D. Canniere, H. Sato, and D. Watanabe, Hash function Luffa: Specification, Submission to NIST (Round 2), 2009.

[25] D. Chaum and J.-H. Evertse, Crytanalysis of DES with a reduced number of rounds: Sequences of linear factors in block ciphers, in H. C. Williams, editor, *CRYPTO*, volume 218 of *Lecture Notes in Computer Science*, pp. 192–211, Springer, 1985, ISBN 3-540-16463-4.

[26] J. Y. Cho, Linear cryptanalysis of reduced-round PRESENT, in J. Pieprzyk, editor, *CT-RSA*, volume 5985 of *Lecture Notes in Computer Science*, pp. 302–317, Springer, 2010, ISBN 978-3-642-11924-8.

[27] B. Collard and F.-X. Standaert, A statistical saturation attack against the block cipher PRESENT, in M. Fischlin, editor, *CT-RSA*, volume 5473 of *Lecture Notes in Computer Science*, pp. 195–210, Springer, 2009, ISBN 978-3-642-00861-0.

[28] N. Courtois and J. Pieprzyk, Cryptanalysis of block ciphers with overdefined systems of equations, in Zheng [89], pp. 267–287.

[29] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley series in communications. Wiley, 1991, ISBN 0471062596.

[30] J. Daemen, M. Peeters, and G. V. Assche, Bitslice ciphers and power analysis attacks, in B. Schneier, editor, *FSE*, volume 1978 of *Lecture Notes in Computer Science*, pp. 134–149, Springer, 2000, ISBN 3-540-41728-1.

[31] J. Daemen, M. Peeters, G. V. Assche, and V. Rijmen, Nessie proposal: NOEKEON. NESSIE proposal, 27 October 2000.

[32] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*, Information Security and Cryptography, Springer, 2002, ISBN 3-540-42580-2, 978-3-642-07646-6.

[33] Des, Data encryption standard, in *FIPS PUB 46, Federal Information Processing Standards Publication*, pp. 46–2, 1977.

[34] I. Dinur and A. Shamir, Cube attacks on tweakable black box polynomials, in A. Joux, editor, *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pp. 278–299, Springer, 2009, ISBN 978-3-642-01000-2.

[35] O. Dunkelman, S. Indesteege, and N. Keller, A differential-linear attack on 12-round Serpent, in D. R. Chowdhury, V. Rijmen, and A. Das, editors, *INDOCRYPT*, volume 5365 of *Lecture Notes in Computer Science*, pp. 308–321, Springer, 2008, ISBN 978-3-540-89753-8.

[36] D. W. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith, Hummingbird: Ultra-lightweight cryptography for resource-constrained devices, in R. Sion, R. Curtmola, S. Dietrich, A. Kiayias, J. M. Miret, K. Sako, and F. Sebé, editors, *Financial Cryptography Workshops*, volume 6054 of *Lecture Notes in Computer Science*, pp. 3–18, Springer, 2010, ISBN 978-3-642-14991-7.

[37] D. W. Engels, M.-J. O. Saarinen, P. Schweitzer, and E. M. Smith, The Hummingbird-2 lightweight authenticated encryption algorithm, in A. Juels and C. Paar, editors, *RFIDSec*, volume 7055 of *Lecture Notes in Computer Science*, pp. 19–31, Springer, 2011, ISBN 978-3-642-25285-3.

[38] J. Guo, T. Peyrin, A. Poschmann, and M. J. B. Robshaw, The led block cipher, in Preneel and Takagi [62], pp. 326–341.

[39] ISO/IEC 29192-2:2012, Information technology - security techniques - lightweight cryptography - part 2: Block ciphers, 2011.

[40] L. Knudsen, Deal - a 128-bit block cipher, in *NIST AES Proposal*, 1998.

[41] L. R. Knudsen, Truncated and higher order differentials, in Preneel [61], pp. 196–211.

[42] L. R. Knudsen, editor, *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, volume 1636 of *Lecture Notes in Computer Science*, Springer, 1999, ISBN 3-540-66226-X.

[43] L. R. Knudsen, G. Leander, A. Poschmann, and M. J. B. Robshaw, Printcipher: A block cipher for IC-printing, in S. Mangard and F.-X. Standaert, editors, *CHES*, volume 6225 of *Lecture Notes in Computer Science*, pp. 16–32, Springer, 2010, ISBN 978-3-642-15030-2.

