DIFFERENTIAL CRYPTANALYSIS ON LBLOCK USING DIFFERENTIAL FACTORS

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

BY

MERVE ÖĞÜNÇ

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

DECEMBER, 2018

Approval of the thesis:

DIFFERENTIAL CRYPTANALYSIS ON LBLOCK USING DIFFERENTIAL FACTORS

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

Prof. Dr. Ömür UĞUR
Director, Graduate School of **Applied Mathematics**

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

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

Dr. Cihangir Tezcan
Co-supervisor, **Mathematics, METU**

Examining Committee Members:

Prof. Dr. Ferruh Özbudak
Institute of Applied Mathematics, METU

Assoc. Prof. Dr. Ali Doğanaksoy
Department of Mathematics, METU

Assoc. Prof. Dr. Murat Cenk
Institute of Applied Mathematics, METU

Assoc. Prof. Dr. Fatih Sulak
Department of Mathematics, Atilim University

Assoc. Prof. Dr. Zülfükar Saygı
Department of 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: MERVE ÖĞÜNÇ

Signature :

ABSTRACT

DIFFERENTIAL CRYPTANALYSIS ON LBLOCK USING DIFFERENTIAL FACTORS

Öğünç, Merve
M.S., Department of Cryptography
Supervisor : Assoc. Prof. Dr. Ali Doğanaksoy
Co-Supervisor : Dr. Cihangir Tezcan

December, 2018, 71 pages

Cryptography had actually a long history and comes to today by evolving day by day. Now, it is a huge area in terms of the applications in industry and research topics in academia. Even if we do not realize, the cryptographic tools are placed in every single day of our life. To protect the information, the cryptographic algorithm is used in several areas from the basic website to smart devices. One of the classes of cryptographic algorithm is symmetric-key algorithms which cover block ciphers and stream ciphers. To evaluate the security of block ciphers, several cryptanalytic methods are used as a tool in cryptography. One of the most important methods is differential cryptanalysis. Since it is commonly used, cipher designers specify the cipher principles to be secure against differential attack. In differential cryptanalysis, attacker observes that the difference between chosen plaintexts how affects the difference between corresponding ciphertexts. After finding a relation between plaintext and ciphertext, an attacker tries to get round keys. With the recently introduced S-box property called Differential Factors, all of the attacked key bits may not be determined if the S-box has a differential factor property and that S-box is activated in the distinguisher.

With advances in technology, the usage of embedded systems has increased and the needs for new cryptographic instruments has emerged. Therefore, the subclasses of cryptography become diversified. One of the diversification is lightweight cryptography. Lightweight cryptography is based on optimizing the trade-off between security, cost, and performance. With increasing use of low resource devices such as RFID tags and sensor networking in different areas, the needs for lightweight cryptographic modules have started to increase. For this reason, lightweight cryptography has become prominent for the last few years. To fulfill the need, several lightweight block ciphers have been designed such as PRESENT, SEA, LED. In this work, we briefly present some lightweight block ciphers, their cryptanalysis and corrected cryptanalysis via differential factors.

LBLOCK, as one of these lightweight block ciphers, is a 32-round block cipher proposed at Applied Cryptography and Network Security Conference 2011 by Wenling Wu and Lei Zhang.

In this thesis, we study on the lightweight block cipher LBLOCK and observe the differential cryptanalysis to LBLOCK. Since the attackers do not consider the differential factors while performing the attack, the time complexity needs a correction. We correct the time complexity of the attack.

Keywords: Differential Cryptanalysis, Differential Factors, S-box, Block Ciphers, Lightweight Block Ciphers

ÖZ

LBLOCK ALGORİTMASININ DİFERANSİYEL KRİPTANALİZİNİN DİFERANSİYEL FAKTÖRLER KULLANILARAK YENIDEN GERÇEKLEŞTİRİLMESİ

Öğünç, Merve
Yüksek Lisans, Kriptografi Bölümü
Tez Yöneticisi : Yrd. Doç. Dr. Ali Doğanaksoy
Ortak Tez Yöneticisi : Dr. Cihangir Tezcan

Aralık, 2018, 71 sayfa

Uzun bir tarihe sahip olan kriptografi, hergün dahada gelişerek günümüze kadar gelmiştir. Günümüzde ise hem endüstriyel uygulamaları hem de akademik çalışmaları kapsayan büyük bir alana yayılmıştır. Bizler kullandığımızı farketmesek bile kriptografik araçlar günlük hayatımızda önemli bir yere sahiptir. Bilgilerin saklanmasında basit bir internet sitesinden akıllı cihazların kullanımına kadar birçok alanda kriptografik algoritmalar kullanılmaktadır. Blok şifreleri ve akan şifreleri kapsayan simetrik anahtar algoritmaları, kriptografik algoritmanın sınıflarından biridir. Blok şifrelerin güvenli olup olmadığını değerlendirebilmek için birçok çeşitli kriptanalik metod kullanılır. Bu metodların en önemlilerinden biride diferansiyel kriptanalizdir. Bu atağın blok şifrelere sıklıkla uygulanmasından dolayı, şifrelerin tasarım sürecinde bu atağa karşı güçlü olmasını sağlayacak özellikler seçilmeye çalışılır. Diferansiyel kriptanalizde saldırgan seçilmiş şifresiz metine ufak bir değişiklik yapıldığında şifreli metin üzerindeki değişikliği inceler. Şifresiz ve şifreli metin arasındaki bağlantıyı bulduktan sonrada turlarda kullanan anahtarların bitlerini ele geçirmeye çalışır. Eğer algoritmanın S-kutusu yeni bulunan bir S-kutu özelliği olan Diferansiyel Faktör özelliğine sahip ise ve bu S-kutu diferansiyel atakta çalıştırılıyorsa, saldırgan ele geçirmeyi hedeflediği bütün bitleri ele geçiremeyebilir.

Teknolojinin gelişmesiyle birlikte gömülü sistemlerinde kullanımı ve bu sistemlerde kullanılacak yeni kritografik araçlara olan ihtiyaç artmaya başladı. Bu yüzden kriptografinin alt dalları değişmeye ve çeşitlenmeye başladı. Bu çeşitlenmenin sonuçlarından biride hafif sıklet kriptografidir. Hafif sıklet kriptografinin amacı güvenlik, maliyet ve performans arasında en iyi şekilde bir denge kurmaktır. Radyo Frekanslı Tanımla etiketleri ve sensörler gibi özkaynağı kısıtlı cihazların kullanımının artmasıyla, hafif sıklet kriptografik modüllere olan ihtiyaçta artmaya başladı. Bu yüzden hafif sıklet kriptografi son yılların en önemli konu başlıklarından biri oldu. Hafif sıklet blok şifre ihtiyacını gidermek üzere tasarlanan blok şifrelere örnek olarak PRESENT, SEA, LED blok şifrelerini verebiliriz. Bu çalışmada, birkaç hafif sıklet blok şifre, onların kriptanalizleri ve bu kriptanalizlerin diferansiyel faktör kullanılarak düzeltilmiş halleri kısaca anlatıldı.

Hafif sıklet blok şifrelerden biri olan LBLOCK, 2011'de düzenlenen Uygulamalı Kriptografi ve İnternet Güvenliği konferansında Wenling Wu and Lei Zhang tarafından önerilen 32-turluk bir yapıya sahip olan blok şifredir.

Bu çalışmada hafif sıklet blok şifrelerden biri olan LBLOCK üzerinde çalıştık. Bu şifreye yapılan diferansiyel kriptanalizi saldırganlar diferansiyel faktörleri gözardı ederek gerçekleştirmiştir ve atağın zaman karmaşıklığıda buna göre hesaplanmıştır. Biz bu atağı diferansiyel faktörleri dikkate alarak inceledik ve atağın zaman karmaşıklığında bir düzeltme yaptık.

Anahtar Kelimeler : Diferansiyel Kriptanaliz, Diferansiyel Faktörler, S-box, Blok Şifreler, Hafif Blok Şifreler

To my father İsmail, my mother Serap and my brother Sefa

ACKNOWLEDGMENTS

Firstly, I would like to express my deepest gratitude to my supervisor Ali DOĞANAKSOY for his invaluable support, guidance, and inspiration. He deserves another thank by introducing me to cryptography.

I would like to give my special thanks to Cihangir TEZCAN for his advices, corrections, constructive criticism, patience, help and for being a constant source of help and guidance. This study would not be complete without his help.

I am heartily thankful to Fatih SULAK, Muhiddin UĞUZ, Neşe KOÇAK, Elif SAYGI and Zülfikar SAYGI for their support, suggestions and being so cooperative during the Authenticated Encryption Group Studies.

It is a great pleasure to thank Murat CENK and Ahmet SINAK for their huge guidance, help, support and understanding during my master studies.

I am grateful to my colleagues Betül AŞKIN ÖZDEMİR, Beyza BOZDEMİR and my friend Birgül KOÇ for their close friendship, collaboration, entertainment, help, and positive energy. They always provided encouragement, motivation, and support since we met. They have been great collaborators and supporters.

I also thank to all my friends and everyone at the Institute of Applied Mathematics.

Lastly, and most importantly, I would like to thank my family whose continuous encouragement, constant support and endless love I have relied throughout of my life. They supported every decision of mine and had always faith in me. None of this would be possible without my family who worked hard to sharpen my skills and have always encouraged and supported me to go further in life.

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 115E447 and titled ”Sözde Diferansiyel Faktörler ve Blok Şifre Atakları Zaman Karmaşıklıkları”.

TABLE OF CONTENTS

CHAPTERS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ACCA	Adaptive chosen ciphertext attack
ACPA	Adaptive chosen plaintext attack
AES	Advanced Encryption Standard
$\mathcal{C}$	Ciphertext space
CCA	Chosen ciphertext attack
COA	Ciphertext-only attack
CPA	Chosen plaintext attack
CS	Cipher State
D_d	Decryption transformation
DDT	Difference distribution table
DES	Data Encryption Standard
E_e	Encryption transformation
IBM	International Business Machines
IoT	Internet of Things
$\mathcal{K}$	Key space
KPA	Known plaintext attack
CS	Key State
$\mathcal{M}$	Message space
NIST	National Institute of Standards and Technology
p_{fa}	Probability of false alarm
p_{nd}	Probability of non-detection
RFID	Radio frequency identification
S/N	Signal to noise ratio
SPN	Substitution-permutation network
$wt(x)$	Hamming weight of the bit vector x
XOR	Exclusive-ored

CHAPTER 1

INTRODUCTION

Cryptology is a science of the design, security, and analysis of the cryptographic algorithms, providing the information security in communication essentially. Cryptology is a general term which covers cryptography and cryptanalysis. While designing secure algorithms to keep messages secret are studied under the science of cryptography, analysing algorithms in terms of security are studied under the science of cryptanalysis.

To achieve information security, cryptography uses various techniques such as data confidentiality, data integrity, authentication, and non-repudiation both in theory and practice. These four are known as major goals and defined as follows [53].

- *Data Confidentiality:* It provides to preserve the information from unjustified people. Its synonyms are secrecy and privacy.
- *Data Integrity:* It provides that the information has not been altered by unauthorized. If a piece of information is manipulated by unauthorized, data integrity provides to detect it.
- *Authentication:* It refers to identification. It is usually utilized that authentication covers both entity authentication and data origin authentication.
- *Non-repudiation:* It provides that an entity in a communication cannot disclaim a previous commitment or action.

Block ciphers, stream ciphers, and hash functions are types of cryptographic primitives. While designing process of these primitives Kerckhoffs's principle is taken into consideration by designers. Auguste Kerckhoffs claimed that the security of the cryptographic system should be guaranteed even if all the system details, except the key, are known by third parties. Kerckhoffs's principle is the following;

- If the system is secure in theory, it must also be secure in practice.
- The secrecy of the system is not a necessity. Moreover, the system can be captured by third parties without inconvenience. That means, in the case of compromising of the system details, the two parties should not be in any trouble.

- The key of the cryptographic system must be transmissible and rememberable without any record. Also, immediately after the willing of two parties, the key must be easily changeable or modifiable.

- The encrypted message, namely cipher, should be deliverable via telegraphic correspondence.

- The equipment and documents of the encryption should be manageable. Especially for the emergency status, a single person should operate without the aggregation of several people.

- The encryption system should be easily used. It should be used without observing a long list of rules.

In a standard communication, there exists two parties, namely a sender and a receiver. Sender applies cryptographic algorithms to make a message unclear to third parties. Plaintext is a message which is intended to send and ciphertext is an unclear message. A plaintext p is an element of message space $\mathcal{M}$ that includes several symbols. A ciphertext c is an element of ciphertext space $\mathcal{C}$ that includes several symbols. The key k used in algorithm is an element of key space $\mathcal{K}$ consists of strings of symbols of an alphabet.

Ciphertext is the output of the encryption transformation. An encryption transformation E_e which contains an encryption function e is a uniquely determined bijection from $\mathcal{M}$ to $\mathcal{C}$ under for each element $e \in \mathcal{K}$. To make the ciphertext readable we want to apply reverse of the encryption process, therefore being a bijection of E_e is the necessary condition. So that, for each distinct ciphertext, we can recover a unique plaintext.

The decryption transformation is an operation of reverting a ciphertext back into a plaintext. A decryption transformation D_d is a uniquely determined bijection from $\mathcal{C}$ to $\mathcal{M}$ under for each element $d \in \mathcal{K}$ contains a decryption function d.

Combination of encryption and corresponding decryption transformation, a message space $\mathcal{M}$, a ciphertext space $\mathcal{C}$, a key space $\mathcal{K}$ construct an encryption scheme. In this scheme, d is uniquely determined by for each e. In mathematical notation, we have $D_d(E_e(p)) = p$ for all $p \in \mathcal{M}$, shortly D_d = E_e^{-1} where e is the encryption key and d is the decryption key.

Asymmetric and symmetric key cryptography can be main parts of the cryptography. In symmetric key cryptography, the same key is used in both encryption and decryption transformation. Parties of a communication use a single secret shared key. In other words, the same key is used for both encrypting and decrypting message. However security of the encryption is provided by a single secret shared key, the symmetric key cryptography is a fast and effective method to protect the confidentiality of the encrypted message. Symmetric key cryptography is classified as block ciphers and stream ciphers.

1.1 Block Ciphers

Block ciphers proceed on blocks of bits or bytes of plaintext at a time instead of each character (bit or byte) of plaintext and produce a corresponding output in a block of ciphertext bits. In other words, block cipher algorithms partition plaintext into fixed size blocks, each block contains the same number of bits, and encryption process is performed on a block at a time, and then ciphertext blocks are produced correspondingly. Fixed size block of plaintext is referred to as the block size of the block cipher. For example, AES which is one of the most known symmetric encryption algorithms has block size 128.

Definition 1.1. Assume that we have a plaintext with a block size b and key with a block size k. Then, a bijective function $BC : \{0,1\}^b \times \{0,1\}^k \to \{0,1\}^b$ is called as a block cipher. Due to the bijection, encryption function is an invertible function for each $k \in \mathcal{K}$.

There are two different types on the basis of the design of the block ciphers.

1. ***Substitution-Permutation Network (SPN)***

   Definition 1.2. In substitution-permutation network (SPN), substitutions and permutations are used in several rounds. Substitution layer contains a permutation $\pi_s : \{0,1\}^l \to \{0,1\}^l$ substitutes each set of l bits for another and that permutation is called as an S-box. Permutation layer contains a permutation $\pi_p : \{0,...,m\} \to \{0,...,m\}$ to mix everything up where m is the number of bit string. While S-box is used for confusion property, permutation is used for diffusion property. Confusion is an encryption operation which is applied to construct an ambiguous link between the key and ciphertext. Diffusion is also an encryption operation that is applied to distribute the impact of an element of plaintext over many ciphertext elements.

2. ***Feistel Network***

   Definition 1.3. A Feistel network performs a series of iterative ciphers on a block of the input. A Feistel network is based on dividing a message into left half (L) and right half (R), and applying encryption function in multiple rounds. In each round, R, the right half the message, remains unchanged. On the other hand, the inputs of the round function F are L and the round key which is also declared as a subkey. The output of the round function is exclusive-ored (XOR) with L and then the result two pieces are swapped. Since the security of the cipher is not affected by the process of swapping two pieces in the last round, that operation is not used. After last round is completed, the two halves, R and L are concatenated to get the ciphertext. Contrary to encryption, subkeys are applied in a reversal order in decryption.

The Data Encryption Standard (DES) [34] which is one of the most popular block ciphers based on Feistel network was developed by a team of cryptographers in the early 1970s. It was extensively accepted and has been published as a government standard. W. Diffie and M. Hellman in [29] and [36] show that the private key of DES is weak to exhaustive key search attack due to the shortness of key, 56-bits. Moreover, it has been theoretically broken by applying a differential attack. Further, software implementation of DES is relatively slow. Therefore, to replace DES with another algorithm, NIST organized competition in 1997. After 3-years analysis, the Rijndael cipher had been announced by NIST as a winner in 2000 and chosen as Advanced Encryption Standard (AES) [1].

Relatively inexpensive machines are distributed with advances in technology, fast and inexpensive computer hardware, and computing performance so that key of some algorithms could be broken in a short time. Consequently, the attack types on cryptographic algorithms have increased and diversified.

1.2 Lightweight Block Ciphers

The embedded system devices have rapidly increased in recent years to improve the quality of our everyday lives in different aspects. These devices are mostly placed in wireless sensor network, smart cards, IoT and RFID technologies. The common feature of these is to be low resource devices. They operate on a limited resource and limited computing power. While designing process of the devices with respect to both hardware and software, the needs of energy consumption, power consumption, security level, communication protocols, data processing and even if area must be considered and chosen to be balanced to operate the device in the best way. When the security of the information on the devices is considered, it must provide an admissible security level to protect the information from the third parties. Therefore, it is crucial to apply the secure cryptographic instruments. Due to the low resources, the integration of the ordinary cryptographic tools is not suitable for constrained devices. Therefore, the needs for new cryptographic tools become an important issue so that lightweight cryptography become a growing trend among both industry and academia to integrate the cryptographic tools into constrained devices.

1.2.1 The Design Process

While conventional cryptography only aims to provide a high level of security, lightweight cryptography must also consider the limitations of memory, power, energy consumption and storage. These limitations have a huge impact on choices of the implementation and design of the algorithms. The designer of the lightweight ciphers usually divides the design process into four stages which are specification, design, implementation, and cryptanalysis.

1. *Specification Stage:* This stage is a determining period on the specification of required threshold values for each design criteria such as the cost of one implementation, latency, memory consumption, power consumption, throughput, chip

area, and side channel resistance. The platform which the algorithm is implemented is generally seen as the most important dependency to specify design criteria.

2. *Design Stage:* This step is a design period of the lightweight block ciphers. The block cipher algorithms are designed according to the design criteria which is determined in the specification stage. The new lightweight block cipher algorithm would be the optimized version of the conventional block cipher algorithms as a totally new algorithm. In general, the lightweight ciphers have smaller block size, linear and non-linear transformations and key schedule of its are simpler with compared to conventional block cipher algorithms.
3. *Implementation Stage:* This step is an implementation period of the lightweight block ciphers as determined in the design stage. If the cost of the implementation is higher than the expectation or desired value, the designer can go back to previous stages and make some changes on these stages.
4. *Cryptanalysis Stage:* This step is a test period of the lightweight block ciphers to find the security level against several cryptographic attacks. If the security level is lower than the expectation or desired value, the designer can go back to previous stages and make some changes on these stages.

The examples of the lightweight algorithms LBLOCK, PRESENT, PRIDE, and RECTANGLE and their cryptanalysis will be given the following chapters in detailed.

1.3 S-Boxes

Definition 1.4. S-box is a permutation $\pi_s : \{0,1\}^s \to \{0,1\}^s$ substitutes each set of s bits for another. In other words, it replaces s-bits with a different set of s-bits.

The security of the ciphers against the attack mostly depends on the substitution layer due to the non-linearity. That means well-formed substitution layer results in strong algorithms which also contain diffusion property. S-boxes are the main element of this layer. They are generally used in substitution layer of cryptographic algorithms to provide confusion property. An n-bit S-box S can also be considered as a vector of Boolean functions.

$$S = (f_1, f_2, ..., f_{n-1}) \quad \text{where} \quad f_i : F_2^n \to F_2.$$

The dot production of x, $y \in F_2^n$:

$$\langle a, b \rangle = \sum_{i=0}^{n-1} x_i y_i.$$

1.3.1 Some S-box Properties

To be more secure against attacks, S-boxes must satisfy some properties.

1.3.1.1 Differential Uniformity

The maximum probability that a specific nonzero input difference activates a specific output difference for the S-box is used to evaluate an algorithm in terms of resistance against differential-like attacks. This parameter is called as the differential uniformity of S.

Definition 1.5. Let S is a function that $S : F_2^n \to F_2^m$. For any $x \in F_2^n$ and $y \in F_2^m$, we define

$$\delta(x, y) = \#\{a \in F_2^n : S(a + x) + S(a) = y\}$$

This forms a multi-set $\{\delta(x, y), x \in F_2^n\ \{0\}, y \in F_2^m\}$. The maximum of this multi-set

$$\delta_S = \max_{x \neq 0, y} \delta(x, y)$$

is the differential uniformity of S.

In simple terms, the differential uniformity of S is the maximum value in a difference distribution table of an S-box, excluding the zero difference case. In differential-like attacks, the differential characteristic is used with high differential probability. Therefore, the intention of the S-box designer is to keep differential uniformity at the minimum value.

