DIFFERENTIAL FACTORS AND DIFFERENTIAL CRYPTANALYSIS OF BLOCK CIPHER PRIDE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS INSTITUTE
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EROL DOĞAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CYBER SECURITY

JULY 2017

Approval of the thesis:

DIFFERENTIAL FACTORS AND DIFFERENTIAL CRYPTANALYSIS OF BLOCK CIPHER PRIDE

submitted by **EROL DOĞAN** in partial fulfillment of the requirements for the degree of **Master of Science in Cyber Security Department, Middle East Technical University** by,

Prof. Dr. Deniz ZEYREK BOZŞAHİN _____
Director, Graduate School of **Informatics**

Assist. Prof. Dr. Aybar Can ACAR _____
Head of Department, **Cyber Security**

Assoc. Prof. Dr. Sevgi Özkan YILDIRIM _____
Supervisor, **Information Systems, METU**

Dr. Cihangir TEZCAN _____
Co-supervisor, **Department of Mathematics, METU**

Examining Committee Members:

Assist. Prof. Dr. Aybar Can ACAR
Department of Cyber Security, METU _____

Assoc. Prof. Dr. Sevgi Özkan YILDIRIM
Information Systems, METU _____

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

Assist. Prof. Dr. Banu Yüksel ÖZKAYA
Industrial Engineering Department, Hacettepe University _____

Prof. Dr. Ali Aydın SELÇUK
Computer Engineering Department, TOBB _____

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: EROL DOĞAN

Signature :

ABSTRACT

DIFFERENTIAL FACTORS AND DIFFERENTIAL CRYPTANALYSIS OF BLOCK CIPHER PRIDE

DOĞAN, Erol
M.S., Department of Cyber Security
Supervisor : Assoc. Prof. Dr. Sevgi Özkan YILDIRIM
Co-Supervisor : Dr. Cihangir TEZCAN

July 2017, 83 pages

Today, IoT devices are used in very critical areas like payment cards, contactless keys and biometric authentication. Moreover, while the number of IoT Technologies increases, cryptographic systems that are optimized for IoT devices that require less cost, less power, and less memory are highly required in today's industry. Therefore, in recent years several lightweight block ciphers are published to satisfy industry needs. However, there are still more work needed to be sure about the security of these block ciphers.

Differential cryptanalysis is one of the important methods used in block cipher analysis. This method deals with how minor differences made in the plaintext can lead to certain differences in the cipher text. It is examined that whether the expected differences are observed or not by testing all candidate key bits on a number of plaintext-ciphertext pairs. The correct key is expected to provide these differences more times than the wrong keys. By this means the correct key is captured. However, a recent study, Differential Factors showed that it may not be possible to fully capture the attacked round key bits when performing a differential attack. Besides, another recent study Undisturbed Bits can be used for discovering longer differential characteristics that provides opportunity for more powerful differential attacks.

In this thesis, we have investigated several lightweight block ciphers for the existence

of Differential Factors and Undisturbed Bits. We have also shown how differential factors can be used to reduce the time complexity of differential attacks by summarizing the corrected attacks on PRESENT and SERPENT block ciphers. Moreover, we have also investigated the 18-round, 19-round and 20-round differential attacks on PRIDE block cipher and we have corrected these attacks considering differential factors. As a result, by our correction we have shown that these attacks require more time complexity than they were claimed.

Keywords: Differential Cryptanalysis, PRIDE, Lightweight Block Ciphers, Differential Factors

ÖZ

DİFERANSİYEL FAKTÖRLER VE PRIDE BLOK ŞİFRESİNİN DİFERANSİYEL KRİPTANALİZİ

DOĞAN, Erol
Yüksek Lisans, Siber Güvenlik Bölümü
Tez Yöneticisi : Doç. Dr. Sevgi Özkan YILDIRIM
Ortak Tez Yöneticisi : Dr. Cihangir TEZCAN

Temmuz 2017 , 83 sayfa

Günümüzde IoT cihazları ödeme kartları, temassız anahtarlar ve biyometrik kimlik doğrulama gibi çok kritik alanlarda kullanılmaktalar. Dahası, IoT teknolojilerinin sayısı arttıkça, günümüzde IoT cihazları için optimize edilmiş daha düşük maliyet, daha düşük enerji ve daha düşük bellek gerektiren kriptografik sistemlere çok ihtiyaç bulunmaktadır. Bu nedenle, geçtiğimiz yıllarda endüstrinin bu ihtiyacını karşılayabilmek için birçok hafif ağırlıklı blok şifre yayınlanmıştır. Ancak, bu şifrelerin güvenliğinden emin olabilmek için hala çok çalışmaya ihtiyaç vardır.