[44] P. C. Kocher, J. Jaffe, and B. Jun, Differential power analysis, in M. J. Wiener, editor, *CRYPTO*, volume 1666 of *Lecture Notes in Computer Science*, pp. 388–397, Springer, 1999, ISBN 3-540-66347-9.

[45] T. Kohno, J. Kelsey, and B. Schneier, Preliminary cryptanalysis of reduced-round Serpent, in *AES Candidate Conference*, pp. 195–211, 2000.

[46] X. Lai and J. L. Massey, A proposal for a new block encryption standard, in I. Damgård, editor, *EUROCRYPT*, volume 473 of *Lecture Notes in Computer Science*, pp. 389–404, Springer, 1990, ISBN 3-540-53587-X.

[47] S. K. Langford and M. E. Hellman, Differential-linear cryptanalysis, in Y. Desmedt, editor, *CRYPTO*, volume 839 of *Lecture Notes in Computer Science*, pp. 17–25, Springer, 1994, ISBN 3-540-58333-5.

[48] C. H. Lim, Crypton: A new 128-bit block cipher - specification and analysis, 1998.

[49] C. H. Lim, A revised version of crypton - crypton v1.0, in Knudsen [42], pp. 31–45.

[50] J. Lopez and G. Tsudik, editors, *Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings*, volume 6715 of *Lecture Notes in Computer Science*, 2011, ISBN 978-3-642-21553-7.

[51] H. Mala, M. Dakhilalian, and M. Shakiba, Impossible differential attacks on 13-round CLEFIA-128, J. Comput. Sci. Technol., 26(4), pp. 744–750, 2011.

[52] M. Matsui, Linear cryptoanalysis method for DES cipher, in T. Helleseth, editor, *EUROCRYPT*, volume 765 of *Lecture Notes in Computer Science*, pp. 386–397, Springer, 1993, ISBN 3-540-57600-2.

[53] M. Matsui, New block encryption algorithm MISTY, in E. Biham, editor, *FSE*, volume 1267 of *Lecture Notes in Computer Science*, pp. 54–68, Springer, 1997, ISBN 3-540-63247-6.

[54] J. McLaughlin and J. A. Clark, Filtered nonlinear cryptanalysis of reduced-round Serpent, and the wrong-key randomization hypothesis, in M. Stam, editor, *IMA Int. Conf.*, volume 8308 of *Lecture Notes in Computer Science*, pp. 120–140, Springer, 2013, ISBN 978-3-642-45238-3.

[55] R. C. Merkle, Fast software encryption functions, in A. Menezes and S. A. Vanstone, editors, *CRYPTO*, volume 537 of *Lecture Notes in Computer Science*, pp. 476–501, Springer, 1990, ISBN 3-540-54508-5.

[56] P. H. Nguyen, H. Wu, and H. Wang, Improving the algorithm 2 in multidimensional linear cryptanalysis, in U. Parampalli and P. Hawkes, editors, *ACISP*, volume 6812 of *Lecture Notes in Computer Science*, pp. 61–74, Springer, 2011, ISBN 978-3-642-22496-6.

[57] NIST, SKIPJACK and KEA Algorithm Specifications Version 2.0, Technical report, National Institute of Standards and Technology (NIST), May 1998.

[58] K. Nyberg and L. R. Knudsen, Provable security against differential cryptanalysis, in E. F. Brickell, editor, *CRYPTO*, volume 740 of *Lecture Notes in Computer Science*, pp. 566–574, Springer, 1992, ISBN 3-540-57340-2.

[59] N. B. of Standards, Data Encryption Standard. FIPS PUB 46. National Bureau of Standards, U.S. Department of Commerce, Washington D.C.,, 15 January 1977.

[60] K. Ohkuma, Weak keys of reduced-round PRESENT for linear cryptanalysis, in M. J. J. Jr., V. Rijmen, and R. Safavi-Naini, editors, *Selected Areas in Cryptography*, volume 5867 of *Lecture Notes in Computer Science*, pp. 249–265, Springer, 2009, ISBN 978-3-642-05443-3.

[61] B. Preneel, editor, *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, volume 1008 of *Lecture Notes in Computer Science*, Springer, 1995.

[62] B. Preneel and T. Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, Springer, 2011, ISBN 978-3-642-23950-2.