1.3.1.2 Non-linearity and Linearity

The Hamming distance is the principal criteria. To design more powerful S-box with regard to linear attacks, designer aim to maximize the Hamming distance. The correlation of a linear approximation can be calculated by using the Walsh transform and it is defined as

$$W_S(x, y) := \sum_{a \in F_2^n} (-1)^{\langle x,a \rangle + \langle y,S(a) \rangle}$$

Definition 1.6 ([20])**.** The linearity of a given S-box is defined as

$$Lin(S) = \max_{x, y \neq 0} |W_S(x, y)|\,.$$

The least value of the hamming distance, namely d, between the component functions of the S-box and all affine functions is called as a non-linearity of an S-box and is represented by N_S.

$$d = 2^{n-1} - \frac{Lin(S)}{2} = N_S$$

To have stronger S-box against linear attacks, $Lin(S)$ must be smaller. In other words, N_S must be higher.

1.3.1.3 Branch Number

Definition 1.7 ([59])**.** The branch number of an $n \times n$ S-box is

$$BN = \min_{x,y \neq y}(wt(x \oplus y) + wt(S(x) \oplus S(y))).$$

where $x, y \in F_2^n$, and $wt(x)$ is the Hamming weight of the bit vector x.

The branch number of a bijective S-box can be at least 2. The algebraic [25] and cube attacks[30] are strongly linked with the branch number property.

1.3.1.4 Balanced Function

Definition 1.8 ([23])**.** Assume that we have a function $S : F_2^n \to F_2^m$ where $m \leq n$. The function S is called as a balanced function if each value of F_2^m appears 2^{n-m} times.

In other words, if the number of ones and zeros equal to each other in the truth table of an S-box is said as a balanced.

If a cryptographic algorithm contains an unbalanced S-boxes, the diffusion layers of that algorithm must be designed heavier [57].

1.3.1.5 Robustness

Definition 1.9 ([63])**.** Let $S = (f_1, ..., f_l)$ be an $m \times l$ S-box, where f_i is a function on V_m, $i = 1, ..., l$ and $m \leq l$. The representation D and T are for the greatest value in the difference distribution table of S and the number of nonzero entries in the first column of the DDT, respectively. In either case, the value of the first entry of the first row 2^n is not counted. S is said that R-robust against differential attacks and the description of R is following equation:

$$R = (1 - \frac{T}{2^n})(1 - \frac{D}{2^n})$$

To take more accurate information on S-boxes in terms of strength, robustness of an S-box should be observed. Since the discussion of robustness is more complicated compared with the differential uniformity, differential uniformity is applied as the first indicator for the strength of an S-box. Robustness is usually applied when the need for more information about the strength arises.

1.3.1.6 Strict Avalanche Criteria (SAC)

If a bit of the plaintext is changed, half of the output bits should be changed. That can be given as the informal definition of the strict avalanche criteria. In other words, if there is a minor change on the input, that should result in an important change in the output.

If $S(x) \oplus S(x \oplus a)$ is balanced where $wt(a) = 1$, S-box satisfy SAC.

1.3.1.7 Undisturbed Bits

Definition 1.10 ([73])**.** 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.

Some bits of the output difference does not change for a specific input difference and these bits are called as undisturbed bits. The difference distribution tables of L-BLOCK cipher is considered as an example. We notice that a bit or two bits of the corresponding output difference does not change for the input differences 1,2,3,8 and B of each S-boxes. The undisturbed bits of the S-boxes of LBLOCK are shown in Table 1.1. To find longer differential characteristics, undisturbed bits can be applied to the algorithm and this result with more effective differential attacks. The differential attacks of PRESENT, RECTANGLE, PRIDE and SERPENT block ciphers in the literature are corrected or improved by applying undistırbed bitd.

1.3.1.8 Differential Factors

Tezcan [77] defines the differential factors as follows.

Definition 1.11 ([77])**.** Let S be a function from F_2^n to F_2^m. For all $x, y \in F_2^n$ 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.

The details are given in Chapter 2.4.

We can summarize the properties for a good S-box against most known attacks as follows.

- The S-box should contain balanced component functions,
- The S-box should have high non-linearity,
- The differential uniformity of the S-box should be low,
- The S-box should satisfy SAC,

Table 1.1: Undisturbed Bits of LBLOCK S-boxes

S-box	Input Difference	Output Difference
S_0, S_8	0001, 0010	???1
	0011	??10
	1000	??1?
	1011	??0?
S_1, S_9	0001, 0010	??1?
	0011	??01
	1000	???1
	1011	???0
S_2	0001, 0010	??1?
	0011	1?0?
	1000	1???
	1011	0???
S_3	0001, 0010	???1
	0011	???0
	1000	?1??
	1011	?0??
S_4	0001, 0010	???1
	0011	1??0
	1000	1???
	1011	0???
S_5	0001, 0010	???1
	0011	?1?0
	1000	?1??
	1011	?0??
S_6	0001, 0010	??1?
	0011	??01
	1000	???1
	1011	???0
S_7	0001, 0010	??1?
	0011	??01
	1000	???1
	1011	???0

- The S-box should have high algebraic degree.

1.4 Classification of Cryptanalytic Attacks

A cipher is said to be totally broken when the secret encryption key is captured by an attacker so that the ciphertext can be decrypted to plaintext. On the other hand, to be partially broken, some parts of plaintext from the ciphertext can be captured by an attacker even if the key is unknown. Some of the widely discussed attack techniques are given below.

1.4.1 Cryptanalytic Techniques

1. ***Ciphertext-only attack (COA):*** Aim of an attacker is to derive the decryption key or the plaintext by just the knowledge of ciphertext. If the attacker successively perform the attack, the encryption scheme of the algorithm is completely insecure.

2. ***Known plaintext attack (KPA):*** The aim of the attackers is to find unknown plaintexts-ciphertext pairs or derive directly the key since some plaintexts-ciphertext pairs are known by the attacker in this technique.

3. ***Chosen plaintext attack (CPA):*** The attacker is assumed to have a chance to select arbitrary plaintexts and then get corresponding ciphertexts. Detection of

some ciphertext dependencies to use later to complete the analysis and attack is aimed by the attacker. The most known example of this technique is differential cryptanalysis.

4. ***Chosen ciphertext attack (CCA)*** Opposed to CPA, the attacker has a chance to select arbitrary ciphertexts and get corresponding plaintexts.

5. ***Adaptive chosen plaintext/ciphertext attack (ACPA/ACCA):*** Likewise CPA, the plaintexts can be selected by the attacker and then corresponding ciphertexts can be get but this time choices are not random so that the attacker choose other ciphertexts depending on the information received from previous choices.

1.4.2 Complexities and Elementary Attacks

Required resources to mount an attack are also used to compare attacks between each other. The definitions of these resources are given below.

1. ***Data Complexity:*** Expected number of plaintexts (ciphertexts) to mount the attack.

2. ***Memory Complexity:*** Expected number of memory (storage) units to mount the attack.

3. ***Time Complexity:*** Expected number of operations required for execution of the attack. The number of the encryptions or decryptions of concerned cipher is taken as a time complexity.

In [56], to attack any block cipher in any structure, three fundamental cryptanalytic techniques are described as follows:

1. ***Dictionary Attack:*** It is a kind of brute-force attack, in which the attacker attempts to guess a key by trying plaintexts. The attacker chooses a plaintext and encrypts it with 2^k possible keys and stores the corresponding ciphertexts in a sorted dictionary. If the attacker found the ciphertext which is searching for, the secret key can be found by checking for a match in the sorted dictionary. Clearly, since the attacker generates the dictionary table, which requires 2^k encryptions, before performing the attack, total time complexity are not affected by this step. Also, the time complexity of searching for a match in the sorted dictionary is negligible. When the dictionary attack is performed for a block cipher with n-bit block size, the data complexity is 1 plaintext and the memory complexity is 2^k n-bit words.

2. ***Codebook Attack:*** This attack is an example of a known plaintext attack scenario. The attacker construct codebook with these pairs. For any given ciphertext, the attacker could partially or fully decrypt it by looking up the codebook even if the key is secret. When the Codebook attack is performed for a block

cipher with b-bit block size, the data complexity of the attack is 2^b pairs of plaintexts and ciphertexts. The time complexity is negligible since it is just one access of look-up table. The memory complexity is $2^b 2b$-bit values.

3. ***Exhaustive Key Search:*** It is basic and the simplest technique for a cryptosystem. The attacker tries all 2^k possible keys by encrypting the plaintext, till the correct key is found. This type of attack can be performed on any cipher. The time complexity is 2^k.

Table 1.2: The Complexities of the Elementary Attacks

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

1.5 The Structure of the Thesis

In this thesis, we study on LBLOCK's security against to differential attack. This thesis is organized as follows. In this Chapter 1, a brief instruction on the block ciphers, lightweight block ciphers and the design principles of the lightweight block ciphers are provided. In addition, we shortly define S-boxes which the non-linear layer of the algorithms and give some properties of the S-boxes. Moreover, we investigate cryptanalytic attacks that are applied to determine the security level of the ciphers. In Chapter 2, we present the differential cryptanalysis that is one of the most commonly used cryptanalytic attack and the theory of the differential factors. Also, this chapter covers some information about the algorithms that the S-box is used in the algorithm have the differential factors. Moreover, the differential cryptanalysis of some algorithm and the corrected version by using differential factors are given as an example in this chapter. In Chapter 3, the block cipher LBLOCK is expressed. Also, the differential attack on LBLOCK is briefly discussed. We explain our modification attacks to the mentioned differential attack. In chapter 4, we finalize the thesis.

CHAPTER 2

DIFFERENTIAL CRYPTANALYSIS

Differential cryptanalysis is one of the most known analysis method. In this chapter, we will give brief information about it.

2.1 Introduction to Differential Cryptanalysis

Eli Biham and Adi Shamir [14] introduced the differential cryptanalysis in 1990 to break reduced-round versions of DES [34]. Then it was extended in 1991 to break full 16-round of DES [15]. In cryptanalysis of a block cipher, differential cryptanalysis is one of the most effective technique. It is a chosen plaintext attack (CPA). Since the attacker has arbitrary plaintexts-ciphertexts pairs, the attacker can constitute plaintext pairs under the specific differences and then can analyze the corresponding ciphertext pairs that how to be affected under specific differences. Let us say, the attacker has a plaintext pair namely P_1 and P_2 which has a specific difference and their corresponding ciphertext pairs after r-rounds encryption under the same key namely C_1 and C_2. Now, the attacker observes that the difference between P_1 and P_2 how affects the difference between C_1 and C_2. In other words, for a specific difference between plaintext pairs, the aim of the attacker is to get an output difference that occurs with high probability in a certain round. Because of being CPA, it is possible to distinguish the algorithm from random permutation using known plaintext-ciphertext pairs. Moreover, subkeys of the attacked rounds can be determined. The differences of input pairs for each round are independent from the round subkey. These difference are used to mount differential cryptanalysis. To exemplify, assume X and X^* are two-bit strings, K is the subkey of a round and $\oplus$ is a group operation used to combine input with the subkey of a round in a cipher, the difference ΔX between the two-bit strings is calculated with

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

The impact of the key is eliminated for the differences between $(X \oplus K)$and $(X^* \oplus K)$ when the key addition is performed under the same key K.

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

The operation $\oplus$ is chosen as the XOR operation in most of the ciphers. Since S-boxes have a nonlinear structure, the output difference of the S-box may not be absolutely identified for each given an input difference. In other words, if there is ΔX difference between X and X^* which are inputs of S-box, the ΔX difference may yield several output differences ΔY between Y and Y^* which are outputs of S-box, respectively. That gives the attacker many choices to mount the attack. For this reason, the attacker creates a difference distribution table (DDT) that is also called as XOR table of an S-box. The number in the entries of table represents that how many times given an input difference result in an output difference.

2.1.1 Difference Distribution Table

Difference distribution table is a useful and important tool for differential attack. The attacker can create a table by using computer or by manual computation. If the size of an S-box in an algorithm is small, the attacker can create the table quickly by manually. On the other hand, it is a time lost to create the table for big size S-boxes. For the LBLOCK case, since the algorithm has eight different S-boxes, we create the table by the help of computer. Yet, we construct a table for the S-box S_0 of LBLOCK by manually to exemplify. We first choose an input difference $\Delta X = 0101$ and the start to create a row of the DDT. The second input values are the XOR of each input value of the S-box and ΔX. For illustration, let $X = 0000$, then the second input $X^{'}$ equals to $X^{'} = X \oplus \Delta X = 0000 \oplus 0101 = 0101$. The outputs for the inputs $X, X^{'}$ are $Y = 1110$ and $Y^{'} = 0100$, respectively. The output difference $\Delta Y = Y \oplus Y^{'} = 1110 \oplus 0100 = 1010$. The same procedure is applied to all X values to create the Table 2.1.

From Table 2.1,

$$\begin{array}{llll}
\Delta Y = 0100 & \text{occurs} & 4 & \text{times,} \\
\Delta Y = 0101 & \text{occurs} & 2 & \text{times,} \\
\Delta Y = 0111 & \text{occurs} & 2 & \text{times,} \\
\Delta Y = 1010 & \text{occurs} & 4 & \text{times,} \\
\Delta Y = 1101 & \text{occurs} & 2 & \text{times,} \\
\Delta Y = 1111 & \text{occurs} & 2 & \text{times.}
\end{array}$$

Table 2.2 shows that the number of occurrences of output differences when input difference $\Delta X = 0101$.

Table 2.2 is a row of DDT. To construct a whole table, 2.3, we apply the same procedure.

In Table 2.3, while each row that is represented by ΔX stands for an input difference, each column that is represented by ΔY stands for an output difference. Each entry of the table represents the number of occurrences of input-output difference pairs.

2.2 Differential Attack on Sample Cipher

Definition 2.1. The differential refers to the difference pair (ΔX,ΔY).

Table 2.1: Sample Difference Pairs of the S-box S_0

X	Y	$X^{'} = X \oplus \Delta X$	$Y^{'}$	$\Delta Y = Y \oplus Y^{'}$
0000	1110	0101	0100	1010
0001	1001	0100	1101	0100
0010	1111	0111	1011	0100
0011	0000	0110	1010	1010
0100	1101	0001	1001	0100
0101	0100	0000	1110	1010
0110	1010	0011	0000	1010
0111	1011	0010	1111	0100
1000	0001	1101	0110	0111
1001	0010	1100	0111	0101
1010	1000	1111	0101	1101
1011	0011	1110	1100	1111
1100	0111	1001	0010	0101
1101	0110	1000	0001	0111
1110	1100	1011	0011	1111
1111	0101	1010	1000	1101

Table 2.2: Occurrences of ΔY when ΔX = 0101

ΔY	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Occurrence	0	0	0	0	4	2	0	2	0	0	4	0	0	2	0	2

Table 2.3: The Difference Distrubution Table of the S-box S_0

ΔY ΔX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	4	0	2	0	0	0	2	0	4	0	2	0	0	0	2
2	0	4	0	2	0	0	0	2	0	4	0	2	0	0	0	2
3	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0
4	0	0	0	2	4	2	4	0	0	0	0	2	0	2	0	0
5	0	0	0	0	4	2	0	2	0	0	4	0	0	2	0	2
6	0	0	4	0	4	2	0	2	0	0	0	0	0	2	0	2
7	0	0	0	2	4	2	0	0	0	0	0	2	0	2	4	0
8	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2
9	0	0	0	2	0	0	0	2	4	0	0	2	4	0	0	2
A	0	4	2	0	0	0	2	0	0	0	2	0	0	4	2	0
B	0	4	0	0	0	0	0	0	4	0	0	0	4	4	0	0
C	0	0	2	2	0	2	2	0	2	2	0	0	2	0	0	2
D	0	0	0	2	0	2	0	0	2	2	2	0	2	0	2	2
E	0	0	2	0	0	2	2	2	2	2	0	2	2	0	0	0
F	0	0	0	0	0	2	0	2	2	2	2	2	2	0	2	0

Definition 2.2. Differential characteristic is a sequence of input-output differences of each round. The output difference of a round is also the input difference of the next round. A high probability characteristic should be found to perform the attack efficiently. To accomplish that the high probability occurrences of difference pairs should be investigated from difference distribution table. An adequate number of rounds characteristic to mount the attack can be obtained by joining one round characteristics logically.

Definition 2.3. Suppose that we have an $l \times r$ S-box S, α and β with l and r-bit block, respectively. The number of obtaining the output difference β is equal to the cardinality of the set for given the input difference α. It is defined as

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

Definition 2.4. Suppose that we have an $l \times r$ S-box S, α and β with l and r-bit block, respectively. The probability of the differential $\alpha \rightarrow \beta$ for S is defined as

$$Pr_S(\alpha \rightarrow \beta) = Pr(S(x) \oplus S(x \oplus \alpha) = \beta), x \in \{0,1\}^l$$

Definition 2.5. Suppose that we have an $l \times r$ S-box S, α and β with l and r-bit block, respectively. The probability of obtaining an output difference β from S if the input difference is α is defined as

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

From the definitions, one can obviously see that zero input difference produces always zero output difference. By definitions, the number of obtaining that difference pair is 2^m and the probability of that difference pair is 1. If the S-boxes have a nonzero input difference, they are called as active S-boxes. Moreover, product of probabilities of active S-boxes equals the probability of a round, in general.

If the attacker gets a one-round differential characteristic $\alpha \rightarrow \beta$ with probability p, then to get second one-round characteristic by concatenating that the output difference β corresponds to the input difference of the second characteristic. In a similar way, the attacker can construct r-rounds differential characteristics. When the attacker performs the attack, the rounds are assumed as independent from each other. Therefore, the multiplication of probabilities of one-round characteristics equals to probability of the concatenated differential characteristic.

Definition 2.6. The description of the right pair regards to a characteristic is that a plaintext pair should have the same intermediate differences with the values specified by the characteristic. If any pair is not a right pair, then it is referred to wrong pair.

For differential cryptanalysis, the attacker first aims to find a scenario provides a differential characteristic with a very high probability for a distinguisher. It is possible to distinguish the cipher from a random permutation by using the characteristic with high probability and enough number of chosen plaintexts, say N. After determining the distinguisher, to mount a differential attack, some rounds can be added to the beginning and/or to the end of the distinguisher, and guess the corresponding subkeys by checking the differences. The important point is to mount attack that the use of enough number of plaintexts, N. Suppose that the attacker performs the attack on the algorithm with n-bit block size and determines a differential characteristic with probability p, $p >> 2^n$, the attacker requires theoretically $1/p$ chosen plaintext pairs. That is required to distinguish the algorithm from a random permutation. On the other hand, some wrong plaintext pairs can also give the right input and output difference but not satisfies the intermediate differences. These pairs are called as noise. For this situation, instead of choosing $1/p$ plaintext pairs, $c \times 1/p$ plaintext pairs should be chosen. Signal to Noise Ratio property is applied to find a small constant c.

Definition 2.7 ([64]). The signal to noise ratio is defined as the ratio of the probability of the right key being suggested by a right pair to the probability of a random key being suggested by a random pair with the given initial difference is called and is denoted by S/N,

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

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

When performing attack, the attacker assumes that the cipher with wrong keys acts as a random permutation. Assume that we have probability of the distinguisher from a plaintext-ciphertext pair with the correct key that is represented by p_0. On the other hand, p represents the probability of the distinguisher from a plaintext-ciphertext pair with the wrong subkeys, a random permutation.

If the attacker keeps a counter to determine the distinguisher for every key, s/he must observe whether or not counter of the correct subkey and a wrong subkey is distributed binomially with parameters N, p_0, p. The attacker needs a threshold T between the expected values of these distributions which is big enough to distinguish. The production of $N \cdot p$ and $N \cdot p_0$ is equal to the expected values of distributions. The correct key candidates are represented by the keys on the right side of the threshold. Exhaustive attack can be applied to get remaining key bits to find the right key. Note that, in general, the probability of the distinguisher p_0 is bigger than the probability of random permutation p and $p_0/p \geq 4$.

Definition 2.8. Non-detection is the counter for the right key that is on the left side of the threshold. The probability of non-detection is denoted by p_{nd}.

Definition 2.9. False alarm is the counter for the wrong key that is on the right side of the threshold. The probability of false alarm is denoted by p_{fa}.

The probabilities of non-detection and false alarm should be very small to accomplish the attack in optimal time and data complexity. While the probabilities of non-detection and false alarm decrease in the case of increasing plaintext-ciphertext pairs N, obviously the data complexity increases. The increase in data complexity will also cause an increase in time complexity. As we mentioned above, enough number plaintext-ciphertext pairs N must be found to perform the attack in optimal complexities.

One of the expectations from a cryptographic algorithm is that finding the correct key must be harder than discarding most of wrong ones. Therefore the time complexity and data complexity is much higher for finding the correct key. That is why the advantage is defined:

Definition 2.10 ([64])**.** Let n-bit key are attacked and then we have 2^n possible key candidates. If the right key is positioned among the top t out of all candidates, then it can be said that the attack obtained an $(n - \log t)$-bit advantage over exhaustive search.

2.3 Types of Differential Cryptanalysis

Differential cryptanalysis can be categorized into several types. These types:

2.3.1 Truncated Differential Cryptanalysis

It is first introduced by [43] in 1994. It is an extension of differential cryptanalysis. Differentials are partially predicted in truncated differential cryptanalysis. With the help of this information, the complexity of standard differential cryptanalysis can be reduced since prediction is made on only a part of the differences.

Definition 2.11 ([43])**.** Suppose that we have an r-round differential $(P; C)$. If there exists a subsequence P_0 and C_0 of P and C, respectively and these subsequences form a differential, then $(P_0; C_0)$ is called an r-round truncated differential.

If the attacker attacks to a byte-oriented block cipher, bytewise truncated differentials can be used where 1 (non-zero) represents one-byte difference, 0 (zero) represents no difference.