Diferansiyel kriptanaliz blok şifre analizinde kullanılan önemli yöntemlerden biridir. Bu yöntem şifresiz metinde yapılan küçük değişikliğin, şifreli metinde ne tür değişikliklere yol açtığıyla ilgilenir. Birçok şifresiz-şifreli metin çiftleri üzerinde aday anahtar bitlerinin tümü denenerek beklenilen farkların gözlemlenip gözlemlenmediği incelenir. Doğru anahtarın yanlış anahtarlara göre daha fazla kez bu farkları sağlaması beklenir. Ancak son zamanlarda yapılan Diferansiyel Faktörler çalışması, diferansiyel saldırıda anahtar bitlerinin tamamının ele geçirilmesinin mümkün olamayabileceğini göstermiştir. Ayrıca, son zamanlarda yapılan bir başka çalışma olan Karıştırılmamış Bitler güçlü diferansiyel ataklara imkan sağlayan daha uzun diferansiyel karakteristikler keşfetmek için kullanılabilir.

Bu tezde, Diferansiyel Faktör ve Karıştırılmamış Bit varlığı için birçok hafif ağırlıklı

blok şifreyi inceledik. Ayrıca, PRESENT ve SERPENT blok şifrelerinin düzeltilmiş saldırılarını özetleyerek, diferansiyel saldırıların zaman karmaşıklığını azaltmak için diferansiyel faktörlerin nasıl kullanılabileceğini gösterdik. Buna ek olarak, PRIDE blok şifresine yapılmış 18-raund, 19-raund ve 20-raund diferansiyel saldırılarını inceledik ve diferansiyel faktörleri dikkate alarak bu saldırıları düzelttik. Sonuç olarak, yaptığımız düzeltme ile bu atakların iddia edildiğinden daha fazla zaman karmaşıklığı gerektirdiğini gösterdik.

Anahtar Kelimeler: Diferansiyel Kriptanaliz, PRIDE, Hafif Ağırlıklı Blok Şifreler, Diferansiyel Faktörler

to my family

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Information security is the set of technologies, processes and practices designed to protect information from unauthorized access, use, disclosure, disruption, modification, inspection, recording or destruction. It is also composed of three main concepts Confidentiality, Integrity and Availability.

- **Condifentiality** of information refers to protecting the information from disclosure to unauthorized parties.
- **Integrity** of information refers to protecting information from being modified by unauthorized parties.
- **Availability** of information refers to ensuring that authorized parties are able to access the information when needed.

The Science of Cryptology is the best method to provide the security of information. Cryptographic encryption algorithms that are implemented on software and hardware devices ensure the confidentiality and integrity of information. Therefore, today cryptology, is used in all areas of information security.

Cryptology has two components, cryptography and cryptanalysis. Cryptography is the science of designing secure ciphers and cryptanalysis is the science of analyzing the security of ciphers by trying to find weaknesses in the design.

1.1 Cryptography Basics

Cryptography has several areas like Symmetric Cryptography, Asymmetric Cryptography, Hash Functions and Randomness. In this section a brief information will be provided for these basic cryptographic areas.

In Symmetric Cryptography, same cryptographic key material is used for both encrypting plaintext and decrypting ciphertext. In other words, when sending a secret message, sender uses the key material for encrypting the message and the receiver uses the same key material for decrypting the encrypted message. Symmetric key algorithms are fast and simple but they have a main drawback that is the sender and the receiver must somehow exchange the keys in a secure way. This can be accomplished with other cryptographic features that are discussed later in this section.

In general there are two types of symmetric algorithms; Block Ciphers and Stream Ciphers. Stream ciphers encrypt the message as the data streams from the origin. In other words, message is not divided into parts in stream ciphers when encrypting a message. Because stream ciphers are beyond the scope of this thesis, there will not be more information provided about it but block ciphers will be explained in detail later in this section.

Caesar cipher, Spartan Scytale or Enigma machine are examples of some historical symmetric algorithms. Same key is used to encrypt and decrypt the secret message in these algorithms. Some of the examples of modern symmetric algorithms are DES, AES [20], CAST [1], Blowfish [56], Twofish [57], Pride [3], Present [14], Rectangle [78] block ciphers.

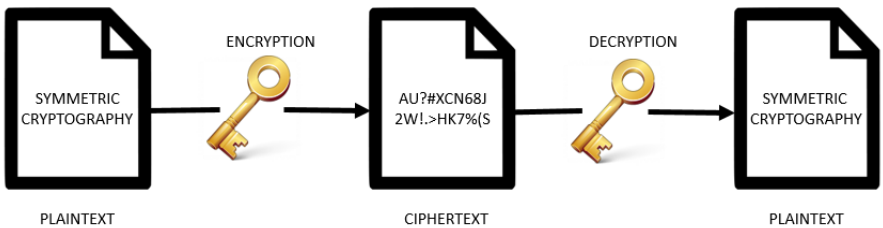

Figure 1.1: Symmetric Cryptography

However, in contrast to Symmetric Cryptography, in Asymmetric Cryptography pairs of two keys are used which are called public key and private key. In these algorithms, when any information is encrypted with one of these keys, it can also be decrypted with the other key. While, public key is available for any party, private key is only known by the owner of the key. While symmetric algorithms generally provide confidentiality by encrypting a secret message, asymmetric algorithms can also provide authentication, integrity and non-repudiation.