[63] M.-J. O. Saarinen, Cryptographic analysis of all $4\times$ 4 s-boxes, in A. Miri and S. Vaudenay, editors, *Selected Areas in Cryptography*, volume 7118 of *Lecture Notes in Computer Science*, pp. 118–133, Springer, 2011, ISBN 978-3-642-28495-3.

[64] B. Schneier and J. Kelsey, Unbalanced feistel networks and block cipher design, in D. Gollmann, editor, *FSE*, volume 1039 of *Lecture Notes in Computer Science*, pp. 121–144, Springer, 1996, ISBN 3-540-60865-6.

[65] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, Twofish: A 128-bit block cipher, in *First Advanced Encryption Standard (AES) Conference*, 1998.

[66] A. A. Selçuk, On probability of success in linear and differential cryptanalysis, J. Cryptology, 21(1), pp. 131–147, 2008.

[67] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, Piccolo: An ultra-lightweight blockcipher, in Preneel and Takagi [62], pp. 342–357.

[68] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, The 128-bit blockcipher CLEFIA (extended abstract), in Biryukov [15], pp. 181–195.

[69] Sony Corporation, The 128-bit Blockcipher CLEFIA, Security and Performance Evaluations, Revision 1.0, June 1 (2007), `http://www.sony.net/Products/cryptography/clefia/`.

[70] A. Sorkin, LUCIFER, a cryptographic algorithm, j-CRYPTOLOGIA, 8(1), pp. 22–42, January 1984, ISSN 0161-1194 (print), 1558-1586 (electronic), see also erratum, Cryptologia **7**, 1978, p. 118.

[71] F.-X. Standaert, G. Piret, N. Gershenfeld, and J.-J. Quisquater, Sea: A scalable encryption algorithm for small embedded applications, in J. Domingo-Ferrer, J. Posegga, and D. Schreckling, editors, *CARDIS*, volume 3928 of *Lecture Notes in Computer Science*, pp. 222–236, Springer, 2006, ISBN 3-540-33311-8.

[72] S. Sun, L. Hu, and P. Wang, Automatic security evaluation for bit-oriented block ciphers in related-key model: Application to present-80, lblock and others, IACR Cryptology ePrint Archive, 2013, p. 676, 2013.

[73] X. Tang, B. Sun, R. Li, and C. Li, Impossible differential cryptanalysis of 13-round CLEFIA-128, Journal of Systems and Software, 84(7), pp. 1191–1196, 2011.

[74] C. Tezcan, The improbable differential attack: Cryptanalysis of reduced round CLEFIA, in G. Gong and K. C. Gupta, editors, *INDOCRYPT*, volume 6498 of *Lecture Notes in Computer Science*, pp. 197–209, Springer, 2010, ISBN 978-3-642-17400-1.

[75] C. Tezcan, Improbable differential attack on PRESENT using undisturbed bits, in Ömür Uğur, editor, *International Conference on Applied and Computational Mathematics, Book of Abstracts*, p. 85, 2012, ISBN 978-3-642-17400-1.

[76] C. Tezcan, Improbable differential cryptanalysis, in A. Elçi, M. S. Gaur, M. A. Orgun, and O. B. Makarevich, editors, *SIN*, p. 457, ACM, 2013, ISBN 978-1-4503-2498-4.

[77] C. Tezcan, Improbable differential attacks on PRESENT using undisturbed bits, Journal of Computational and Applied Mathematics, 259, Part B(0), pp. 503 – 511, 2014, ISSN 0377-0427, recent Advances in Applied and Computational Mathematics: ICACM-IAM-METU. On the occasion of 10th anniversary of the foundation of Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey.

[78] C. Tezcan and A. Temizel, Cryptanalysis of PRESENT via CUDA devices, in *2014 GPU Technology Conference (GTC 2014)*, 2014.

[79] C. Tezcan and S. Vaudenay, On hiding a plaintext length by preencryption, in Lopez and Tsudik [50], pp. 345–358.

[80] Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Saito, T. Suzaki, and H. Kubo, Impossible differential cryptanalysis of CLEFIA, in K. Nyberg, editor, *FSE*, volume 5086 of *Lecture Notes in Computer Science*, pp. 398–411, Springer, 2008, ISBN 978-3-540-71038-7.