Since the differential can be predicted partially, the ratio of p_0 and p differs from the differential attacks. While the ratio is $p_0/p \geq 4$ in differential attacks, the ratio will be $p_0/p \approx 1$ in truncated differential attacks. That is the main difference between two attacks.

2.3.2 Higher-Order Differential Cryptanalysis

It examines the effects of a set of differences between a larger set of input texts contrary to the differential cryptanalysis that examines the difference between only two input texts [43].

2.3.3 Related-Key Differential Cryptanalysis

It is first presented by L. Knudsen [42] and Biham [11] independently. In the related key differential attack, the relationship between several keys is known by the attacker. These type attacks are generally mounted to reveal the vulnerabilities of the key schedule of the algorithm. Due to the vulnerabilities of the key schedule of the algorithm, the strength of the algorithm against some attacks may decrease.

2.3.4 Impossible Differential Cryptanalysis

It is first presented by Biham, Biryukov, and Shamir in [13].

Attacker tries to find an impossible differential characteristic with probability 0. In other words, the impossible differential examines the differences that are impossible contrary to the differential cryptanalysis that examines the difference higher than expected probability. Such a differential can be get by using the miss-in-the-middle technique.

If impossible differential characteristic is provided under a candidate key, that candidate key can not be the correct key so that it can be discarded.

2.3.5 Improbable Differential Cryptanalysis

It is first presented by Tezcan in [72].

In the improbable differential cryptanalysis, attacker tries to find an improbable differential characteristic with less probability than a random permutation. In other words, the given differences occur less under the correct key than a wrong key in the improbable differential. To get such a differential, one can apply to impossible differential technique. Therefore, the impossible differential cryptanalysis can be considered as a subset of improbable differential cryptanalysis.

2.4 Differential Factors

A way of providing confusion is the usage of S-boxes and they are a crucial point in the security of the cryptographic algorithms. Designers select the S-boxes according to special cryptographic purposes which would vary from algorithm to algorithm. For example, small S-boxes are selected in most of the lightweight algorithms because lightweight algorithms have generally hardware-oriented design and they are desired to work fast in hardware. In addition, designers also consider which S-boxes have the best security against known cryptanalytic techniques: If an S-box has low non-linear and differential uniformity [55], one can say that this S-box is resistant to linear and differential cryptanalysis, respectively. Also, to resist to algebraic [25] and cube [30] attacks, an S-box should have high algebraic degree and branch number, respectively. Therefore, properties of the selected S-box should provide the best security against known attacks. One of these properties is differential factors which were introduced by Tezcan [77] at LightSEC 2014. If an S-box has the differential factor, the attacker may not capture all bits of the attacked round key bits while performing the differential cryptanalysis or a variant of it. Due to the differential factors, the right and some wrong keys may have a relation. That means they would get the same hits for counter in key guessing step of the differential attack. This helps to attackers to separate the key space into two or more disjoint sets. Hence, attacker may shorten the attacked key space significantly.

In differential cryptanalysis, an attacker can observe the relation between a specific difference in plaintext pairs, which is called as an input difference, and the difference of the corresponding ciphertext pairs, which is called as an output difference. While performing the differential cryptanalysis on an algorithm, the aim of the attacker is to get subkeys corresponding to the S-box activated by a differential on differential path. An S-box activated by a differential means that an S-box has a nonzero input difference and hence it has a nonzero output difference. If attacker finds a differential to use it as a distinguisher, statistical tests can be applied to find the right key. For example, attacker takes N plaintext/ciphertext pairs and keeps counters for each round keys that satisfy this distinguisher in differential attack. The right key has the highest counter value among candidate subkeys. If the S-box has the differential factors property, attacker gets the same hits for both a wrong key and the right key. Therefore, attacker can not distinguish the guessed keys. We can say that an S-box which has differential factors may be a disadvantage for the attackers since they prevent to detect some part of the attacked round key.

2.4.1 The Theory of Differential Factors

Cihangir Tezcan firstly defined the differential factors in [77] as follows.

Definition 2.12 ([77])**.** Let S be a function from F_2^n to F_2^m. For all $x, y \in F_2^n$ 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.

As an example;

In Table 2.4 the values of S_2 of LBLOCK can be seen. First, we investigate the existence of the differential factors of S_2 by trying all possible values of x, y, λ and μ which satisfy above condition. We find that S_2 has a differential factor $\lambda = 3_x$ for the output difference $\mu = 1_x$. That means, for all $x, y \in F_2^n$ satisfies $S(x) \oplus S(y) = 1_x$, then we also have $S(x \oplus 3_x) \oplus S(y \oplus 3_x) = 1_x$. We create Table 2.5 according to these calculations. If we take $x = 0_x$ and $y = 6_x$, we get

$$S(0_x) \oplus S(6_x) = 1_x \quad \text{and}$$
$$S(0_x \oplus 3_x) \oplus S(6_x \oplus 3_x) = S(3_x) \oplus S(5_x) = 1_x.$$

All values of x and y which satisfy the condition can be seen in Table 2.5.

Table 2.4: An S-box of LBLOCK, S_2

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S_2(x)$	1	14	7	12	15	13	0	6	11	5	9	3	2	4	8	10

Table 2.5: Differential factor example of S_2 of LBLOCK for $\lambda = 3_x$ and $\mu = 1_x$

(x, y)	6 0	12 11	13 9	7 2	14 10	15 8	3 5	1 4
$(S(x), S(y))$	0 1	2 3	4 5	6 7	8 9	10 11	12 13	14 15

Theorem 2.1 ([77]). *Let us assume that we have input pair (x, y), the partial subkey k and an S-box S containing a differential factor λ for an output difference μ in a block cipher based algorithm. We know that a subkey and an intermediate value of the message are generally XORed right before the S-box operation so that they form the input of S-box. If the partial subkey k and the input pair (x, y) give the output difference μ, then input pair (x, y) and the partial subkey $k \oplus \lambda$ would also give the same output difference μ. Thus, we can think that the attacker can not detect a bit of the partial subkey which corresponds to the output difference μ. The advantage of the cryptanalyst is reduced by 1 bit since a bit can not be detected. Hence, the time complexity of this key guess step is halved.*

Proof. We know that the attacker keeps a counter for each candidate key in a differential attack. Since we have an S-box containing a differential factor λ for an output difference μ, the differential path is satisfied by both key *k* and $k \oplus \lambda$. Therefore, their counter would be equal to each other. If the attacked key bits are equal to n, the attacker would form two key sets which include 2^{n-1} candidate keys corresponding to k and $k \oplus \lambda$. Since counters of each k and $k \oplus \lambda$ are the same, the two key sets can not be distinguished by the attacker. That means half of the guessed keys can not be distinguished with the other half. Thus, it is enough to use one key set. Therefore, the time complexity of this step is halved. Since the number of the guessed keys is equal to 2^{n-1} instead of 2^n, we can think that the advantage of the cryptanalyst is reduced by 1 bit.

The following theorem gives information about the number of differential factors of an S-box and inverse of it.

Theorem 2.2 ([74]). *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

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

We note other useful properties of differential factors.

Theorem 2.3 ([74])**.** *If λ_1 and λ_2 are differential factors for an output difference μ, then $\lambda_1 \oplus \lambda_2$ is also a differential factor for the output difference μ i.e. All differential factors λ_i for μ form a vector space.*

Corollary 2.4 ([77])**.** *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, the 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.*

The corollary is actually a generalization of the Theorem 2.3 and says that different differential factors in S-box form a vector space for the output difference μ. Also, r-dimensional vector space indicates that the attacker can not detect r-bits of attacked subkey by in the key guess step of a differential attack. Therefore, if the attacked key bits are equal to n, the time complexity of key guessing step would be equal to 2^{n-r} instead of 2^n.

Theorem 2.5 ([74])**.** *Differential factors reduce the key space for the key guess process and therefore reduce the data complexity of the attack. Thus, memory required to keep the counters for the guessed keys also reduces. Reduction in the data complexity may also reduce the time complexity depending on the attack.*

As we mentioned in Theorem 2.1, if we attack n-bits and have an S-box containing a differential factor λ for an output difference μ, it is enough to form a subkey set which includes 2^{n-1} different candidate subkeys instead of forming a set including all 2^n candidate subkeys. Hence, the guessed key space is reduced because of the effect of the differential factors on key guess step. Moreover, the guessed key space is used in the computation of the required data to perform the attack. Since the guessed key space is reduced, data complexity of the attack is also reduced. Overall time complexity of the attack depends on both complexities of guessing some key bits using distinguisher and the complexity of the obtaining remaining key bits with exhaustive search. If differential factors are used in the attack, complexity of guessing some key bits using distinguisher are reduced by differential factors. On the other hand, remaining key bits are increased. Therefore, complexity of obtaining remaining key bits with an exhaustive search is increased. If the complexity of obtaining remaining key bits with an exhaustive search is higher than the complexity of the guessing some key bits using distinguisher, the reduction on the complexity of the guessing some key bits does not affect overall time complexity of the attack.

Since differential factors depend on several components like the design of cipher, the differential used in the attack, and the choice of S-box, attackers may not use them in all differential attacks. If they can use, time and data complexity of the attack can be directly affected by this usage.

2.4.2 Studied Algorithms for Differential Factors

In this section, we briefly describe some algorithms which are investigated as part of this thesis. The S-boxes of these algorithms contain differential factors and they are given in [74], [76]. However, we do not investigate some algorithms that are given in [75] in this work. The names of the algorithms and differential factors of their S-boxes are shown in Table 2.6.

Table 2.6: The Differential factors of S-boxes of new algorithms

Algorithm	λ	μ
ROADRUNNER [8]	1	1
ROADRUNNER	8	2
QTL [46]	1	5
QTL	F	F
MIDORI [7]	2	2
MIDORI	A	A
HISEC [5]	4	4
HISEC	F	E
KHUDRA [45]	1	5
KHUDRA	F	F
LAC [87]	B	1
LAC	3	4
PROST [41]	1	1
PROST	8	8
JOLTIK [37]	1	2
JOLTIK	2	5
FOX [38]	44	11
FOX	88	22
FOX	CC	33
FOX	55	44
FOX	11	55
FOX	DD	66
FOX	99	77
FOX	AA	88
FOX	EE	99
FOX	22	AA
FOX	66	BB
FOX	FF	CC
FOX	BB	DD
FOX	77	EE
FOX	33	FF
SAFER [51]	80	01
SAFER	80	80
SAFER	80	81

1. ***CRYPTON:*** CRYPTON was introduced by Chae Hoon Lim in 1998 [47] and modified in 1999 [48]. CRYPTON is a block cipher consists of a 12-round substitution-permutation network with 16 bytes data block and up to 32-bytes key length. In the substitution layer of initial version of CRYPTON, 2 different 8×8 S-boxes are used. These S-boxes contain differential factors. But the new S-boxes in revised version of CRYPTON do not contain differential factors.

   Since the two S-boxes S_0 and S_1 are big enough, we do not give them in this work. We just give their differential factors in Table 2.7. For those who want to check S-boxes, they found them in [47].

   We observe differential-like cryptanalysis of CRYPTON to improve the attack using differential factors. But the differential attacks on CRYPTON in the literature are not given in detail. Therefore, we could not correct or improve these attacks.

   Table 2.7: The Differential factors of CRYPTON S-box S_0 and S_1

S-box	λ	μ
S_0, S_1	10	10
S_0, S_1	20	20
S_0, S_1	30	30
S_0, S_1	40	40
S_0, S_1	50	50
S_0, S_1	60	60
S_0, S_1	70	70
S_0, S_1	80	80
S_0, S_1	90	90
S_0, S_1	A0	A0
S_0, S_1	B0	B0
S_0, S_1	C0	C0
S_0, S_1	D0	D0
S_0, S_1	E0	E0
S_0, S_1	F0	F0

2. ***GOST:*** GOST is one of Russian cryptographic standard algorithms. It is a Feistel network type cipher with 32-round, 64-bit block size and 256-bit key size[33]. In substitution layer of GOST, 8 different 4×4 S-boxes are used. We do not give these S-boxes in this work, we just give their differential factors in Table 2.8.

   We observe differential cryptanalysis of GOST [26] to improve the attack using differential factors. Since the differential attack is used multiple characteristics, we do not give any correction or improvements to the attack.

   Table 2.8: The Differential factors of GOST S-box

S-box	λ	μ
S_1	5	3
S_4	D	5
S_6	9	B
S_8	7	5
S_8	E	6

3. ***LBLOCK:*** LBLOCK was introduced by Wenling Wu and Lei Zhang in 2011. LBLOCK is a Feistel network cipher with 32-round, 64-bit block size, and 80-bit key size.

   We observe differential cryptanalysis of LBLOCK [24] to improve the attack using differential factors. The details are given in Chapter 3.

4. ***LED:*** LED was introduced by Jian Guo et al. in 2011 [35]. It is an AES-like design with 64-bit block size and supports 64-bit and 128-bit key size. LED cipher uses the same S-box with PRESENT. S-box and differential factors of it are given in Section 2.5.

   We study on the differential cryptanalysis of LED to improve the attack using differential factors. But in the literature, there are no differential attacks on LED. Therefore, we do not give any correction or improvements to the attack.

5. ***LUFFA:*** LUFFA was introduced by De Canniére et al. as a candidate algorithm for SHA-3[22]. For the diffusion, a 4×4 S-box is used in LUFFA. Table 2.9 and Table 2.10 shows S-box and its differential factors, respectively.

   Since it is a hash function, we don't observe it in terms of differential factors.

Table 2.9: The S-box of LUFFA

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s(x)$	7	D	B	A	C	4	8	3	5	F	6	0	9	1	2	E

Table 2.10: The Differential factors of LUFFA S-box

λ	μ
4	1
2	2

6. ***NOEKEON:*** NOEKEON was presented by Daemen et al. [27]. The block and key size of the algorithm are 64-bit and 128-bit, respectively. It works with 16-round. In the substitution layer of NOEKEON, a 4×4 S-box is used. Table 2.11 and in Table 2.12 shows S-box and its differential factors, respectively.

   We study on the differential-like cryptanalysis of NOEKEON to improve the attack using differential factors. But in the literature, there are no differential attacks on NOEKEON, we do not give any correction or improvements to the attack.

Table 2.11: The S-box of NOEKEON

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s(x)$	7	A	2	C	4	8	F	0	5	9	1	E	3	D	B	6

7. ***PICCOLO:*** Piccolo was introduced by Kyoji Shibutani et al. in 2011 [66]. It is a lightweight block cipher and Feistel network. The block size equals to 64-bit. 80-bit and 128-bit key size are supported in PICCOLO. While Piccolo-80 works

Table 2.12: The Differential factors of NOEKEON S-box

λ	μ
1	1
B	B

with 25-rounds, Piccolo-128 works with 31-rounds. In substitution layer, a 4×4 S-box is used for 4 times. 2.13 and Table 2.14 shows S-box and its differential factors, respectively.

We study on the differential-like cryptanalysis of PICCOLO to improve the attack using differential factors. But in the literature, there are no differential attacks on PICCOLO. Therefore, we do not give any correction or improvements to the attack.

Table 2.13: The S-box of PICCOLO

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s(x)$	E	4	B	2	3	8	0	9	1	A	7	F	6	C	5	D

Table 2.14: The Differential factors of PICCOLO S-box

λ	μ
1	2
2	5

8. ***PRESENT:*** PRESENT was introduced by Andrey Bogdanov et al. in 2007 [18]. It is a 31-round SPN ultra-lightweight block cipher. The cipher works with 64-bit block size. PRESENT supports the key sizes 80-bit and 128-bit.

   Cihangir Tezcan et al. observed the differential cryptanalysis of the algorithm to improve the attack using differential factors. By the help of differential factors, they give a correction to the attack. The details are given in Section 2.5.

9. ***PRIDE:*** PRIDE was introduced by Martin R. Albrecht et al. in 2014 [18]. It is a 20-round SPN ultra-lightweight block cipher. The cipher works with 64-bit block size and supports 128-bit key size.

   Cihangir Tezcan et al. observed the differential cryptanalysis of the algorithm to improve the attack using differential factors. By the help of differential factors, they give a correction to the attack. The details are given in Section 2.7.

10. ***RECTANGLE:*** RECTANGLE was introduced by Wentao Zhang et al. in 2014 [88]. It is 25-round SPN block cipher. The cipher works with 64-bit block size and supports 80-bit and 128-bit key size. Cihangir Tezcan et al. observed the differential cryptanalysis of the algorithm to improve the attack using differential factors. By the help of differential factors, they give a correction to the attack. The details are given in Section 2.8.

11. ***SARMAL:*** SARMAL was introduced by Kerem Varici et al. as a candidate algorithm for SHA-3 [78]. All operations of the algorithm work on 64-bit word

and the digest size of the algorithm either 224, 256, 384 or 512 bits. The S-box of SARMAL is generated from 4 different 4×4 S-boxes. We do not give these S-boxes in this work, we just give their differential factors in Table 2.15.

Since it is a hash function, we don't observe it in terms of differential factors.

Table 2.15: The Differential factors of SARMAL S-box S_2

λ	μ
F	4
A	9

12. ***SERPENT:*** SERPENT was introduced by Eli Biham et al. in 1998 as a candidate algorithm for AES [12]. It is an SPN-type block cipher with 128-bit block size and supports 0 to 256-bit key size. The cipher works with 32-round. In the substitution layer of SERPENT, 32 different 4×4 S-boxes are used. We do not give these S-boxes in this work, we just give their differential factors in Table 2.22.

    Cihangir Tezcan observed the differential cryptanalysis of the algorithm to improve the attack using differential factors. By the help of differential factors, he gave a correction to the attack. The details are given in Chapter 2.6.

13. ***SPONGENT:*** SPONGENT was introduced by Bogdanov et al. [17]. In the substitution layer of SPONGENT, a 4×4 S-box is used. Table 2.16 and in Table 2.17 shows S-box and its differential factors, respectively.

    Since it is a hash function, we don't observe it in terms of differential factors.

Table 2.16: The S-box of SPONGENT

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s(x)$	E	D	B	0	2	1	4	F	7	A	8	5	9	C	3	6

Table 2.17: The Differential factors of SPONGENT S-box

λ	μ
F	9
1	F

14. ***TWOFISH:*** TWOFISH was introduced by Bruce Schneier et al. in 1998 as a candidate algorithm for AES[62]. It is a 16-round Feistel network. The algorithm works with 128-bit block size. 128 to 256-bit key size are supported in TWOFISH. In substitution layer of TWOFISH, 4 different 8×8 S-boxes are used. The 4 different S-boxes are derived from by using permutations and key material. We do not give these S-boxes in this work, we just give their differential factors in Table 2.18.

    We study on the differential-like cryptanalysis of TWOFISH to improve the attack using differential factors. But the structure of the algorithm is not suitable to use the differential factors. The reason is that the key XOR is not used before the S-box operation. Therefore, we do not give any correction or improvements to the attack by using differential factors.

Table 2.18: The Differential factors of TWOFISH S-boxes

S-box	λ	μ
q_0, t_1	6	9
q_1, t_2	5	B

2.5 An Example of Differential Cryptanalysis: Differential Cryptanalysis of PRESENT

PRESENT was introduced by Andrey Bogdanov et al. in 2007 [18]. It is a 31-round SPN ultra-lightweight block cipher with 64-bit block size and supports 80-bit and 128-bit key size. The cipher is described in Figure 2.1.

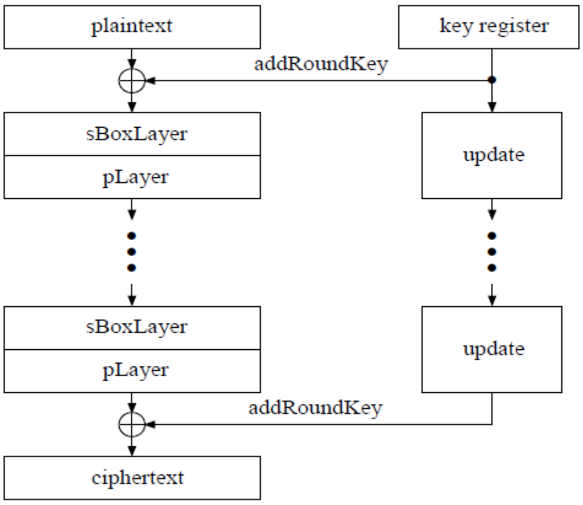

Figure 2.1: 31-round PRESENT Encryption Algorithm.

Each round of PRESENT described in Figure 2.2 consists of an XOR operation, a permutation, and a substitution layer. In substitution layer of PRESENT, a 4×4 S-box is used and shown in Table 2.19.

Table 2.19: The S-box of PRESENT

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s(x)$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

The difference distribution table (DDT) is shown in Table 2.20. The differential uniformity is 4 that is the highest values in DDT.

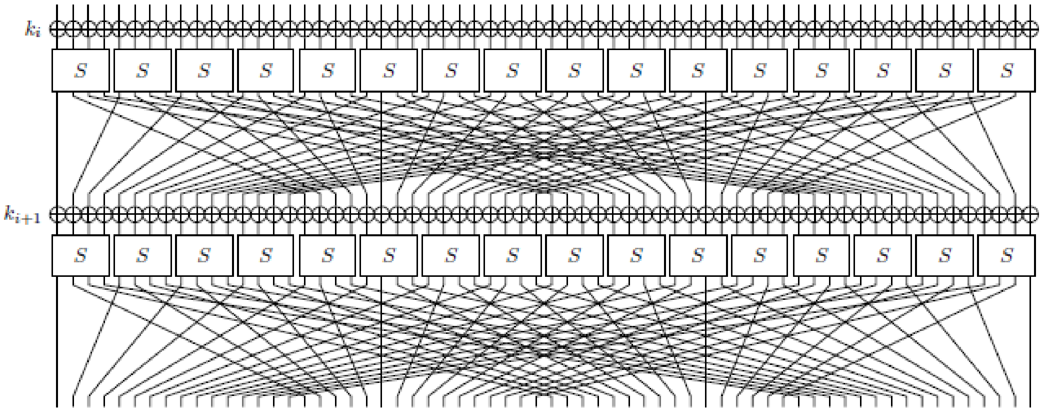

Figure 2.2: Round Function of PRESENT

Table 2.20: The Difference Distrubution Table of the PRESENT S-box

ΔX \ ΔY	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	4	0	0	0	4	0	4	0	0	0	4	0	0
2	0	0	0	2	0	4	2	0	0	0	2	0	2	2	2	0
3	0	2	0	2	2	0	4	2	0	0	2	2	0	0	0	0
4	0	0	0	0	0	4	2	2	0	2	2	0	2	0	2	0
5	0	2	0	0	2	0	0	0	0	2	2	2	4	2	0	0
6	0	0	2	0	0	0	2	0	2	0	0	4	2	0	0	4
7	0	4	2	0	0	0	2	0	2	0	0	0	2	0	0	4
8	0	0	0	2	0	0	0	2	0	2	0	4	0	2	0	4
9	0	0	2	0	4	0	2	0	2	0	0	0	2	0	4	0
A	0	0	2	2	0	4	0	0	2	0	2	0	0	2	2	0
B	0	2	0	0	2	0	0	0	4	2	2	2	0	2	0	0
C	0	0	2	0	0	4	0	2	2	2	2	0	0	0	2	0
D	0	2	4	2	2	0	0	2	0	0	2	2	0	0	0	0
E	0	0	2	2	0	0	2	2	2	2	0	0	2	2	0	0
F	0	4	0	0	4	0	0	0	0	0	0	0	0	0	4	4

In [79], the 16-round differential attack is mounted by adding two rounds to the bottom of the 24 different 14-round differentials which hold with probability 2^{-62}. One of these differentials is

$$\Delta_1 = 0700000000000700 \rightarrow_{14r} 0000000900000009$$

In 15^{th} round, the S-boxes S_0 and S_8, in 16^{th} round S_4, S_6, S_8, S_{10}, S_{12} and S_{14} are activated by the output difference of the characteristics. Since 8 S-boxes are activated, the authors claimed that 32-bits of the key can be captured in this attack. For this attack, the time complexity of 2-round PRESENT encryptions is $2^{33.18}$, the data complexity is 2^{64} chosen plaintexts, the memory complexity is 2^{32} 6- bit counters. Time complexity of getting the 48 remaining bits via exhaustive search is 2^{48} 16-round PRESENT encryptions.

2.5.1 Corrected Attack on PRESENT

Cihangir Tezcan observed differential cryptanalysis of PRESENT [79] to improve the attack using differential factors. By the help of differential factors, which is shown in Table 2.21 , he gave a correction to the attack in [77] and [74].

Table 2.21: The Differential Factors of PRESENT S-box

λ	μ
1	5
F	F

The input difference of 16^{th} round is 1. As mentioned in 2.2, when $\mu = 1$, there exists a differential factor $\lambda = 5$. The input difference of activated six S-boxes in 16^{th} round coincides with $\mu = 1$. Therefore, the advantage of the attack is decreased to 26-bits from 32-bits. Thus, the new time complexity of 2-round PRESENT encryptions is $2^{27.18}$ and time complexity of getting the 52 remaining bits is 2^{52} 16-round PRESENT encryptions. Moreover, Tezcan gives a new correction for this attack by using undisturbed bits. The detailed information can be found in [77], [74] and [76].

2.6 An Example of Differential Cryptanalysis: Differential Cryptanalysis of SERPENT

SERPENT was introduced by Eli Biham et al. in 1998 as a candidate algorithm for AES [12]. It is a 32-round SPN type block cipher with 128-bit block size and supports 0 to 256-bit key size. In the substitution layer of SERPENT, 32 different 4×4 S-boxes are used. We do not give these S-boxes in this work. The cipher contains:

- An initial permutation IP

- A key mixing operation, a substitution transformation, and a linear transformation exist in each round. Instead of this linear transformation, an extra operation is used to mix key in the last round.

  - ***Key Mixing:*** This is a basic XOR operation between a 128-bit round key and current intermediate data B_i.
  - ***Substitution transformation:*** The S-box is applied to four 32-bit words X_0, X_1, X_2, X_3, and the result is four output words. 32 copies of the S-box are executed at the same time,
  - ***Linear transformation:*** To mix 32-bit words X_0, X_1, X_2, X_3 linearly, follow these steps,

$$\begin{aligned}
X_0, X_1, X_2, X_3 &= S_i(B_i \oplus K_i) \\
X_0 &= X_0 <<< 13 \\
X_2 &= X_2 <<< 3 \\
X_1 &= X_1 \oplus X_0 \oplus X_2 \\
X_3 &= X_3 \oplus X_2 \oplus (X_0 << 3) \\
X_1 &= X_1 <<< 1 \\
X_3 &= X_3 <<< 7 \\
X_0 &= X_0 \oplus X_1 \oplus X_3 \\
X_2 &= X_2 \oplus X_3 \oplus (X_1 << 7) \\
X_0 &= X_0 <<< 5 \\
X_2 &= X_2 <<< 22 \\
B_{i+1} &= X_0, X_1, X_2, X_3
\end{aligned}$$

- A final permutation FP

128-bit plaintext P gives $\tilde{B}_0$ after the initial permutation. $\tilde{B}_0$ is also the input to the first round. $\tilde{B}_1$ and $\tilde{B}_2$ represent the outputs of the 1^{st} and 2^{nd} round and this continues until the last round output $\tilde{B}_{32}$. The final permutation is applied to $\tilde{B}_{32}$ and it gives the ciphertext C. Each $\tilde{B}_i$, $i\in \{1, ..., 31\}$ contains four 32-bit words X_0, X_1, X_2, X_3 where X_0 is the leftmost word.

The cipher can be formally described by;

$$\begin{aligned}
X_0, X_1, X_2, X_3 &= S_i(\tilde{B}_i \oplus K_i) \\
\tilde{B}_0 &= IP(P) \\
\tilde{B}_{i+1} &= R_i(\tilde{B}_i) \\
C &= FP(\tilde{B}_r)
\end{aligned}$$

where

$$\begin{aligned}
P &= \tilde{B}_0 \\
C &= \tilde{B}_{32} \\
R_i(X) &= L(\tilde{S}_i(X \oplus \tilde{K}_i)) \quad \text{where} \quad i = 0, ..., r-2 \\
R_i(X) &= \tilde{S}_i(X \oplus \tilde{K}_i) \oplus \tilde{K}_r \quad \text{where} \quad i = r-1
\end{aligned}$$

$\tilde{S}_i$, L denote an application of S-box of S_i and Linear Transformation, respectively.

In [32], the differential-linear attack is mounted for 10, 11, and 12 rounds for the key sizes 128, 192, 256, respectively. The combination of 3-round differential and 6-round linear approximation is used to perform the attack. While former is satisfied with probability 2^{-7}, latter has bias $q = 2^{-27}$. The 3-round differential

Δ : 00000000000000000000000040050000 $\rightarrow$ 0??00?000?000000000?00?0??0??0?0

The 6-round linear approximation

Λ : 20060040000001001000000000000000 $\rightarrow$ 00001000000000005000010000100001

By appending one round to head and one round to bottom of the 9-round distinguisher and one round to the bottom, the 11-round attack is performed. To mount the 12-round attack, again one round is appended to head. But, time complexity of the 11-round attack is more higher than time complexity of exhaustive search of 128 bits. Therefore, to perform the 10-round attack the last round of the distinguisher is removed.

2.6.1 Corrected Attack on SERPENT

Cihangir Tezcan observed differential cryptanalysis of SERPENT [32] to improve the attack using differential factors. By the help of differential factors which is shown in Table 2.22, they give a correction to the attack in [74].

Table 2.22: The Differential factors of SERPENT S-boxes

S-box	λ	μ
S_0	4	4
S_0	D	F
S_1	4	4
S_1	F	E
S_2	2	1
S_2	4	D
S_6	6	2
S_6	F	F

Since the input difference of activated S-boxes S_0 and S_2 coincides with two differential factors for the output differences 4 and E, the attacker can not capture all bits of the attacked key. As mentioned in [77], the advantage of the attack is decreased to 38, 46 and 156 bits from 40, 48 and 160 bits, respectively. He reduces the time complexity with factor of 4, 4 and 8, respectively. Moreover, he improves these attacks in [74] by using Theorem 2.5. Since key space is reduced, data complexities are also reduced from $2^{101.2}$, $2^{121.8}$ and $2^{123.5}$ to $2^{100.55}$, $2^{120.8}$ and $2^{122.45}$.

2.7 An Example of Differential Cryptanalysis: Differential Cryptanalysis of PRIDE

PRIDE was introduced by Martin R. Albrecht et al. in 2014 [4]. It is a 20-round SPN ultra-lightweight block cipher with 64-bit block size and supports 128-bit key size. The round function $\mathcal{R}$ of the algorithm contains key addition, substitution, and linear layers. In substitution layer of PRIDE, 16 identical 4×4 S-boxes are used and shown in Table 2.23. The last round $\mathcal{R}^{'}$ ends after the substitution layer. Linear layer is not used in the last round. Overall structure of the algorithm is described in Figure 2.3.

The round function $\mathcal{R}$ has an ordinary substitution-permutation network structure. Firstly, the current state is XORed with subkey, then result state is split into 16 4-bit nibbles and fed into S-boxes. Finally, linear layer is applied to the state to permute

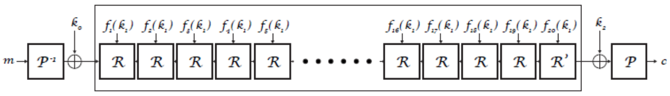

Figure 2.3: Overall structure of PRIDE.

and process. The inside of the round function is shown in Figure 2.4.

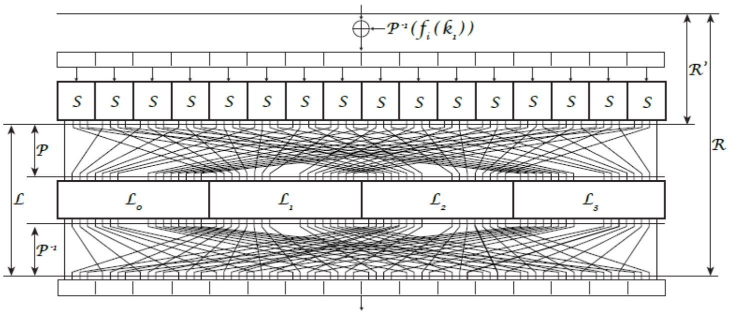

Figure 2.4: Round Function of PRIDE.

Three sub-layers form the linear layer $\mathcal{L}$ of PRIDE.

- A matrix layer $\mathcal{M}$,
- A permutation layer $\mathcal{P}$
- A permutation layer $\mathcal{P}^{-1}$ that is inverse of $\mathcal{P}$.

Table 2.23: The S-box of PRIDE

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s(x)$	0	4	8	F	1	5	E	9	2	7	A	C	B	D	6	3

2.7.1 18-round Differential Attack on PRIDE

The authors first observe the DDT of S-box of PRIDE and they notice that output difference $0x8$ occurs with probability 2^{-4} if given input difference is $0x8$ in [89]. Depending on these differences, they find 2-round differential characteristic. They calculate the probability of the characteristic as 2^{-8}. To obtain 15-round differential characteristic, authors iterate the 2-round differential characteristic for 7 times. Finally, they append a round below 14-round differential characteristic. The 15-round differential characteristic holds with probability 2^{-58}.

The 18-round differential attack is mounted by appending a round to the head and two rounds to the bottom of the 15-round a differential characteristic. This differential is

$$\Delta_1 = 0800000000000000 \rightarrow 0000080008008000$$

The data, time and memory complexities of the attack are 2^{60} chosen plaintexts, 2^{66} encryptions, and 2^{64} bytes, respectively. The success probability of the attack is approximately 61%.

2.7.1.1 Corrected 18-round Differential Attack on PRIDE

Cihangir Tezcan et al. observed differential cryptanalysis of PRIDE [89] to improve the attack using differential factors. By the help of differential factors, which is shown in Table 2.25, they give a correction to the attack in [76]. They summarize the attack in Table 2.24.

Table 2.24: 18 Round Differential Attack of PRIDE. Differential factors are given in bold [76].

	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
ΔI_1	0000	0000	0000	0000	0000	????	0000	0000	0000	????	0000	0000	0000	????	0000	0000
ΔX_1	0000	0000	0000	0000	0000	????	0000	0000	0000	????	0000	0000	0000	????	0000	0000
ΔY_1	0000	0000	0000	0000	0000	**1000**	0000	0000	0000	**1000**	0000	0000	0000	**1000**	0000	0000
ΔZ_1	0000	0100	0100	0100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔW_1	0100	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔI_2	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
	15-Round Differential Δ_1															
ΔX_{17}	0000	0000	0000	0000	0000	**1000**	0000	0000	0000	**1000**	0000	0000	0000	**1000**	0000	0000
ΔY_{17}	0000	0000	0000	0000	0000	????	0000	0000	0000	????	0000	0000	0000	????	0000	0000
ΔZ_{17}	0000	0?00	0?00	0?00	0000	0?00	0?00	0?00	0000	0?00	0?00	0?00	0000	0?00	0?00	0?00
ΔW_{17}	0?00	0?00	0?00	0?00	00?0	???0	0??0	0??0	???0	00?0	0??0	0??0	0?00	0?00	0?00	0?00
ΔI_{18}	00?0	?0??	0??0	0000	0?00	??0?	0??0	0000	0000	????	0???	0000	0000	????	0?00	0000
ΔX_{18}	00?0	?0??	0??0	0000	0?00	??0?	0??0	0000	0000	????	0???	0000	0000	????	0?00	0000
ΔY_{18}	????	????	????	0000	????	????	????	0000	0000	????	????	0000	0000	????	????	0000
ΔO_{18}	????	????	????	0000	????	????	????	0000	0000	????	????	0000	0000	????	????	0000

Since 16 S-boxes are activated, the authors claim that they perform the attack with 2^{66} 18-round encryptions to capture 64-bits of the key. In addition, exhaustive search is used to obtain remaining 64 key bits with 2^{64} 18-round PRIDE encryptions.

Since the input difference of activated S-boxes coincides with 6 differential factors that are given as a bold in Table 2.24, the correction for the attack is given in [76]. As mentioned in [76], the advantage of the attack is decreased to 58-bits from 64-bits. 6 bits of the key can not be obtained by the attacker. Thus, new time complexity of key guessing is 2^{60} and time complexity of getting the 70 remaining bits is 2^{70} 18-round PRIDE encryptions. The overall time complexity of 18-round PRIDE encryptions is 2^{70}, not 2^{66}.

Table 2.25: The Differential factors of PRIDE S-box

λ	μ
1	1
8	8

2.7.2 19-round Differential Attack on PRIDE

The paper [86], authors find 56 iterative differential characteristics by applying the automatic search methods [70], [71] and give 24 one round iterative characteristics between 56 iterative differential characteristics. 15-round characteristic is found by the authors with probability 2^{-60}. Two rounds are appended to head and bottom of 15-round characteristic that corresponds ΔY_2 in Table 2.26 to mount 19-round differential attack. The authors claimed that the 68-bits of the key can be captured with the time complexity of 2^{63} encryptions in this attack. In addition, exhaustive search is used to obtain remaining 60 key bits with 2^{60} PRIDE encryptions. Complexities of the data, time and memory equal to 2^{62}, 2^{63} and 2^{71}, respectively.

2.7.2.1 Corrected 19-round Differential Attack on PRIDE

Cihangir Tezcan et al. observed differential cryptanalysis of PRIDE [86] to improve the attack using differential factors. By the help of differential factors, which is shown in Table 2.25, they give a correction to the attack in [76]. Table 2.26 summarize the attack.

Table 2.26: 19 Round Differential Attack of PRIDE. Differential factors are given in bold [76].

	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
ΔI_1	????	????	????	0000	????	0000	????	0000	????	????	0000	0000	????	????	0000	0000
ΔX_1	????	????	????	0000	????	0000	????	0000	????	????	0000	0000	????	????	0000	0000
ΔY_1	?00?	00?0	00?0	0000	?00?	0000	00?0	0000	?0??	00?0	0000	0000	?00?	00?0	0000	0000
ΔZ_1	?000	?000	?000	?000	0000	0000	0000	0000	0??0	00?0	??00	0?00	?000	?000	?000	?000
ΔW_1	0000	?000	?000	0000	0000	0000	0000	0000	0000	?000	?000	0000	0000	?000	?000	0000
ΔI_2	0000	0000	0000	0000	?0??	0000	0000	0000	?0??	0000	0000	0000	0000	0000	0000	0000
ΔX_2	0000	0000	0000	0000	?0??	0000	0000	0000	?0??	0000	0000	0000	0000	0000	0000	0000
ΔY_2	0000	0000	0000	0000	**1000**	0000	0000	0000	**1000**	0000	0000	0000	0000	0000	0000	0000
ΔZ_2	0000	1000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔW_2	0000	1000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔI_3	0000	0000	0000	0000	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000
	15-Round Differential Δ_1															
ΔX_{18}	0000	0000	0000	0000	**1000**	0000	0000	0000	**1000**	0000	0000	0000	0000	0000	0000	0000
ΔY_{18}	0000	0000	0000	0000	?0??	0000	0000	0000	?0??	0000	0000	0000	0000	0000	0000	0000
ΔZ_{18}	0000	?000	?000	0000	0000	0000	0000	0000	0000	?000	?000	0000	0000	?000	?000	0000
ΔW_{18}	?000	?000	?000	?000	0000	0000	0000	0000	??00	000?	??00	0000	?000	?000	?000	?000
ΔI_{19}	?0??	00?0	0000	0000	?00?	0000	0000	00?0	?0??	00?0	0000	0000	?00?	0000	0000	0000
ΔX_{19}	?0??	00?0	0000	0000	?00?	0000	0000	00?0	?0??	00?0	0000	0000	?00?	0000	0000	0000
ΔY_{19}	????	????	0000	0000	????	0000	0000	????	????	????	0000	0000	????	0000	0000	0000
ΔO_{19}	????	????	0000	0000	????	0000	0000	????	????	????	0000	0000	????	0000	0000	0000

Since the input difference of activated S-boxes of 2^{nd} and 18^{th} round coincides with differential factors that are given as a bold in Table 2.26, the correction for the attack is given in [76]. As mentioned in [76], the advantage of the attack is decreased to 64-bits from 68-bits which requires 2^{59} PRIDE encryptions. The attacker can not capture the 4 bits of the key. Then, exhaustive search is applied to obtain remaining 64 key bits. Thus, new time complexity of 19-round PRIDE encryptions is 2^{59} and the time complexity of getting the 64 remaining bits is 2^{64} 19-round PRIDE encryptions. Therefore, the time complexity increased to 2^{64} from 2^{63} that means the overall time complexity is 2^{64} 19-round PRIDE encryptions, not 2^{63}.

2.7.3 20-Round Related-Key Differential Attack on PRIDE

In [28], the authors present two attacks to break fully PRIDE.

- *First Attack:* To mount 20-round attack, two rounds are appended to bottom of the 18-round related-key differential characteristics Δ_3 that holds with probability 2^{-36}. This differential is

  $$\Delta_3 = 8880000000000000 \to 8000800080000000$$

  The authors claimed that the 68-bits of the key can be captured with time complexity of 2^{41} encryptions. Exhaustive search is applied to obtain remaining 60 key bits with 2^{60} PRIDE encryptions. Then, the time complexity of the attack is 2^{60}.

- *Second Attack:* To mount 20-round attack, authors append two rounds to bottom and a round to head of the 17-round related-key differential characteristics Δ_4 that satisfies with probability 2^{-32} and Δ_4 equals to

  $$\Delta_4 = 8880000000000000 \to 8000800080000000 \to 0000000000000000$$

  The authors claimed that the 80-bits of the key can be captured with time complexity of $2^{53.7}$ encryptions using 17-round path. Then, exhaustive search is applied to obtain remaining 48 key bits with 2^{48} PRIDE encryptions. Then, the authors perform the attack with a time complexity with $2^{53.7}$.

2.7.3.1 Corrected 20-Round Related-Key Differential Attack on PRIDE

Cihangir Tezcan et al. observed differential cryptanalysis of PRIDE [28] to improve the attack using differential factors. By the help of differential factors, which is shown in Table 2.25, they give a correction to the attack in [76].

- *First Attack:* They summarize the attack in Table 2.27.

  Because of existence of differential factors which are shown bold in Table 2.27 in the attack, the correction for the attack is given in [76]. As mentioned in [76], the advantage of the attack is decreased to 67-bits from 68-bits which requires

Table 2.27: 20-Round Attack with 18-Round Differential. Differential factors are given in bold [31].