[81] Y. Tsunoo, E. Tsujihara, M. Shigeri, T. Suzaki, and T. Kawabata, Cryptanalysis of CLEFIA using multiple impossible differentials, in *International Symposium on Information Theory and Its Applications - ISITA 2008.*, pp. 1–6, 7-10 2008.

[82] V. Dolmatov (Ed.), GOST 28147-89: Encryption, decryption, and message authentication code (MAC) algorithms, in *Internet Engineering Task Force RFC 5830*, March 2010.

[83] K. Varıcı, O. Özen, and Çelebi Kocair, Sarmal: Sha-3 proposal, Submission to NIST, 2008.

[84] M. Wang, Differential cryptanalysis of reduced-round PRESENT, in S. Vaudenay, editor, *AFRICACRYPT*, volume 5023 of *Lecture Notes in Computer Science*, pp. 40–49, Springer, 2008, ISBN 978-3-540-68159-5.

[85] W. Wu and L. Zhang, LBlock: A lightweight block cipher, in Lopez and Tsudik [50], pp. 327–344.

[86] Özgül Küçük, The hash function Hamsi, Submission to NIST (updated), 2009.

[87] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, Rectangle: A bit-slice ultra-lightweight block cipher suitable for multiple platforms, IACR Cryptology ePrint Archive, 2014, p. 84, 2014.

[88] W. Zhang and J. Han, Impossible differential analysis of reduced round CLEFIA, in M. Yung, P. Liu, and D. Lin, editors, *Inscrypt*, volume 5487 of *Lecture Notes in Computer Science*, pp. 181–191, Springer, 2008, ISBN 978-3-642-01439-0.

[89] Y. Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, Springer, 2002, ISBN 3-540-00171-9.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Tezcan, Cihangir
Nationality: Turkish
Date and Place of Birth: 18 March 1985, Ankara
Marital Status: Single
Phone: 0090 312 210 5016

EDUCATION

Degree	Institution	Year of Graduation
M.S.	METU, Department of Cryptography	2009
B.S.	METU, Department of Mathematics	2007
High School	Çankaya Milli Piyango Anadolu Lisesi	2003

PROFESSIONAL EXPERIENCE

Year	Place	Enrollment
2008-2010	METU, Turkey	Research Assistant at Dept. of Cryptography
2010-2011	EPFL, Switzerland	Teaching Assistant at Computer Sciences
2012-*Present*	METU, Turkey	Research Assistant at Dept. of Mathematics

PUBLICATIONS

International Journal Publications

1. **C. Tezcan**, Improbable Differential Attacks on PRESENT Using Undisturbed Bits, Journal of Computational and Applied Mathematics, 259, Part B(0), pp. 503 – 511, 2014, ISSN 0377-0427.

International Conference Publications

1. **C. Tezcan**, The improbable differential attack: Cryptanalysis of reduced round CLEFIA, in G. Gong and K. C. Gupta, editors, Progress in Cryptology - INDOCRYPT 2010 - 11th International Conference on Cryptology in Hyderabad, India, December 12-15, 2010. Proceedings, volume 6498 of Lecture Notes in Computer Science, Springer, 2010, ISBN 978-3-642-17400-1, pp. 197–209.

2. **C. Tezcan** and S. Vaudenay, On hiding a plaintext length by preencryption, in J. Lopez and G. Tsudik, editors, Applied Cryptography and Network Security - 9th International Conference, ACNS 2011, Nerja, Spain, June 7-10, 2011. Proceedings, volume 6715 of Lecture Notes in Computer Science, 2011, ISBN 978-3-642-21553-7., pp. 345–358.

3. O. Özen, K .Varıcı, **C. Tezcan** and Ç. Kocair, Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT. In Boy C., Nieto J. G. (eds.), ACISP 2009. LNCS, vol. 5594, pp. 90-107. Springer 2009.

4. **C. Tezcan**, Improbable Differential Attack on PRESENT using Undisturbed Bits. In International Conference on Applied and Computational Mathematics (ICACM 2012), Book of Abstracts, Ankara, TURKEY (3 October 2012).

5. **C. Tezcan**, Improbable differential cryptanalysis, in A. Elçi, M. S. Gaur, M. A. Orgun, and O. B. Makarevich, editors, SIN, p. 457, ACM, 2013, ISBN 978-1-4503-2498-4.

6. **C. Tezcan** and A. Temizel, Cryptanalysis of PRESENT via CUDA devices, in 2014 GPU Technology Conference (GTC 2014), 2014.