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
18-Round Differential Δ_3																
ΔI_{19}	1000	0000	0000	0000	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000
ΔX_{19}	0000	0000	0000	0000	0000	0000	0000	0000	**1000**	0000	0000	0000	0000	0000	0000	0000
ΔY_{19}	0000	0000	0000	0000	0000	0000	0000	0000	????	0000	0000	0000	0000	0000	0000	0000
ΔZ_{19}	0000	0000	?000	0000	0000	0000	?000	0000	0000	0000	?000	0000	0000	0000	?000	0000
ΔW_{19}	?000	?000	0000	?000	0000	0?00	0000	??00	0?00	0000	??00	0000	?000	?000	0000	?000
ΔI_{20}	?00?	00?0	0000	0000	?00?	0?00	0000	0000	00?0	00?0	0000	0000	??0?	0?00	0000	0000
ΔX_{20}	?00?	00?0	0000	0000	?00?	0?00	0000	0000	00?0	00?0	0000	0000	??0?	0?00	0000	0000
ΔY_{20}	????	????	0000	0000	????	????	0000	0000	????	????	0000	0000	????	????	0000	0000
$\oplus\Delta k_0$	????	????	0000	0000	????	????	0000	0000	????	????	0000	0000	????	????	0000	0000
ΔC	??00	??00	??00	??00	??00	??00	??00	??00	??00	??00	??00	??00	??00	??00	??00	??00

2^{40} encryptions. The attacker can not capture a bit of the key. Exhaustive search is applied to obtain remaining 61 key bits with 2^{61} PRIDE encryptions. Therefore, new time complexity of 20-round PRIDE encryptions is 2^{40} and the time complexity of getting the 61 remaining bits is 2^{61} 20-round PRIDE encryptions. Therefore, the time complexity increased to 2^{61} from 2^{60} that means the overall time complexity is 2^{61} 20-round PRIDE encryptions, not 2^{60}.

- *Second Attack:* They summarize the attack in Table 2.28.

Table 2.28: 20-Round Attack with 17-Round Differential. Differential factors are shown in bold [31].

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
ΔI_1	????	0000	0000	0000	????	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔX_1	????	0000	0000	0000	????	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔY_1	**1000**	0000	0000	0000	**1000**	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔZ_1	1000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔW_1	1000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔI_2	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
17-Round Differential Δ_4																
ΔI_{19}	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔX_{19}	**1000**	0000	0000	0000	**1000**	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔY_{19}	????	0000	0000	0000	????	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔZ_{19}	?000	?000	0000	0000	?000	?000	0000	0000	?000	?000	0000	0000	?000	?000	0000	0000
ΔW_{19}	?000	?000	?000	?000	?00?	?00?	?000	?000	?00?	?00?	?000	?000	?000	?000	?000	?000
ΔI_{20}	????	0000	0000	0??0	????	0000	0000	0??0	????	0000	0000	0000	????	0000	0000	0000
ΔX_{20}	????	0000	0000	0??0	????	0000	0000	0??0	????	0000	0000	0000	????	0000	0000	0000
ΔY_{20}	????	0000	0000	????	????	0000	0000	????	????	0000	0000	0000	????	0000	0000	0000
$\oplus\Delta k_0$	????	0000	0000	????	????	0000	0000	????	????	0000	0000	0000	????	0000	0000	0000
ΔC	?00?	?00?	?000	?000	?00?	?00?	?000	?000	?00?	?00?	?000	?000	?00?	?00?	?000	?000

Because of existence of the four differential factors which are shown bold in Table 2.28 in the attack, the correction for the attack is given in [76]. As mentioned in [76], the advantage of the attack is decreased to 76-bits from 80-bits which requires $2^{49.7}$ encryptions. The attacker can not capture the 4 bits of the key. Exhaustive search is applied to obtain remaining 52 key bits with 2^{52} PRIDE encryptions. Therefore, new time complexity of 20-round PRIDE encryptions is $2^{49.7}$ and the time complexity of getting the 52 remaining bits is 2^{52} 20-round PRIDE encryptions. Thus, time complexity of the attack is still $2^{53.7}$ but exhaustive search requires 2^{52} PRIDE encryptions.

2.8 An Example of Differential Cryptanalysis: Differential Cryptanalysis of RECTANGLE

RECTANGLE was introduced by Wentao Zhang et al. in 2014 [88]. It is an iterative 25-round SPN block cipher with 64-bit block size and supports 80-bit and 128-bit key size. In substitution layer of RECTANGLE, a 4×4 S-box is used and shown in Table 2.29.

Cipher state (CS) is a term to be used to express a plaintext or an intermediate result, or a ciphertext with 64-bit. It can be represented as 4×16 rectangular array of bits w_i.

$$\begin{bmatrix} w_{15} & w_{14} & w_{13} & \cdots & w_0 \\ w_{31} & w_{30} & w_{29} & \cdots & w_{16} \\ w_{47} & w_{46} & w_{45} & \cdots & w_{32} \\ w_{63} & w_{62} & w_{61} & \cdots & w_{48} \end{bmatrix}$$

Each round of the algorithm contains three steps:

- ***AddRoundkey:*** This step is a basic XOR operation between CS and round subkey.
- ***SubColumn:*** This step is an S-box application for each column of CS.
- ***ShiftRow:*** It is a left rotation operation of the last three rows of CS by 1, 12, and 13 bits, respectively.

There exists a final subkey XOR in the last round.

Table 2.29: S-box of RECTANGLE

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s(x)$	6	5	C	A	1	E	7	9	B	0	3	D	8	F	4	2

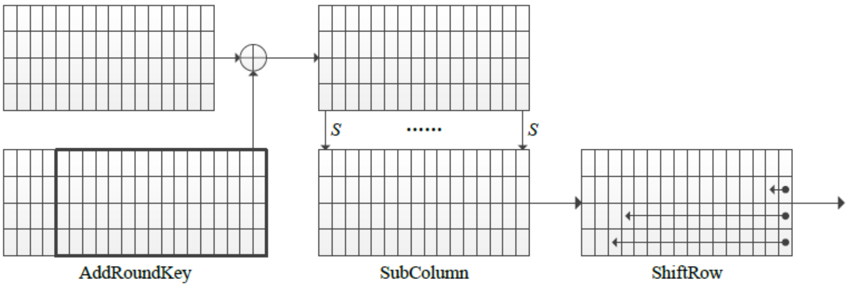

Figure 2.5: Round Transformation of RECTANGLE Encryption Algorithm.

The key schedule of the algorithm also contains three steps. For an 80-bit key, a key register is used to keep 80 bits and the key state (KS) can be represented as 5×16

matrix. Each entry of matrix is a bit of the key and it is denoted by v_i.

$$\begin{bmatrix} v_{15} & v_{14} & v_{13} & \cdots & v_0 \\ v_{31} & v_{30} & v_{29} & \cdots & v_{16} \\ v_{47} & v_{46} & v_{45} & \cdots & v_{32} \\ v_{63} & v_{62} & v_{61} & \cdots & v_{48} \\ v_{79} & v_{78} & v_{77} & \cdots & v_{64} \end{bmatrix}$$

A 64-bit round subkey K_i is constructed with bits from v_0 to $v_6$3 of current contents of KS. To update the key below steps are followed:

- This step is an S-box application. Input bits of S-box are the intersection of the first 4 rows and the last 4 columns in right,
- This step is a Feistel transformation for 1-round,
- This step is a simple XOR operation between a 5-bit round constant and 5-bit KS.

At the end, the updated KS is used to derive the last round key K_{25}. Round constants ($RC[i]$) are produced by using a 5-bit LFSR (Linear Feedback Shift Register).

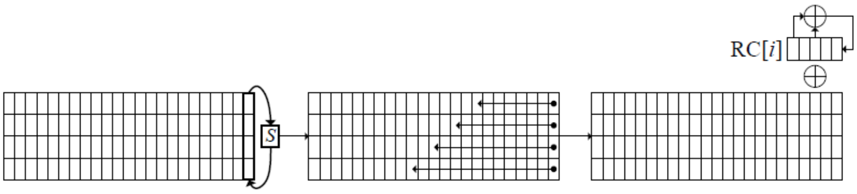

Figure 2.6: Key Schedule of RECTANGLE.

2.8.1 REC-0

REC-0 refers to initial design of RECTANGLE. The authors revised the key schedule and S-box, because of 19-round related-key differential attack [65] and software performance. The S-box of REC-0 is given in Table 2.30.

Table 2.30: The S-box of REC-0

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$s(x)$	9	4	F	A	E	1	0	6	C	7	3	8	2	B	5	D

The same three steps in 2.8 are also used in each round of REC-0 2.8. The key schedule of REC-0 is a bit different from RECTANGLE. REC-0 is also composed of three steps.

For an 80-bit key, a key register is used to keep 80 bits and KS can be represented as 4×20 matrix. Each entry of matrix is a bit of the key and it is denoted by v_i.

$$\begin{bmatrix} v_{19} & v_{18} & \cdots & v_0 \\ v_{39} & v_{38} & \cdots & v_{20} \\ v_{59} & v_{48} & \cdots & v_{40} \\ v_{79} & v_{78} & \cdots & v_{60} \end{bmatrix}$$

The rows of the KS is rotated to the left by 7, 9, 11 and 13 bits, respectively. A 5-bit LFSR is used to generate round constants.

2.8.2 19-Round Related-Key Differential Attack on REC-0

15-round differential characteristic is obtained by authors and satisfied with probability 2^{-64} [65]. They fixed input difference ΔI_2 on 2^{nd} round, output difference ΔO_{16} on 16^{th} round, and the differences ΔK_2 and ΔK_{16} of the 2^{nd} and the 16^{th} round keys. To mount 19-round related-key differential attack, 2 rounds are appended to head and bottom of characteristic.

They choose 2^x structures having plaintext pairs corresponding to ΔI_2 and ΔO_{16}. They found that the expected number of these plaintext pairs equals to $2^{x-24.5}$ and the expected number of other plaintext pairs to satisfy ΔO_{18} equals to $2^{x+34.54}$. Key guess part of the attack involves 4 steps.

- *Step 1:* 1^{st} round is encrypted partially in this step. This step is performed with a time complexity $2^{x+40.54}$, approximately.
- *Step 2:* 2^{nd} round is encrypted partially in this step. This step is performed with a time complexity $2^{x+39.54}$, approximately.
- *Step 3:* 18^{th} round is decrypted partially in this step. This step is performed with a time complexity $2^{x+38.54}$, approximately.
- *Step 4:* 17^{th} round is decrypted partially in this step. TThis step is performed with a time complexity $2^{x+28.54}$, approximately.

Total complexity of the 19-round reduced REC-0 is $2^{67.42}$. The memory complexity is 2^{72} key counters.

2.8.2.1 Corrected Attack on REC-0

Tezcan et al. observed the related key differential cryptanalysis of RECTANGLE [65] to improve the attack using differential factors. By the help of differential factors, which is shown in Table 2.33, they give a correction to the attack in [76].

They summarize the attack in Table 2.31.

Table 2.31: 19-round differential-linear attack of REC-0. Differential factors are given in bold [76].

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
$X_{0,O}$	0000	**0?00**	??00	?000	0000	000?	001?	0?10	**0?00**	?000	0000	0000	000?	0000	0000	0000
$X_{1,I}$	0000	0000	0000	0000	0000	??1?	????	0000	0000	0000	0000	0000	??0?	0000	0000	0000
$X_{1,O}$	0000	0000	0000	0000	0000	0001	0010	0000	0000	0000	0000	0000	0101	0000	0000	0000
15-Round Differential Δ_1																
$X_{17,I}$	0000	0000	0000	0000	0000	0000	0001	0000	0000	0000	0000	0000	0000	0000	0000	0100
$X_{17,O}$	0000	0000	0000	0000	0000	0000	????	0000	0000	0000	0000	0000	0000	0000	0000	?1??
$X_{18,I}$	0000	0000	?000	0100	0100	**00?0**	00*?	0000	000*	?000	0?00	0000	0000	0000	**00?0**	000?
$X_{18,O}$	0000	0000	????	?1??	?1??	????	????	0000	????	????	????	0000	0000	0000	????	????

Authors changed the S-box of RECTANGLE, they used its inverse in REC-0. That S-box has differential factor $\lambda = 4$ for $\mu = 2$. Table 2.33 and 2.32 shows differential factors of both version. The inverse property is explained in Theorem 2.2.

Table 2.32: The Differential factors of RECTANGLE S-box

λ	μ
2	4
E	C

Table 2.33: The Differential factors of REC-0 S-box

λ	μ
4	2
C	E

Theorem 2.1 can not be applied in this attack because differential factors are placed in two rounds below and above of the characteristic Δ_1. But, the authors examine how time complexity is affected by these differential factors. The impacts of these differential factors are analyzed with the help of Java codes. By applying following property for the first round, they decrease time complexity.

Property 2.1 ([76]). The differential factor $\lambda = 4$ for $\mu = 2$ flips the value of the bit that corresponds to $\mu = 2$. Namely, the second bits from the right of $S(x)$ and $S(y \oplus 4)$ are the same (similarly for $S(y)$ and $S(x \oplus 4)$).

Time complexity of the first key guessing step (on page 40) is reduced by a factor of 2^2 using this property. Due to Property 2.1, complements of some key bits in the second key guessing step are tried not to omit the correct key. Since there is no property similar to Property 2.1 in inverse of REC-0, they could not make any changes on the third and fourth key guessing step. They corrected the time complexities of these steps. The time complexities of steps of their modified attack are $2^{x+38.29}$, $2^{x+39.29}$, $2^{x+38.55}$, and $2^{x+28.54}$, respectively. Time complexity of the attack is decreased from $2^{67.42}$ to $2^{66.35}$ 19-round encryptions, if $x = 26$ is chosen as in [65]. They reduce the time complexity with factor of $2^{1.07}$. While this reduction seems to be little, they corrected this attack by using undisturbed bit. If the attack is tested practically by the authors, it could not be performed because of the undisturbed bit.

2.8.3 18-round Differential Attack on RECTANGLE

The designers made revisions on the key schedule and S-boxes of REC-0 so that RECTANGLE is not as insecure as previous version to related-key attacks. Therefore, the above attack is not valid anymore.

The authors found a 14-round difference propagation for the single key scenario with the probability $2^{-62.83}$ in [88]. This propagation is represented as in Table 2.34. To mount 18-round differential attack, designers used this 14-round characteristic in [88]. But, they did not give the exact details of the attack. The time complexities of data, memory and time are 2^{64} plaintext, 2^{72} key counters, and $2^{78.67}$ for 80-bit seed key and $2^{126.66}$ for 128-bit seed key 18-round encryption, respectively. Moreover, authors claim that the success probability is approximately 67.5%

Table 2.34: Input-output difference of the 14-round difference propagation. Differential factors are shown in bold.

Input difference of Round 0	Output difference of Round 13
00**0**0000000000000	000**0**000000000000
00**1**0000100000000	000**0**000000000010
00**0**0000100000000	000**1**000000000000
00**0**0000000000000	000**0**000000000000

2.8.3.1 Corrected Attack on RECTANGLE

Tezcan et al. observed the differential cryptanalysis of RECTANGLE [88] to improve the attack using differential factors.

Since RECTANGLE S-box has differential factors and these exist in 14-round characteristic, Tezcan et al. claim that the effects of differential factors on complexities should be analyzed. Although authors did not give any detail about attack, Tezcan et al. make an approximate calculation on the time complexities. If key size is 80-bit, time complexity of this attack can be between $2^{76.67}$ and $2^{80.67}$. If key size is 128-bit, time complexity can be between $2^{124.66}$ and $2^{128.66}$.

CHAPTER 3

OVERVIEW OF LBLOCK

3.1 LBLOCK

With the development of technology in each area, usage of small devices like RFID tags and sensor networking in different areas are increased when compared with traditional computers. Since capacitance of memory, capabilities of computing, and available power supply are very constraint in small devices, new security, and privacy concerns on these devices and the needs for lightweight cryptographic module become a current issue [58]. Implementation of conventional cryptographic standards generally is not suitable for constrained devices. To create and analyze of new lightweight primitives and protocols, and solve the efficient implementation problem, lightweight cryptography which is based on optimizing the trade-off between security, cost, and performance become prominent among academicians and cryptographers.

Joint Technical Committee 1 (JTC 1) [3] was constituted under organizations of the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC) to establish the terminologies and standards of IT. The properties of lightweight cryptography for both hardware and software platforms are described in a new lightweight cryptography standardization project [2]. Size of chip and energy consumption take an important role to evaluate the lightweight properties in hardware implementations. On the other hand, smaller codes and RAM are chosen in software implementations of lightweight applications [40]. Moreover, a lightweight cryptography project was initiated by NIST in 2013 and following this project the first Lightweight Cryptography Workshop was held in 2015 to discuss the need for dedicated lightweight cryptography standards, requirements of real-world applications of lightweight cryptography [52]. The International Association of Cryptologic Research (IACR) hold 1^{st} International Workshop on Lightweight Cryptography for Security and Privacy Workshop (LightSec) in 2011 to create a platform to discuss concerns and propose solutions to problems of lightweight cryptography. In 2016, the fifth LightSec was held.

Several lightweight block ciphers have been created for some targeted platforms. PRESENT [18] and CLEFIA [67] have been standardized as a lightweight block cipher. PRIDE [4] is optimized for 8-bit micro-controllers. PRINTcipher [44] is designed to use in integrated circuit (IC) printing. The design goal of SKINNY [10] is to compete with SIMON [9] with respect to hardware and software performances. The designers of PRINCE [19] optimize the cipher in terms of latency if it is implemented in hardware. SEA [69] and LED [35] are also examples of lightweight block ciphers and there are more examples in the literature.

Table 3.1 shows the results of implementation comparison in terms of hardware between LBLOCK and other lightweight block ciphers. The comparison is done by the designers of LBLOCK.

Table 3.1: The results of implementation comparison in terms of hardware of lightweight block ciphers [85].

Algorithm	Block Size	Key Size	Area #GE	Speed kbps @100Khz	Logic Process
XTEA	64	128	3490	57.1	0.13 μm
HIGHT	64	128	3048	188.2	0.25 μm
mCrypton	64	128	2500	492.3	0.13 μm
DES	64	56	2300	44.4	0.18 μm
DESXL	64	184	2168	44.4	0.18 μm
KATAN	64	80	1054	25.1	0.13 μm
KTANTAN	64	80	688	25.1	0.13 μm
PRESENT	64	80	1570	200	0.18 μm
LBLOCK	64	80	1320	200	0.18 μm

LBLOCK is a 32-round lightweight block cipher proposal at Applied Cryptography and Network Security Conference 2011 by Wenling Wu and Lei Zhang [85]. Some cryptanalysis of LBLOCK are given in Table 3.2

Table 3.2: Cryptanalysis on LBLOCK

Model	Attacks	Round	Time Complexity	Data Complexity	Reference
Single key	Differential	13	$2^{42.08}$	$2^{42.08}$	[50]
		17	$2^{67.52}$	$2^{59.75}$	[24]
	Boomerang	18	$2^{70.84}$	$2^{63.27}$	[24]
	Impossible Differential	20	$2^{72.7}$	2^{63}	[85]
		21	$2^{73.7}$	$2^{62.5}$	[50]
		21	$2^{69.5}$	2^{63}	[39]
		22	$2^{79.28}$	2^{58}	[39]
		23	$2^{75.36}$	2^{59}	[21]
		24	$2^{77.50}$	2^{59}	[81]
	Integral	22	$2^{70.54}$	2^{64}	[85]
		22	$2^{71.27}$	$2^{62.1}$	[61]
		22	2^{79}	2^{60}	[60]
	Zero-correlation linear	20	$2^{63.7}$	2^{64}	[68]
		20	$2^{39.6}$	$2^{63.6}$	[68]
		22	2^{70}	2^{61}	[68]
		23	2^{76}	$2^{62.1}$	[82]
	Biclique	32	$2^{78.4}$	2^{52}	[83]
		32	$2^{78.338}$	2^{2}	[6]
Related-key	Differential	22	2^{67}	$2^{63.1}$	[49]
	Impossible Differential	22	2^{70}	2^{47}	[54]
		23	$2^{78.3}$	$2^{61.4}$	[84]

3.1.1 Notations

For the simplicity and parallelism, we use exactly the same notations for LBLOCK [85].

- M: 64-bit plaintext
- C: 64-bit ciphertext
- K: 80-bit master key
- K_i: 32-bit round subkey
- F: round function
- s: 4×4 S-box
- S: S-box layer consists of eight s in parallel
- P, P_1: permutations operate on 32-bit
- $\oplus$: Bitwise exclusive-OR operation
- $<<<$ 8: 8-bit left cyclic shift operation
- $||$: Concatenation of two binary strings
- $[i]_2$: Binary form of an integer *i*

3.1.2 Specifications

LBLOCK was introduced by Wenling Wu and Lei Zhang. It is in Feistel network with 64-bit block size and 80-bit key size. It is composed of 32-round. Encryption scheme is depicted in Figure 3.1. Let a 64-bit plaintext is represented by $M = X_1||X_0$ where X_1 and X_0 have 32-bits. 32-bit left side X_1 is XORed with round key and the result fed into round function F. 32-bit right side X_0 is rotated by 8-bit to the left. The result of rotation and F is XORed. Result of this operation is an updated 32-bit left side X_2. An updated 32-bit right side X_3 is X_1. These processes continue until the last round. All of these steps construct encryption algorithm and it can be summarized as follows:

1. $X_i = F(X_{i-1}, K_{i-1}) \oplus (X_{i-2} <<< 8)$ where $2 \leq i \leq 33$
2. Output $X_{32}||X_{33}$ as the 64-bit ciphertext.

LBLOCK uses three main components in each round:

Round Function F: It is composed of confusion and diffusion layers and defined as follows:

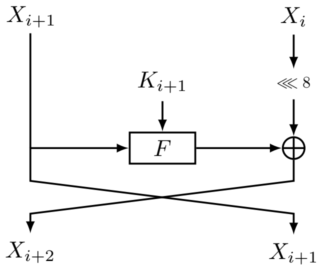

Figure 3.1: Encryption scheme of LBLOCK

$$\begin{aligned} F : \{0,1\}^{32} &\rightarrow \{0,1\}^{32} \\ (X, K_i) &\rightarrow U = P(S(X \oplus K_i)) \end{aligned}$$

Confusion function S**:** It is non-linear part of F. 8 different 4-bit × 4-bit S-boxes are used in S and represented as s_i.

$$\begin{aligned} S : \{0,1\}^{32} &\rightarrow \{0,1\}^{32} \\ I = I_7||I_6||I_5||I_4||I_3||I_2||I_1||I_0 &\rightarrow O = O_7||O_6||O_5||O_4||O_3||O_2||O_1||O_0 \\ O_i = s_i(I_i) &\quad \text{for} \quad i = 0, 1, \ldots, 7 \end{aligned}$$

Diffusion function P**:** It is the eight 4-bit permutation.

$$\begin{aligned} P : \{0,1\}^{32} &\rightarrow \{0,1\}^{32} \\ O = O_7||O_6||O_5||O_4||O_3||O_2||O_1||O_0 &\rightarrow R = R_7||R_6||R_5||R_4||R_3||R_2||R_1||R_0 \\ R_7 = O_6,\ R_6 = O_4, &\quad R_5 = O_7,\ R_4 = O_5, \\ R_3 = O_2,\ R_2 = O_0, &\quad R_1 = O_3,\ R_0 = O_1. \end{aligned}$$

Key schedule algorithm is used to create subkeys. **Key Scheduling:** LBLOCK uses 80-bit key master key denoted by $K = k_{79}k_{78}k_{77}k_{76}\ldots k_1k_0$. A subkey is the 32 leftmost bits of current key. For i is from 1 to 31, key is updated by,

1. $K <<< 29$
2. $[k_{79}k_{78}k_{77}k_{76}] = s_9[k_{79}k_{78}k_{77}k_{76}]$
   $[k_{75}k_{74}k_{73}k_{72}] = s_8[k_{75}k_{74}k_{73}k_{72}]$
3. $[k_{50}k_{49}k_{48}k_{46}] \oplus [i]_2$
4. The round key K_{i+1} is the 32 leftmost bits of current key.

8 different S-boxes are given in Table 3.3. The S-boxes S_8 and S_9 are the same S-boxes with S_0 and S_1, respectively.

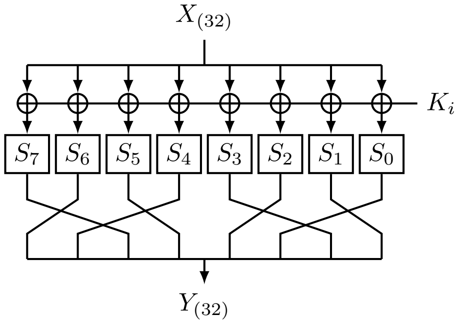

Figure 3.2: Round Function of LBLOCK

Table 3.3: S-boxes of LBLOCK

S_0	14, 9, 15, 0, 13, 4, 10, 11, 1, 2, 8, 3, 7, 6, 12, 5
S_1	4, 11, 14, 9, 15, 13, 0, 10, 7, 12, 5, 6, 2, 8, 1, 3
S_2	1, 14, 7, 12, 15, 13, 0, 6, 11, 5, 9, 3, 2, 4, 8, 10
S_3	7, 6, 8, 11, 0, 15, 3, 14, 9, 10, 12, 13, 5, 2, 4, 1
S_4	14, 5, 15, 0, 7, 2, 12, 13, 1, 8, 4, 9, 11, 10, 6, 3
S_5	2, 13, 11, 12, 15, 14, 0, 9, 7, 10, 6, 3, 1, 8, 4, 5
S_6	11, 9, 4, 14, 0, 15, 10, 13, 6, 12, 5, 7, 3, 8, 1, 2
S_7	13, 10, 15, 0, 14, 4, 9, 11, 2, 1, 8, 3, 7, 5, 12, 6
S_8	14, 9, 15, 0, 13, 4, 10, 11, 1, 2, 8, 3, 7, 6, 12, 5
S_9	4, 11, 14, 9, 15, 13, 0, 10, 7, 12, 5, 6, 2, 8, 1, 3

3.2 Differential Attacks on LBLOCK

Chen and Miyaji [24] applied the differential attack with the help of a single differential and multiple differentials. They found 15-round single differential path and mount 17-round attack based on that path. The best time complexity for breaking 17-round cipher was $2^{67.5211}$ given by the authors. To calculate the complexity for the multiple differential attack, authors used the formula which was given by Blondeau and Gerard [16].

Authors claimed that non-iterative differential path gave better result compared to iterative differential path for LBLOCK. While finding the differential paths, they performed truncated differential search. Since the property of diffusion layer of LBLOCK that permutation of internal bits is to be just between the S-boxes, they behave for each input of S-box as 0 (non-active S-box) or 1 (active S-box) to get r round characteristic. They first checked and then confirmed the smallest number of S-boxes up to 20-round [85]. After they accomplished to construct the structures by activating S-boxes ideally, differentials are derived from application of branch and bound algorithm for each structure. They just took all differential paths which have the probability greater than 2^{-72}. The largest probabilities corresponding to differential paths are summarized in Table 3.4.

As it is seen in the table, differential paths of the truncated form
$0000000011010000 \rightarrow 0001111000100100$ have the largest probability $2^{-61.2351}$.

Table 3.4: Best Differential Paths for 15-round [24].

Truncated Diff	Best Diff	$log_2(Prob)$	#Diff with Prob $< 2^{-64}$
0000000011010000 ↓ 0001111000100100	0000000011030000, 0003222000200100	-61.2351	1290
	0000000011030000, 0003422000200900		
	0000000011030000, 0003522000200900		
	0000000011030000, 0003622000200100		
	0000000011030000, 000b222000200100		
	0000000011030000, 000b422000200900		
	0000000011030000, 000b522000200900		
	0000000011030000, 000b622000200100		

From the given output truncated differential Δ_{15} $= (00011110, 00100100)$, the differentials in rounds 16 and 17 are calculated as

$$(00011110, 00100100) \rightarrow (11011011, 00011110)$$
$$\rightarrow (1 \star \star 11111, 11011011)$$

Since the S-boxes S_1, S_2, S_3, S_4 are active in round 16, the authors targeted to guess 16 bits of k_{16}. In round 17, since S_0, S_1, S_3, S_4, S_6, and S_7 are active, the authors targeted to guess 24 bits of k_{17}. They attacked to 40 bits of the key in total.

Results of authors are shown in Table 3.5.

Table 3.5: Size of the key candidate list, data complexity and computational cost with success probability 90% [24]

l	34	35	36	37	38
N	63.8294	63.4343	62.8884	61.9996	59.2471
$log(Time)$	74.2300	75.0917	76.0321	77.0087	78.0007

Notation for the key recovery algorithm [80]:

- m: the block size of the block cipher.
- k: the key size of the block cipher.
- $|\Delta_0|$: the number of differentials.
- Δ_0^i: i^{th} input difference.
- Δ_r: i^{th} output difference.
- p_i: the probability of the differential with input difference Δ_0^i.
- N_{st}: the number of structures is $2^{N_{st}}$.
- N_p: the number of plaintexts bits involved in the active S-boxes in the first round for all differentials.
- N_c: the number of ciphertexts bits involved in the non-active S-boxes in the last round deriving from Δ_r.

- β: the filtering probability for the ciphertext pairs.
- p_f : the filtering probability for the ciphertext pairs according to active S-boxes, $p_f = \beta \cdot 2^{N_c}$.
- l: the size of the candidate list.
- n_k: the number of guessed subkey bits in the last $R - r$ rounds.

The key recovery algorithm which is also used by the authors are summarized as follows:

Algorithm 1 Key Recovery Attack in the (multiple) differential cryptanalysis scenario

```
Input: $2^N$ plaintexts-ciphertexts pairs
Output: Master key $K$
1. Following steps should be done for each structure $2^{N_{st}}$,
    (a) All ciphertexts should be injected into a hash table indexed by $N_c$.
    (b) When an entry equals to $N_c$, it should be checked whether the input difference is in specified differentials by the number $|\Delta_0|$. If an input difference is satisfied by a pair, then move on to next step.
    (c) By using corresponding DDT, it should be verified whether input-output difference may occur in the last the round. If occur, move on to next step. This verification process is to be applied for the pairs in each entry.
    (d) Since ciphertext pairs must be decrypted to round $r$, $n_k$ should be guessed. It should be checked whether the result is the same with $\Delta_r$. If same, the corresponding counter should be increased by 1.
2. The best $l$ should be specified and key candidates should be chosen according to counters.
3. To determine whether or not a candidate key is the right master key, it should be tested. If not, the same process is applied to other candidates.
```

A random value is chosen and all inputs of S-boxes that are not active in the 1^{st} round is set to that value. On the other hand, N_p can obviously take all $2N_p$ possible values. These values are valid for each structure of $2^{N_{st}}$. The number of pairs must be $2^{N_{st}} \cdot 2^{N_p - 1} = 2^{N_{st}+N_p-1}$ for each differential. The expected number for satisfying Δ_r is to be $2^{N_{st}+N_p-1} \cdot \sum_{i=1}^{|\Delta_0|} p_i$ pairs. We call such pairs as right pairs.

Time complexities of (a), (b), (c), (d) and 3 is represented by T_a, T_b, T_c, respectively.

T_a : $2^{N_{st}+N_p}$ memory access;

T_b : $2^{N_{st}+2N_p-N_c}$memory access;

$T_c : |\Delta_0| \cdot 2^{N_{st}+2N_p-N_c}$ memory access;

$T_d : |\Delta_0| \cdot 2^{N_{st}+2N_p-N_c} \cdot p_f \cdot 2^{n_k}$ partial decryptions;

$T_3 : l \cdot 2^{k-n_k}$

Here, we have n_k independent subkey bits from the key schedule. Since $|\Delta_0| < 2^{N_p}$, $T_c < T_b$. The whole complexity;

$$T_a + T_b + T_c + T_d + T_3 \approx \begin{cases} T_a + T_3 & \text{if } N_p < N_c \\ T_b + T_3 & \text{if } N_p > N_c \\ 2T_a + T_3 = 2T_b + T_3 & \text{if } N_p = N_c \end{cases}$$

In Chen et.al attack, $N_p = 4 \times 3 = 12$ bits since the input difference was taken as (0000000011010000). Authors suppose that data complexity is equal to 2^N, then they found the number of structure $N_{st} = N - 12$ and each structure contain 2^{12} plaintexts. Therefore,

1. They first consider $2^{N-12} \cdot 2^{2\times 12-1} = 2^{N+11}$ pairs.

2. They know the nibbles e^2_{17} and e^5_{17} are non-active. Taking into account this condition, they inserted the ciphertexts into hash table. Memory cost and complexity of this process is equal to 2^N.

3. Before inserting the hash table authors must consider 2^{23} pairs. After inserting the hash table authors must consider $2^{23-8} = 2^{15}$ pairs.

4. They noticed that some pairs have impossible differentials by observing propagation of differentials in 2^{nd}-round and eliminated them. Due to $e^0_{17} = e^0_{16} = e^{10}_{15}$, $e^3_{17} = e^3_{16} = e^{13}_{15}$, $e^{11}_{17} = e^{13}_{16} = e^5_{15}$, $e^{12}_{17} = e^{14}_{16} = e^6_{15}$, they have $2^{15-16} = 2^{-1}$ pairs.

5. Since $e^1_{17} = e^1_{16} = S_4(e^3_{15})$, $e^4_{17} = e^4_{16} = S_2(e^5_{15})$, $e^6_{17} = e^6_{16} = S_3(e^4_{15})$, $e^7_{17} = e^7_{16} = S_1(e^6_{15})$, $e^9_{17} = e^3_{15} \oplus S_4(e^{13}_{15})$ and $e^{10}_{17} = e^4_{15} \oplus S_1(e^6_{15})$, by considering the differential table some differentials are not possible for the nibbles 1, 4, 6, 7, 9 and 10 of ciphertexts. They found the average probability of all S-boxes $P_{S_0} \approx P_{S_1} \approx P_{S_2} \approx ..... P_{S_7} \approx 0.4267$.

6. With this filter, for each structure $2^{-1} \cdot (0.4267)^6 = 2^{-8.37}$ pairs left and $2^{N-12-8.37} = 2^{N-20.37}$ pairs in total.

7. The computational cost equals to $2^{N-20.37}$ when they check for each of above pairs whether or not it is possible to see the corresponding input differences. Then $2^{N-20.37-12} = 2^{N-32.37}$ pairs left.

8. The ciphertext pairs are decrypted to control whether the result and $|\Delta_{15}|$ are matched. This decryption is executed by guessing 40-bits in 16^{th} and 17^{th}. This step is performed with time complexity $2^{N-32.37} \times 2^{40} = 2^{N+7.63}$

9. The time complexity of the searching candidate list is equal to 2^{40+l} since the given key candidate list 2^l.

10. $2^N + 2^{N-20.37} + 2^{N+7.63} + 2^{40+l}$ is found as a total complexity.

To find the relations between l, N, and computational complexity, they used the formula;

$$2^N = -4 \cdot \frac{\ln(2\sqrt{\pi}2^l 2^{-n_k})}{|\Delta_0| \cdot D(p_* || p)}$$

Here D implies the Kullback-Leibler divergence. They took $p_* = 2^{-61.2351}$ which is the best probability path and the success probability is as high as 90%.

3.3 Our Correction to the Attack using Differential Factors

Authors give the 8 different differential paths in Table 3.4. Firstly, we investigate these paths that after one round-it is actually 16^{th}round- whether the output of any S-box corresponds one of the differential factors of that S-box. If corresponds, we calculate the effect of that differential factors on time complexity. We do not mention all 8 paths in this paper, we just give details for the path 5 (000b2220, 00200100) $\overset{1round}{\rightarrow}$ (11011011, 00011110) as an example.

Table 3.6: Differential Factors of LBLOCK's S-boxes

S-box	λ	μ
S_0, S_8	B	1
S_0, S_8	3	4
S_1, S_6, S_7, S_9	3	2
S_1, S_6, S_7, S_9	3	4
S_2	3	1
S_2	B	2
S_3	B	1
S_3	3	8
S_4, S_5	B	1
S_4, S_5	3	2

We summarize the 16-round attack in Table 3.7.

Table 3.7: 16-round differential attack of LBLOCK for the 5^{th} path. Differential factors may occur in bold locations.

	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
15-Round Differential Δ_{15}																
$X_{16,I}$	0000	0000	0000	1011	0010	0010	0010	0000	0000	0010	0000	0000	0001	0000	0000	0100
$X_{16,O}$	0010	**0???**	0000	0001	**??1?**	0000	**???1**	**??1?**	0000	0000	0000	1011	0010	0010	0010	0000

For the path 5;

$$\Delta 15 = (000b2220, 00200100) \overset{1round}{\rightarrow} \Delta 16 = (2 * 01 * 0 * *, 000b2220).$$

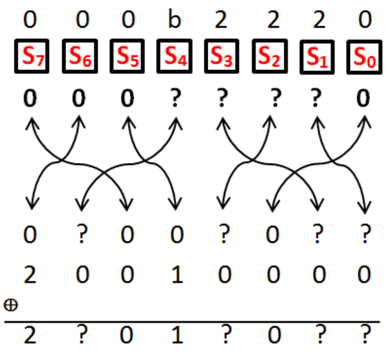

Figure 3.3: Encryption scheme of LBLOCK for the path 5^{th}

The mean of *'s of the output is that these locations must be active. Since these S-box are activated in this round, the attackers claimed that they can capture 16 bits of k_{16}. By taking consideration this information, we observe input-output differences of S_4, S_3, S_2, and S_1 that are activated S-boxes. We find that the output differences of S_4, S_3, S_2, and S_1 can take 4, 6, 6 and 6 different values, respectively. We notice that one of the four and six different output differences has differential factor property for S_4, S_2, and S_1. That means if the output difference is equal to one of the value μ in Table 3.6, it is unnecessary to try half of the keys. Thus, attacker can take advantage $2^{0.12553}$ for S_3, S_2, and S_1, separately and $2^{0.19265}$ for S_4. The overall advantage of this path is $2^{0.56924}$.

Table 3.8: Difference Distribution Table of Activated S-boxes for the path 5^{th}

	S_4	S_3	S_2	S_1
Input differences	b	2	2	2
Output differences	$\boxed{1}$	$\boxed{1}$	$\boxed{2}$	$\boxed{2}$
	4	3	6	3
	6	5	10	7
	7	7	b	a
		d	e	b
		f	f	f

The red boxes in Table 3.8 show the value μ corresponding S-box.

It is important to emphasize Theorem 2.1 because we use this theorem to calculate advantages.

Theorem 2.1. Let us assume that we have input pair (x, y), the partial subkey k and an S-box S containing a differential factor λ for an output difference μ in a block cipher based algorithm. We know that a subkey and an intermediate value of the message are generally XORed right before the S-box operation so that they form the input of S-box. If the partial subkey k and the input pair (x, y) give the output difference μ, then input pair (x, y) and the partial subkey $k \oplus \lambda$ would also give the same output difference μ. Thus, we can think that the attacker can not detect a bit of the partial subkey which corresponds to the output difference μ. the advantage of the cryptanalyst is reduced by 1 bit since a bit can not be detected. Hence, the time complexity of this key guess step is halved.

In summary, we pursue following steps;

1. Find activated S-boxes on differential path and create DDTs of activated S-boxes,
2. Determine whether these S-boxes have differential factor,
3. If the answer yes for previous step, determine whether these differential factors are used in differential path,
4. If the answer yes for previous step, calculate advantages of corresponding S-boxes by applying Theorem 2.1.

For the rest, we explain each step for each activated S-box.

Input difference 2 of S_1 can result in 6 different output difference. One of these output differences is 2 which also congruence the value μ for S_1 according to Table 3.6. In other words, if output difference of S_1 is 2 in the differential attack, the attacker can not capture the key bits of corresponding S-box S_1. Since the attacker has four active S-boxes, attacker try to capture 4-bits of round key corresponding to S_1. However, S_1 has a differential factor for $\mu = 2$ so that attacker tries 2^3 candidate keys instead of 2^4 due to Theorem 2.1. We must consider all output differences to calculate the advantage of S_1.

We have an input difference 2 for the S_1. Suppose that the output difference of S_1 is 3. Since there is no differential factor for $\mu = 3$, the attacker does not have an advantage. Therefore, attacker must try 2^4 key possibilities to capture the corresponding key bits for the S_1. This situation is also the same for the output differences 7, 10, 11, and 15. But if the output difference of S_1 is 2, there exists a differential factor for $\mu = 2$. Because of the Theorem 2.1, it is unnecessary to try half of the keys. Therefore, attacker must try 2^3 key possibilities to capture the corresponding key bits for the S_1. Since we do not exactly know output differences, advantage calculation must depend on all output differences. We find the mean of all advantages.

In the following equation, 2^4 is the advantage for the output differences 3, 7, 10, 11, 15 and 2^3 is the advantage for the output difference 2. We find these advantages by using Theorem 2.1. The advantage calculation for the S_1, we calculate the mean of 6 advantages;

$$\frac{2^4 + 2^4 + 2^4 + 2^4 + 2^4 + 2^3}{6} = 2^3 \cdot \frac{11}{6} = 2^3 \cdot 2^{0.87447}$$

$$2^{4-3.87447} = 2^{0.12553}$$

Input difference 2 of S_2 can result in 6 different output difference. One of these output differences is 2 which also congruence the value μ for S_2 according to Table 3.6. In other words, if output difference of S_2 is 2 in differential attack, the attacker can not capture the key bits of corresponding S-box S_2. Since the attacker has four active S-boxes, attacker try to capture 4-bits of round key corresponding to S_2. However, S_2 has a differential factor for $\mu = 2$ so that attacker tries 2^3 candidate keys instead of 2^4 due to Theorem 2.1. We must consider all output differences to calculate the advantage of S_2.

We have an input difference 2 for the S_2. Suppose that the output difference of S_2 is 3. Since there is no differential factor for $\mu = 3$, the attacker does not have an advantage. Therefore, attacker must try 2^4 key possibilities to capture the corresponding key bits for the S_2. This situation is also the same for the output differences 7, 10, 11, and 15. But if the output difference of S_2 is 2, there exists a differential factor for $\mu = 2$. Because of the Theorem 2.1, the attacker does not need to try half of the keys. Therefore, attacker must try 2^3 key possibilities to capture the corresponding key bits for the S_2. Since we do not know the output difference exactly, the advantage calculation must depend on all output differences. We find the mean of all advantages.

In the following equation, 2^4 is the advantage for the output differences 3, 7, 10, 11, 15 and 2^3 is the advantage for the output difference 2. We find these advantages by using Theorem 2.1. The advantage calculation for the S_2, we calculate the mean of 6 advantages;

$$\frac{2^4+2^4+2^4+2^4+2^4+2^3}{6} = 2^3 \cdot \frac{11}{6} = 2^3 \cdot 2^{0.87447}$$
$$2^{4-3.87447} = 2^{0.12553}$$

Input difference 2 of S_3 can result in 6 different output difference. One of these output differences is 1 which also congruence the value μ for S_3 according to Table 3.6. In other words, if output difference of S_3 is 1 in differential attack, the attacker can not capture the key bits of corresponding S-box S_3. Since the attacker has four active S-boxes, attacker try to capture 4-bits of round key corresponding to S_3. However, S_3 has a differential factor for $\mu = 2$ so that attacker tries 2^3 candidate keys instead of 2^4 due to Theorem 2.1. We must consider all output differences to calculate the advantage of S_3.

We have an input difference 2 for the S_3. Suppose that the output difference of S_3 is 3. Since there is no differential factor for $\mu = 3$, the attacker does not have an advantage. Therefore, attacker must try 2^4 key possibilities to capture the corresponding key bits for the S_3. This situation is also the same for the output differences 7, 10, 11, and 15. But if the output difference of S_3 is 1, there exists a differential factor for $\mu = 1$. Because of the Theorem 2.1, it is unnecessary to try half of the keys. Therefore, attacker must try 2^3 key possibilities to capture the corresponding key bits for the S_3. Since we do not exactly know output differences, the advantage calculation must depend on all output differences. We find the mean of all advantages.

In the following equation, 2^4 is the advantage for the output differences 3, 5, 7, 13, 15 and 2^3 is the advantage for the output difference 1. We find these advantages by using Theorem 2.1. The advantage calculation for the S_3, we calculate the mean of 6 advantages;

$$\frac{2^4 + 2^4 + 2^4 + 2^4 + 2^4 + 2^3}{6} = 2^3 \cdot \frac{11}{6} = 2^3 \cdot 2^{0.87447}$$

$$2^{4-3.87447} = 2^{0.12553}$$

Input difference b of S_4 can result in 4 different output difference. One of these output differences is 1 which also congruence the value μ for S_4 according to Table 3.6. In other words, if output difference of S_4 is 1 in differential attack, the attacker can not capture key bits of corresponding S-box S_4. Since the attacker has four active S-boxes, attacker try to capture 4-bits of round key corresponding to S_4. However, S_4 has a differential factor for $\mu = 2$ so that attacker tries 2^3 candidate keys instead of 2^4 due to Theorem 2.1. We must consider all output differences to calculate the advantage of S_4.

We have an input difference b for the S_4. Suppose that the output difference of S_4 is 4. Since there is no differential factor for $\mu = 4$, the attacker does not have an advantage. Therefore, attacker must try 2^4 key possibilities to capture the corresponding key bits for the S_4. This situation is also the same for the output differences 6, and 7. But if the output difference of S_4 is 1, there exists a differential factor for $\mu = 1$. Because of the Theorem 2.1, it is unnecessary to try half of the keys. Therefore, attacker must try 2^3 key possibilities to capture the corresponding key bits for the S_4. Since we do not exactly know output differences, the advantage calculation must depend on all output differences. We find the mean of all advantages.

In the following equation, 2^4 is the advantage for the output differences 4, 6, 7 and 2^3 is the advantage for the output difference 1. We find these advantages by using Theorem 2.1. The advantage calculation for the S_4, we calculate the mean of 4 advantages;

$$\frac{2^4 + 2^4 + 2^4 + 2^3}{4} = 2^3 \cdot \frac{7}{4} = 2^3 \cdot 2^{0.80735}$$

$$2^{4-3.80735} = 2^{0.19265}$$

To find the total advantage of 5^{th}, we take the average of advantages of four activated S-boxes S_1, S_2, S_3 and S_4.

Total advantage for the path 5;

$$2^{0.12553}\cdot 2^{0.12553}\cdot 2^{0.12553}\cdot 2^{0.19265}=2^{0.56924}$$

We calculate the total advantages for all 8 paths, separately. Then, we combine them to find the average advantage for the 16^{th} round. The average advantage is equal to $2^{0.47613}$. That means the time complexity can be reduced by $2^{0.47613}$ for one round.

The attack [24] is corrected in this work. Authors of [24] did not recognize possibility of existence of differential factors in the 16^{th} round which are given as a bold in Table 3.7. Hence, it may not be possible to obtain some bits of key in the 16^{th} round if outputs correspond the differential factors.

Moreover, we have noticed that undisturbed bits of LBLOCK S-boxes are used as input differences 1011, 0010, 0010, 0010 yields to output differences 0???, ???1, ??1?, ??1? which can be seen in Table 3.7 $X_{16,O}[14]$, $X_{16,O}[11]$, $X_{16,O}[9]$, $X_{16,O}[8]$. Nevertheless, there may an extra step to obtain key bits corresponding to undisturbed bits. Thus the time complexity may increase. We just give a correction for the time complexity of this attack by using differential factors. But, further correction may be given for final time complexity of this attack by using undisturbed bits.

Secondly, we investigate the 8 different paths again this time for 17^{th} round, combination of 16^{th} and 17^{th} round and total average for this attack. We firstly check output of active S-boxes and if they have differential factors we calculate the advantages on time complexity for all paths, separately. Then, we combine the advantages of each path in round 16 and 17. Therefore, we get average advantages for each 8 paths for two rounds. Using these advantages, we find total advantage for the attack.

We summarize the 17- round attack in Table 3.9.

Table 3.9: 17-round differential attack of LBLOCK for the 5^{th} path. Differential factors may occur in bold locations.

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
15-Round Differential Δ_{15}																
$X_{16,I}$	0000	0000	0000	1011	0010	0010	0010	0000	0000	0010	0000	0000	0001	0000	0000	0100
$X_{16,O}$	0010	**0???**	0000	0001	**??1?**	0000	**???1**	**??1?**	0000	0000	0000	1011	0010	0010	0010	0000
$X_{17,I}$	0010	0???	0000	0001	??1?	0000	???1	??1?	0000	0000	0000	1011	0010	0010	0010	0000
$X_{17,O}$	**????**	****	****	**0000**	**0010**	????	**????**	**????**	0000	1011	0010	0010	0010	0000	0000	0000

For the path 5;

$$\Delta 15=(000b2220,00200100)\overset{2round}{\rightarrow}\Delta 17=(?\star\star?????,??0??0??).$$

The mean of ?'s of the output is that these locations must be active, the mean of $\star$'s of the output is that these locations may be active or non-active depending on the output differences of the corresponding S-boxes.

We observe that the output differences $?_1$, $?_2$, $?_3$, and $?_4$ of S_4, S_3, S_2 and S_1 from DDT, respectively. If S-boxes have the differential factors, we assigned to $?_i$'s , $i=1,2,3,4$,

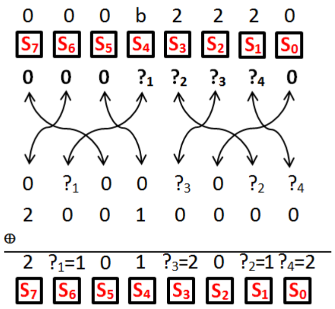

Figure 3.4: Encryption scheme of LBLOCK for the path 5^{th} for two round

the corresponding μ value. After permuted and XORed with the right side, we get the input for the second round. We check whether the S-boxes have differential factors for that input. We find that the output differences of S_7, S_6, S_4, S_3, S_1, and S_0 can all take 6 different values, respectively. We notice that one of the six different output differences has differential factor property for these S-boxes. Thus, attacker can take advantage $2^{0.12553}$ for S_7, S_6, S_4, S_3, S_1, and S_0, separately. The advantage of this path for this round is $2^{0.75318}$. The overall advantage, combination of two round, for this path is $2^{1.32242}$.

One of the output differences $?_2$, $?_3$ and $?_4$ of the input difference 2 and b for S_3, S_2 and S_1 is equal to one of the μ value in Table 3.6. By assigning to $?_i$'s , $i = 1, 2, 3, 4$, the corresponding μ value , we get $?_1 = 1$, $?_2 = 1$, $?_3 = 2$ and $?_4 = 2$.

Table 3.10: Difference Distribution Table of Activated S-boxes for the path 5^{th}

	S_7	S_6	S_4	S_3	S_1	S_0
Input Differences	2	1	1	2	1	2
Output Differences	$\boxed{2}$ 3 7 a b f	$\boxed{2}$ 3 7 a b f	$\boxed{1}$ 5 9 b d f	$\boxed{1}$ 3 5 7 d f	$\boxed{2}$ 3 7 a b f	$\boxed{1}$ 3 7 9 b f

The red boxes in Table 3.10 show the value μ corresponding S-box.

To find advantages, we use the same idea that is provided above. The advantage calculation is for the activated S-boxes S_0, S_1, S_3, S_4, S_6, and S_7, separately;

$$\frac{2^4 + 2^4 + 2^4 + 2^4 + 2^4 + 2^3}{6} = 2^3 \cdot \frac{11}{6} = 2^3 \cdot 2^{0.87447}$$

$$2^{4-3.87447} = 2^{0.12553}$$

Each S-box has an advantage with $2^{0.12553}$. The total advantage is the multiplication of each advantage so that the total advantage is for the 5^{th} in 17^{th} round;

$$2^{0.12553} \cdot 2^{0.12553} \cdot 2^{0.12553} \cdot 2^{0.12553} \cdot 2^{0.12553} \cdot 2^{0.12553} = 2^{0.75318}$$

We combine 16^{th} and 17^{th} then we find the advantage of the combination of two rounds for the 5^{th} path;

$$2^{0.56924} \cdot 2^{0.75318} = 2^{1.32242}$$

We apply the same idea for the rest of given paths to find their advantages. We find the average advantage using advantages of the combination of two rounds for all paths. The multiplication of each advantages give us the result that is equal to $2^{1.04627}$. That means the time complexity can be reduced by $2^{1.04627}$. However, that is an approximate calculation. Since we are away from the differential characteristic by 2-rounds, the reduction on the complexity may not be valid when the attack is performed practically.

Moreover, the authors of [24] claim that they find the best 188 multiple differential paths by investigating the given paths in Table 3.4. According to their claim, the best time and data complexity can be found by applying these 188 paths. The results of author are given the following table.

Table 3.11: Size of the key candidate list, data complexity and computational cost using multiple differential paths [24]

l	24	38
N	59.7523	53.4064
$log(Time)$	67.5211	78

Since the paths are not given in [24], we can just make an approximate calculation for time complexity depending on the calculations of single path. We use the same formula that is given by the attacker in [24] to calculate time complexity. We show that our results in Table 3.12. Time complexity for $l = 38$ does not change since it is time complexity of exhaustive search and it is more higher than complexity of guessing some round key bits using characteristic.

Table 3.12: Modification of the Table 3.11

l	24	38
N	59.7523	53.4064
$log(Time)$	67.09574	78

CHAPTER 4

CONCLUSION

Cryptography, which has existed for centuries, is now being used in many areas of our life. The cryptographic algorithms are selected according to the area and the needs to be used can mainly be divided into two categories; Asymmetric-key algorithms and Symmetric-key algorithms. Symmetric-key algorithms also have own categories. One of these categories is block ciphers. The security evaluation of the block ciphers is done by using several cryptanalytic methods. Generally, the first applied method is differential cryptanalysis. The attacker tries to get round keys using the knowledge of relation between plaintext and ciphertext. However, all of the attacked key bits may not be determined if the S-box of the algorithm has a differential factor property and that S-box is activated in the distinguisher.

A new electronics structure is introduced as embedded systems with the development in technology. To fulfill the needs for new cryptographic tools that are appropriate for embedded systems, a new area in cryptography is occurred and called lightweight cryptography. It is actually based on optimizing the trade-off between security, cost, and performance. Since low resource devices such as RFID tags and sensor networking are started to use in different areas widely, the increase on the needs for lightweight cryptographic instruments has become inevitable. Eventually, lightweight cryptography has become a trend topic for the last few years. As a result, several lightweight block ciphers have been designed such as PRESENT, SEA, LED. In this work, we briefly describe some lightweight block ciphers, their cryptanalysis and corrected cryptanalysis via differential factors.

In this study, we focus on the lightweight block cipher LBLOCK and further investigated its security against differential cryptanalysis. In Chapter 1, we mention about briefly block ciphers, lightweight block ciphers, the S-boxes with properties and cryptanalysis methods. In Chapter 2, we give information on the differential cryptanalysis and types of it. Also, we present the theory of the differential factors. With a recently introduced property called differential factors, we have a new criterion to analyze a S-box. Before the differential factors were introduced, the attackers believe that all bits of the attacked key corresponding to active S-boxes in a differential-like attack could be captured. But then, the work [77] shows that this situation may not be valid if the active S-boxes in the attack have differential factors. In [77] presents that the existence of differential factors may change time, data and memory complexities. Thus, some algorithms having differential-like cryptanalysis in the literature may require corrections. One of these algorithms is LBLOCK. In Chapter 3, we summarize LBLOCK and its differential attack, we show that the all bits of the attacked key that corresponds activated S-box bits cannot be captured. Thus, we reduce the time complexity marginally on this cipher and we note that the time complexity may require some corrections due to existence of the undisturbed bits.

REFERENCES

[1] Fips pub 197, advanced encryption standard (aes), 2001, u.S.Department of Commerce/National Institute of Standards and Technology.

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

[3] ISO/IEC JTC 1 : Vision, mission and principles, 2014.

[4] M. R. Albrecht, B. Driessen, E. B. Kavun, G. Leander, C. Paar, and T. Yalçin, Block ciphers - focus on the linear layer (feat. PRIDE), in J. A. Garay and R. Gennaro, editors, *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*, volume 8616 of *Lecture Notes in Computer Science*, pp. 57–76, Springer, 2014.

[5] S. S. M. AlDabbagh, I. F. T. A. Shaikhli, and M. A. Alahmad, HISEC: A new lightweight block cipher algorithm, in R. Poet and M. Rajarajan, editors, *Proceedings of the 7th International Conference on Security of Information and Networks, Glasgow, Scotland, UK, September 9-11, 2014*, p. 151, ACM, 2014.

[6] R. AlTawy, M. Tolba, and A. M. Youssef, A higher order key partitioning attack with application to lblock, in S. E. Hajji, A. Nitaj, C. Carlet, and E. M. Souidi, editors, *Codes, Cryptology, and Information Security - First International Conference, C2SI 2015, Rabat, Morocco, May 26-28, 2015, Proceedings - In Honor of Thierry Berger*, volume 9084 of *Lecture Notes in Computer Science*, pp. 215–227, Springer, 2015.

[7] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, Midori: A block cipher for low energy, in T. Iwata and J. H. Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pp. 411–436, Springer, 2015.

[8] A. Baysal and S. Sahin, Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors, in T. Güneysu, G. Leander, and A. Moradi, editors, *Lightweight Cryptography for Security and Privacy - 4th International Workshop, LightSec 2015, Bochum, Germany, September 10-11, 2015, Revised Selected Papers*, volume 9542 of *Lecture Notes in Computer Science*, pp. 58–76, Springer, 2015.

[9] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, The SIMON and SPECK families of lightweight block ciphers, IACR Cryptology ePrint Archive, 2013, p. 404, 2013.

[10] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, and S. M. Sim, The SKINNY family of block ciphers and its low-latency variant MANTIS, in M. Robshaw and J. Katz, editors, *Advances in Cryptology - CRYPTO 2016 - 36th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2016, Proceedings, Part II*, volume 9815 of *Lecture Notes in Computer Science*, pp. 123–153, Springer, 2016.

[11] E. Biham, New types of cryptanalytic attacks using related keys, J. Cryptology, 7(4), pp. 229–246, 1994.

[12] E. Biham, R. J. Anderson, and L. R. Knudsen, Serpent: A new block cipher proposal, in *Fast Software Encryption, 5th International Workshop, FSE '98, Paris, France, March 23-25, 1998, Proceedings*, volume 1372 of *Lecture Notes in Computer Science*, pp. 222–238, Springer, 1998.

[13] E. Biham, A. Biryukov, and A. Shamir, Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials, in J. Stern, editor, *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pp. 12–23, Springer, 1999.

[14] E. Biham and A. Shamir, Differential cryptanalysis of des-like cryptosystems, in *Proceedings of the 10th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '90, pp. 2–21, Springer-Verlag, London, UK, UK, 1991, ISBN 3-540-54508-5.

[15] E. Biham and A. Shamir, Differential cryptanalysis of the full 16-round des, in E. F. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, pp. 487–496, Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.

[16] C. Blondeau and B. Gérard, Multiple differential cryptanalysis: Theory and practice, in A. Joux, editor, *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, volume 6733 of *Lecture Notes in Computer Science*, pp. 35–54, Springer, 2011.

[17] A. Bogdanov, M. Knezevic, G. Leander, D. Toz, K. Varici, and I. Verbauwhede, spongent: A lightweight hash function, in *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, pp. 312–325, 2011.

[18] 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, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pp. 450–466, Springer, 2007.

[19] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçin, PRINCE - A low-latency block cipher for pervasive computing applications - extended abstract, in X. Wang and K. Sako, editors, *Advances in Cryptology - ASIACRYPT 2012 - 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings*, volume 7658 of *Lecture Notes in Computer Science*, pp. 208–225, Springer, 2012.

[20] E. Boss, V. Grosso, T. Güneysu, G. Leander, A. Moradi, and T. Schneider, Strong 8-bit sboxes with efficient masking in hardware, IACR Cryptology ePrint Archive, 2016, p. 647, 2016.

[21] C. Boura, M. Naya-Plasencia, and V. Suder, Scrutinizing and improving impossible differential attacks: Applications to clefia, camellia, lblock and simon, in P. Sarkar and T. Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pp. 179–199, Springer, 2014.

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

[23] C. Carlet, S-boxes, boolean functions and codes for the resistance of block ciphers to cryptographic attacks, with or without side channels, in R. S. Chakraborty, P. Schwabe, and J. A. Solworth, editors, *Security, Privacy, and Applied Cryptography Engineering - 5th International Conference, SPACE 2015, Jaipur, India, October 3-7, 2015, Proceedings*, volume 9354 of *Lecture Notes in Computer Science*, pp. 151–171, Springer, 2015.

[24] J. Chen and A. Miyaji, Differential cryptanalysis and boomerang cryptanalysis of lblock, in A. Cuzzocrea, C. Kittl, D. Simos, E. Weippl, and L. Xu, editors, *Security Engineering and Intelligence Informatics*, volume 8128 of *Lecture Notes in Computer Science*, pp. 1–15, Springer Berlin Heidelberg, 2013, ISBN 978-3-642-40587-7.

[25] N. Courtois and J. Pieprzyk, Cryptanalysis of block ciphers with overdefined systems of equations, in 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*, pp. 267–287, Springer, 2002.

[26] N. T. Courtois, An improved differential attack on full GOST, in *The New Codebreakers - Essays Dedicated to David Kahn on the Occasion of His 85th Birthday*, pp. 282–303, 2016.

[27] J. Daemen, M. Peeters, G. V. Assche, and V. Rijmen, Nessie proposal: the block cipher NOEKEON, Nessie submission, 2000, http://gro.noekeon.org/.

[28] Y. Dai and S. Chen, Cryptanalysis of full PRIDE block cipher, SCIENCE CHINA Information Sciences, 60(5), pp. 052108:1–052108:12, 2017.

[29] W. Diffie and M. E. Hellman, Special feature exhaustive cryptanalysis of the NBS data encryption standard, IEEE Computer, 10(6), pp. 74–84, 1977.

[30] I. Dinur and A. Shamir, Cube attacks on tweakable black box polynomials, in A. Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pp. 278–299, Springer, 2009.

[31] E. Doğan, *Differential Factors and Differential Cryptanalysis of Block Cipher PRIDE*, Master's thesis, METU, 2017.

[32] 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, *Progress in Cryptology - INDOCRYPT 2008, 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008. Proceedings*, volume 5365 of *Lecture Notes in Computer Science*, pp. 308–321, Springer, 2008.

[33] V. D. (Ed.):, GOST 28147-89: Encryption, decryption, and message authentication code (mac) algorithms. in: Internet engineering task force rfc 5830, (March,2010).

[34] P. FIPS, 46-3: Data encryption standard (des), National Institute of Standards and Technology, 25(10), pp. 46–2, 1999.

[35] J. Guo, T. Peyrin, A. Poschmann, and M. J. B. Robshaw, The LED block cipher, in 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*, pp. 326–341, Springer, 2011.

[36] M. E. Hellman, Des will be totally insecure within ten years, IEEE Spectrum, 16(7), pp. 32–40, July 1979, ISSN 0018-9235.

[37] J. Jean, I. Nikolic, and T. Peyrin, Joltik v1. submission to the caesar competition, 2014.

[38] P. Junod and S. Vaudenay, FOX : A new family of block ciphers, in H. Handschuh and M. A. Hasan, editors, *Selected Areas in Cryptography, 11th International Workshop, SAC 2004, Waterloo, Canada, August 9-10, 2004, Revised Selected Papers*, volume 3357 of *Lecture Notes in Computer Science*, pp. 114–129, Springer, 2004.

[39] F. Karakoç, H. Demirci, and A. E. Harmanci, Impossible differential cryptanalysis of reduced-round lblock, in I. G. Askoxylakis, H. C. Pöhls, and J. Posegga, editors, *Information Security Theory and Practice. Security, Privacy and Trust in Computing Systems and Ambient Intelligent Ecosystems - 6th IFIP WG 11.2*

International Workshop, WISTP 2012, Egham, UK, June 20-22, 2012. Proceedings, volume 7322 of *Lecture Notes in Computer Science*, pp. 179–188, Springer, 2012.

[40] M. Katagi and S. Moriai, Lightweight cryptography for the internet of things, Sony Corporation.

[41] E. B. Kavun, M. M. Lauridsen, G. Leander, P. S. C. Rechberger, and T. Yalçın, Prost v1. submission to the caesar competition, 2014.

[42] L. R. Knudsen, Cryptanalysis of LOKI91, in J. Seberry and Y. Zheng, editors, *Advances in Cryptology - AUSCRYPT '92, Workshop on the Theory and Application of Cryptographic Techniques, Gold Coast, Queensland, Australia, December 13-16, 1992, Proceedings*, volume 718 of *Lecture Notes in Computer Science*, pp. 196–208, Springer, 1992.

[43] L. R. Knudsen, Truncated and higher order differentials, in *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*, pp. 196–211, 1994.

[44] L. R. Knudsen, G. Leander, A. Poschmann, and M. J. B. Robshaw, Printcipher: A block cipher for ic-printing, in S. Mangard and F. Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pp. 16–32, Springer, 2010.

[45] S. Kolay and D. Mukhopadhyay, Khudra: A new lightweight block cipher for fpgas, in R. S. Chakraborty, V. Matyas, and P. Schaumont, editors, *Security, Privacy, and Applied Cryptography Engineering - 4th International Conference, SPACE 2014, Pune, India, October 18-22, 2014. Proceedings*, volume 8804 of *Lecture Notes in Computer Science*, pp. 126–145, Springer, 2014.

[46] L. Li, B. Liu, and H. Wang, QTL: A new ultra-lightweight block cipher, Microprocessors and Microsystems - Embedded Hardware Design, 45, pp. 45–55, 2016.

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

[48] C. H. Lim, A revised version of crypton - crypton V1.0, in *Fast Software Encryption, 6th International Workshop, FSE '99, Rome, Italy, March 24-26, 1999, Proceedings*, pp. 31–45, 1999.

[49] S. Liu, Z. Gong, and L. Wang, Improved related-key differential attacks on reduced-round lblock, in T. W. Chim and T. H. Yuen, editors, *Information and Communications Security - 14th International Conference, ICICS 2012, Hong Kong, China, October 29-31, 2012. Proceedings*, volume 7618 of *Lecture Notes in Computer Science*, pp. 58–69, Springer, 2012.

[50] Y. Liu, D. Gu, Z. Liu, and W. Li, Impossible differential attacks on reduced-round lblock, in M. D. Ryan, B. Smyth, and G. Wang, editors, *Information Security*

Practice and Experience - 8th International Conference, ISPEC 2012, Hangzhou, China, April 9-12, 2012. Proceedings, volume 7232 of *Lecture Notes in Computer Science*, pp. 97–108, Springer, 2012.

[51] J. L. Massey, SAFER K-64: A byte-oriented block-ciphering algorithm, in R. J. Anderson, editor, *Fast Software Encryption, Cambridge Security Workshop, Cambridge, UK, December 9-11, 1993, Proceedings*, volume 809 of *Lecture Notes in Computer Science*, pp. 1–17, Springer, 1993.

[52] K. A. McKay, L. Bassham, M. S. Turan, and N. Mouha, Report on lightweight cryptography, National Institute of Standards and Technology Internal Report 8114, August 2016.

[53] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, *Handbook of Applied Cryptography*, CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996, ISBN 0849385237.

[54] M. Minier and M. Naya-Plasencia, A related key impossible differential attack against 22 rounds of the lightweight block cipher lblock, Inf. Process. Lett., 112(16), pp. 624–629, 2012.

[55] K. Nyberg, Differentially uniform mappings for cryptography, in T. Helleseth, editor, *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, volume 765 of *Lecture Notes in Computer Science*, pp. 55–64, Springer, 1993.

[56] R. C. Phan, Impossible differential cryptanalysis of 7-round advanced encryption standard (AES), Inf. Process. Lett., 91(1), pp. 33–38, 2004.

[57] G. Piret, T. Roche, and C. Carlet, PICARO - A block cipher allowing efficient higher-order side-channel resistance, in F. Bao, P. Samarati, and J. Zhou, editors, *Applied Cryptography and Network Security - 10th International Conference, ACNS 2012, Singapore, June 26-29, 2012. Proceedings*, volume 7341 of *Lecture Notes in Computer Science*, pp. 311–328, Springer, 2012.

[58] A. Y. Poschmann, *Lightweight cryptography: cryptographic engineering for a pervasive world*, Ph.D. thesis, Ruhr University Bochum, 2009.

[59] M. O. Saarinen, Cryptographic analysis of all 4 x 4-bit s-boxes, in A. Miri and S. Vaudenay, editors, *Selected Areas in Cryptography - 18th International Workshop, SAC 2011, Toronto, ON, Canada, August 11-12, 2011, Revised Selected Papers*, volume 7118 of *Lecture Notes in Computer Science*, pp. 118–133, Springer, 2011.

[60] Y. Sasaki and L. Wang, Comprehensive study of integral analysis on 22-round lblock, in T. Kwon, M. Lee, and D. Kwon, editors, *Information Security and Cryptology - ICISC 2012 - 15th International Conference, Seoul, Korea, November 28-30, 2012, Revised Selected Papers*, volume 7839 of *Lecture Notes in Computer Science*, pp. 156–169, Springer, 2012.

[61] Y. Sasaki and L. Wang, Meet-in-the-middle technique for integral attacks against feistel ciphers, in L. R. Knudsen and H. Wu, editors, *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*, volume 7707 of *Lecture Notes in Computer Science*, pp. 234–251, Springer, 2012.

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

[63] J. Seberry, X. Zhang, and Y. Zheng, Pitfalls in designing substitution boxes (extended abstract), in Y. Desmedt, editor, *Advances in Cryptology - CRYPTO '94, 14th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1994, Proceedings*, volume 839 of *Lecture Notes in Computer Science*, pp. 383–396, Springer, 1994.

[64] A. Selçuk, On probability of success in linear and differential cryptanalysis, Journal of Cryptology, 21(1), pp. 131–147, 2008, ISSN 0933-2790.

[65] J. Shan, L. Hu, L. Song, S. Sun, and X. Ma, Related-key differential attack on round reduced RECTANGLE-80, IACR Cryptology ePrint Archive, 2014, p. 986, 2014.

[66] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, Piccolo: An ultra-lightweight blockcipher, in 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*, pp. 342–357, Springer, 2011.

[67] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, *The 128-Bit Blockcipher CLEFIA (Extended Abstract)*, pp. 181–195, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, ISBN 978-3-540-74619-5.

[68] H. Soleimany and K. Nyberg, Zero-correlation linear cryptanalysis of reduced-round lblock, Des. Codes Cryptography, 73(2), pp. 683–698, 2014.

[69] F. Standaert, G. Piret, N. Gershenfeld, and J. Quisquater, SEA: A scalable encryption algorithm for small embedded applications, in J. Domingo-Ferrer, J. Posegga, and D. Schreckling, editors, *Smart Card Research and Advanced Applications, 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006, Tarragona, Spain, April 19-21, 2006, Proceedings*, volume 3928 of *Lecture Notes in Computer Science*, pp. 222–236, Springer, 2006.

[70] S. Sun, L. Hu, M. Wang, P. Wang, K. Qiao, X. Ma, D. Shi, and L. Song, Automatic enumeration of (related-key) differential and linear characteristics with predefined properties and its applications, IACR Cryptology ePrint Archive, 2014, p. 747, 2014.

[71] S. Sun, L. Hu, P. Wang, K. Qiao, X. Ma, and L. Song, Automatic security evaluation and (related-key) differential characteristic search: Application to simon, present, lblock, DES(L) and other bit-oriented block ciphers, in P. Sarkar and

T. Iwata, editors, *Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I*, volume 8873 of *Lecture Notes in Computer Science*, pp. 158–178, Springer, 2014.

[72] C. Tezcan, The improbable differential attack: Cryptanalysis of reduced round CLEFIA, IACR Cryptology ePrint Archive, 2010, p. 435, 2010.

[73] C. Tezcan, Improbable differential attacks on present using undisturbed bits, J. Computational Applied Mathematics, 259, pp. 503–511, 2014.

[74] C. Tezcan, Differential factors revisited: Corrected attacks on PRESENT and SERPENT, in T. Güneysu, G. Leander, and A. Moradi, editors, *Lightweight Cryptography for Security and Privacy - 4th International Workshop, LightSec 2015, Bochum, Germany, September 10-11, 2015, Revised Selected Papers*, volume 9542 of *Lecture Notes in Computer Science*, pp. 21–33, Springer, 2015.

[75] C. Tezcan, A. Doğanaksoy, G. O. Okan, A. Şenol, E. Doğan, F. Yücebaş, and N. Baykal, On differential factors, 2017.

[76] C. Tezcan, G. O. Okan, A. Şenol, E. Doğan, F. Yücebas, and N. Baykal, Differential attacks on lightweight block ciphers present, pride, and RECTANGLE revisited, in *Lightweight Cryptography for Security and Privacy - 5th International Workshop, LightSec 2016, Aksaray, Turkey, September 21-22, 2016, Revised Selected Papers*, pp. 18–32, 2016.

[77] C. Tezcan and F. Özbudak, Differential factors: Improved attacks on serpent, in T. Eisenbarth and E. Öztürk, editors, *Lightweight Cryptography for Security and Privacy*, volume 8898 of *Lecture Notes in Computer Science*, pp. 69–84, Springer International Publishing, 2015, ISBN 978-3-319-16362-8.

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

[79] M. Wang, Differential cryptanalysis of reduced-round PRESENT, in *Progress in Cryptology - AFRICACRYPT 2008, First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings*, pp. 40–49, 2008.

[80] M. Wang, Y. Sun, E. Tischhauser, and B. Preneel, A model for structure attacks, with applications to PRESENT and serpent, in *Fast Software Encryption - 19th International Workshop, FSE 2012, Washington, DC, USA, March 19-21, 2012. Revised Selected Papers*, pp. 49–68, 2012.

[81] N. Wang, X. Wang, and K. Jia, Improved impossible differential attack on reduced-round lblock, in S. Kwon and A. Yun, editors, *Information Security and Cryptology - ICISC 2015 - 18th International Conference, Seoul, South Korea, November 25-27, 2015, Revised Selected Papers*, volume 9558 of *Lecture Notes in Computer Science*, pp. 136–152, Springer, 2015.

[82] Y. Wang and W. Wu, Improved multidimensional zero-correlation linear cryptanalysis and applications to lblock and TWINE, in W. Susilo and Y. Mu, editors, *Information Security and Privacy - 19th Australasian Conference, ACISP 2014, Wollongong, NSW, Australia, July 7-9, 2014. Proceedings*, volume 8544 of *Lecture Notes in Computer Science*, pp. 1–16, Springer, 2014.

[83] Y. Wang, W. Wu, X. Yu, and L. Zhang, Security on lblock against biclique cryptanalysis, in D. H. Lee and M. Yung, editors, *Information Security Applications - 13th International Workshop, WISA 2012, Jeju Island, Korea, August 16-18, 2012, Revised Selected Papers*, volume 7690 of *Lecture Notes in Computer Science*, pp. 1–14, Springer, 2012.

[84] L. Wen, M. Wang, and J. Zhao, Related-key impossible differential attack on reduced-round lblock, J. Comput. Sci. Technol., 29(1), pp. 165–176, 2014.

[85] W. Wu and L. Zhang, Lblock: A lightweight block cipher, in J. Lopez and G. Tsudik, editors, *Applied Cryptography and Network Security*, volume 6715 of *Lecture Notes in Computer Science*, pp. 327–344, Springer Berlin Heidelberg, 2011, ISBN 978-3-642-21553-7.

[86] Q. Yang, L. Hu, S. Sun, K. Qiao, L. Song, J. Shan, and X. Ma, Improved differential analysis of block cipher PRIDE, in J. Lopez and Y. Wu, editors, *Information Security Practice and Experience - 11th International Conference, ISPEC 2015, Beijing, China, May 5-8, 2015. Proceedings*, volume 9065 of *Lecture Notes in Computer Science*, pp. 209–219, Springer, 2015.

[87] L. Zhang, W. W, Y. Wang, S. Wu, and J. Zhang, Lac: A lightweight authenticated encryption cipher. submission to the caesar competition, 2014.

[88] 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.

[89] J. Zhao, X. Wang, M. Wang, and X. Dong, Differential analysis on block cipher PRIDE, IACR Cryptology ePrint Archive, 2014, p. 525, 2014.

APPENDIX A

Difference Distribution Tables of LBLOCK's S-boxes

Table A.1: The Difference Distrubution Table of the S-box S_0 and S_8

ΔY ΔX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	4	0	2	0	0	0	2	0	4	0	2	0	0	0	2
2	0	4	0	2	0	0	0	2	0	4	0	2	0	0	0	2
3	0	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0
4	0	0	0	2	4	2	4	0	0	0	0	2	0	2	0	0
5	0	0	0	0	4	2	0	2	0	0	4	0	0	2	0	2
6	0	0	4	0	4	2	0	2	0	0	0	0	0	2	0	2
7	0	0	0	2	4	2	0	0	0	0	0	2	0	2	4	0
8	0	0	2	2	0	0	2	2	0	0	2	2	0	0	2	2
9	0	0	0	2	0	0	0	2	4	0	0	2	4	0	0	2
A	0	4	2	0	0	0	2	0	0	0	2	0	0	4	2	0
B	0	4	0	0	0	0	0	0	4	0	0	0	4	4	0	0
C	0	0	2	2	0	2	2	0	2	2	0	0	2	0	0	2
D	0	0	0	2	0	2	0	0	2	2	2	0	2	0	2	2
E	0	0	2	0	0	2	2	2	2	2	0	2	2	0	0	0
F	0	0	0	0	0	2	0	2	2	2	2	2	2	0	2	0

Table A.2: The Difference Distrubution Table of the S-box S_1 and S_9

ΔY ΔX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	4	2	0	0	0	2	0	0	4	2	0	0	0	2
2	0	0	4	2	0	0	0	2	0	0	4	2	0	0	0	2
3	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0
4	0	0	0	2	4	4	2	0	0	0	0	2	0	0	2	0
5	0	0	0	0	4	0	2	2	0	4	0	0	0	0	2	2
6	0	4	0	0	4	0	2	2	0	0	0	0	0	0	2	2
7	0	0	0	2	4	0	2	0	0	0	0	2	0	4	2	0
8	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2
9	0	0	0	2	0	0	0	2	4	0	0	2	4	0	0	2
A	0	2	4	0	0	2	0	0	0	2	0	0	0	2	4	0
B	0	0	4	0	0	0	0	0	4	0	0	0	4	0	4	0
C	0	2	0	2	0	2	2	0	2	0	2	0	2	0	0	2
D	0	0	0	2	0	0	2	0	2	2	2	0	2	2	0	2
E	0	2	0	0	0	2	2	2	2	0	2	2	2	2	2	2
F	0	0	0	0	0	0	2	2	2	2	2	2	2	2	0	0

Table A.3: The Difference Distrubution Table of the S-box S_2

ΔY ΔX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	4	0	0	0	4	0	0	0	2	2	0	0	2	2
2	0	0	4	0	0	0	4	0	0	0	2	2	0	0	2	2
3	0	0	0	0	0	0	0	0	4	4	0	0	4	4	0	0
4	0	4	0	2	0	0	0	2	0	4	2	0	0	0	2	0
5	0	4	0	2	0	0	0	2	0	0	0	2	4	0	0	2
6	0	4	0	2	0	0	0	2	4	0	0	2	0	0	0	2
7	0	4	0	2	0	0	0	2	0	0	2	0	0	4	2	0
8	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2
9	0	0	0	0	4	4	0	0	0	0	2	2	0	0	2	2
A	0	0	4	0	0	0	0	4	2	2	0	0	2	2	0	0
B	0	0	4	0	4	4	0	4	0	0	0	0	0	0	0	0
C	0	0	0	2	2	2	2	0	2	2	2	0	0	0	0	2
D	0	0	0	2	2	2	2	0	0	0	2	0	2	2	0	2
E	0	0	0	2	2	2	2	0	2	2	0	2	0	0	2	0
F	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0

Table A.4: The Difference Distrubution Table of the S-box S_3

ΔY ΔX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	4	0	4	0	2	0	2	0	0	0	0	0	2	0	2
2	0	4	0	4	0	2	0	2	0	0	0	0	0	2	0	2
3	0	0	0	0	4	0	4	0	0	0	0	0	4	0	4	0
4	0	0	0	0	0	2	0	2	4	2	0	2	4	0	0	0
5	0	0	0	0	0	0	4	0	4	2	0	2	0	2	0	2
6	0	0	0	0	4	0	0	0	4	2	0	2	0	2	0	2
7	0	0	0	0	0	2	0	2	4	2	0	2	0	0	4	0
8	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2
9	0	0	4	0	0	2	0	2	0	0	4	0	0	2	0	2
A	0	4	0	0	2	0	2	0	0	0	0	4	2	0	2	0
B	0	4	4	0	0	0	0	0	0	0	4	4	0	0	0	0
C	0	0	2	2	2	2	0	0	0	2	2	0	2	0	0	2
D	0	0	2	2	0	2	2	0	0	2	2	0	0	0	2	2
E	0	0	2	2	2	0	0	2	0	2	2	0	2	2	0	0
F	0	0	2	2	0	0	2	2	0	2	2	0	0	2	2	0

Table A.5: The Difference Distrubution Table of the S-box S_4

ΔY ΔX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	4	0	0	0	4	0	0	0	2	0	2	0	2	0	2
2	0	4	0	0	0	4	0	0	0	2	0	2	0	2	0	2
3	0	0	0	0	0	0	0	0	4	0	4	0	4	0	4	0
4	0	0	4	2	0	0	0	2	0	2	4	0	0	2	0	0
5	0	0	4	2	0	0	0	2	0	0	0	2	4	0	0	2
6	0	0	4	2	0	0	0	2	4	0	0	2	0	0	0	2
7	0	0	4	2	0	0	0	2	0	2	0	0	0	2	4	0
8	0	0	0	0	0	0	0	0	2	2	2	2	2	2	2	2
9	0	0	0	0	4	0	4	0	0	2	0	2	0	2	0	2
A	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
B	0	4	0	0	4	0	4	4	0	0	0	0	0	0	0	0
C	0	0	0	2	2	2	2	0	2	2	2	0	0	0	0	2
D	0	0	0	2	2	2	2	0	0	2	0	0	2	0	2	2
E	0	0	0	2	2	2	2	0	2	0	2	2	0	2	0	0
F	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2	0

Table A.6: The Difference Distrubution Table of the S-box S_5

ΔY ΔX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	4	0	0	0	2	0	2	0	4	0	0	0	2	0	2
2	0	4	0	0	0	2	0	2	0	4	0	0	0	2	0	2
3	0	0	0	0	4	0	4	0	0	0	0	0	4	0	4	0
4	0	0	4	2	0	2	4	0	0	0	0	2	0	2	0	0
5	0	0	4	2	0	0	0	2	0	0	0	2	4	0	0	2
6	0	0	4	2	4	0	0	2	0	0	0	2	0	0	0	2
7	0	0	4	2	0	2	0	0	0	0	0	2	0	2	4	0
8	0	0	0	0	2	2	2	2	0	0	0	0	2	2	2	2
9	0	0	0	0	0	2	0	2	4	0	4	0	0	2	0	2
A	0	4	0	0	2	0	2	0	0	0	0	4	2	0	2	0
B	0	4	0	0	0	0	0	0	4	0	4	4	0	0	0	0
C	0	0	0	2	2	2	2	0	2	2	2	0	0	0	0	2
D	0	0	0	2	0	2	0	0	2	2	2	0	2	0	2	2
E	0	0	0	2	2	0	2	2	2	2	2	0	0	2	0	0
F	0	0	0	2	0	0	0	2	2	2	2	0	2	2	2	0

Table A.7: The Difference Distrubution Table of the S-box S_6

ΔY ΔX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	4	2	0	0	0	2	0	0	4	2	0	0	0	2
2	0	0	4	2	0	0	0	2	0	0	4	2	0	0	0	2
3	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0
4	0	0	0	2	4	4	2	0	0	0	0	2	0	0	2	0
5	0	0	0	0	4	0	2	2	0	4	0	0	0	0	2	2
6	0	4	0	0	4	0	2	2	0	0	0	0	0	0	2	2
7	0	0	0	2	4	0	2	0	0	0	0	2	0	4	2	0
8	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2
9	0	0	0	2	0	0	0	2	4	0	0	2	4	0	0	2
A	0	2	4	0	0	2	0	0	0	2	0	0	0	2	4	0
B	0	0	4	0	0	0	0	0	4	0	0	0	4	0	4	0
C	0	2	0	2	0	2	2	0	2	0	2	0	2	0	0	2
D	0	0	0	2	0	0	2	0	2	2	2	0	2	2	0	2
E	0	2	0	0	0	2	2	2	2	0	2	2	2	0	0	0
F	0	0	0	0	0	0	2	2	2	2	2	2	2	2	0	0

Table A.8: The Difference Distrubution Table of the S-box S_7

ΔY ΔX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	4	2	0	0	0	2	0	0	4	2	0	0	0	2
2	0	0	4	2	0	0	0	2	0	0	4	2	0	0	0	2
3	0	4	0	0	0	4	0	0	0	4	0	0	0	4	0	0
4	0	0	0	2	4	4	2	0	0	0	0	2	0	0	2	0
5	0	0	0	0	4	0	2	2	0	4	0	0	0	0	2	2
6	0	4	0	0	4	0	2	2	0	0	0	0	0	0	2	2
7	0	0	0	2	4	0	2	0	0	0	0	2	0	4	2	0
8	0	2	0	2	0	2	0	2	0	2	0	2	0	2	0	2
9	0	0	0	2	0	0	0	2	4	0	0	2	4	0	0	2
A	0	2	4	0	0	2	0	0	0	2	0	0	0	2	4	0
B	0	0	4	0	0	0	0	0	4	0	0	0	4	0	4	0
C	0	2	0	2	0	2	2	0	2	0	2	0	2	0	0	2
D	0	0	0	2	0	0	2	0	2	2	2	0	2	2	0	2
E	0	2	0	0	0	2	2	2	2	0	2	2	2	0	0	0
F	0	0	0	0	0	0	2	2	2	2	2	2	2	2	0	0