Some of the examples of asymmetric algorithms are RSA [55], Diffie-Hellman [23], Digital Signature Algorithm, ElGamal [31] and Elliptic Curve Cryptography [4].

In some cryptographic applications, symmetric and asymmetric algorithms are used together. For example, symmetric algorithm is used to encrypt the message and asymmetric algorithm is used to exchange symmetric keys in a secure way such that symmetric key is encrypted with the public key of the receiver, so that it can only be decrypted by the private key of the receiver which is only available by the receiver.

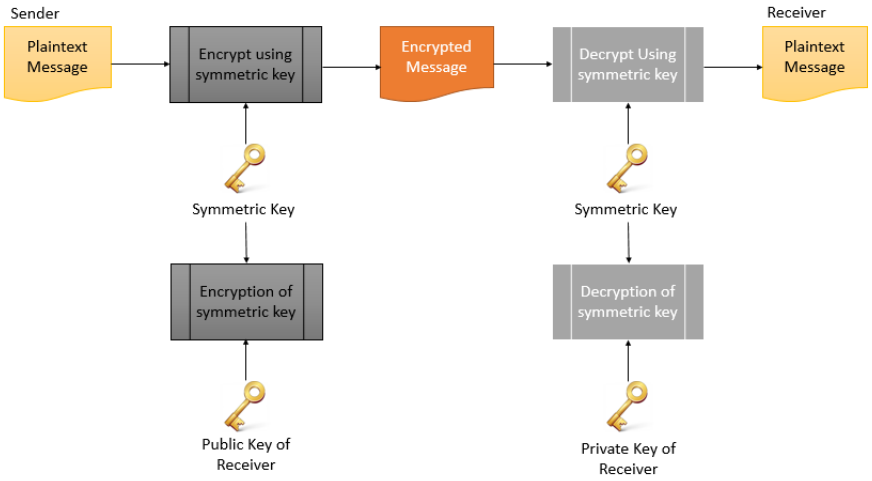

Figure 1.2: Symmetric and Asymmetric Cryptography

In the above example, secret message is encrypted with a symmetric key which is also encrypted with the public key of the receiver. When the receiver receives the encrypted message and the encrypted symmetric key, first, symmetric key is decrypted with the private key of the receiver. Last, encrypted message is decrypted with the symmetric key. As a result, message and symmetric key are sent to the receiver in a secure way. This application of asymmetric algorithm is a good example of key exchange process.

Asymmetric algorithm can also be used for digital signatures which provide to be sure about the integrity of the message and the identity of the sender and receiver. For example; after sending secret message, sender can encrypt the hash of plaintext with his private key, and the receiver decrypt this information with the public key of the sender. As a result of this process, the receiver can be sure that the information is coming from the right sender because public key of the sender can decrypt a message which is encrypted with the private key of sender which is only available by right sender.

Hashing is another special subject of cryptography. A hash funtion creates a fixed-length output from any length of input without using key. Generally hash funtions divide the input message into blocks, and calculate XOR operation of each block. The output is called as hash value of the input message, and this input can not be retrieved from the hash value. The most important thing about hash functions are that it is very difficult to have the same hash value from two different input. As a result of this feature hash functions are used for assuring the integrity of transmitted data. When we think about the above example, if the sender calculates and sends the hash value of the plaintext to the receiver, the receiver can calculate the hash value after decrypting the encrypted message and checks whether the hash value matches with the sender's hash value. If both results are the same, the receiver can be sure about that the message is not changed during transit. Some of the examples of hash functions are MD5, SHA-1, SHA-2, SHA-3 and RIPEMD-160.

Randomness is another cryptographic feature that are commonly used. The main purpose about randomness is producing non-repeating really random numbers. Private keys of digital signature algorithms, initialization values of encryption and password generation are main used areas of randomness. Because we did not work about randomness in this thesis, there will not be more information provided about it.

In this thesis we will focus on design and cryptanalysis of some of the block ciphers, thus in the rest of this thesis, block ciphers will be investigated in detail.

1.2 Block Ciphers

In a block cipher, plaintext information is divided into fixed length parts which are called blocks. Moreover, block ciphers is composed of several rounds that contains encrypting round function. Each block is encrypted at a time, by repeating the same round function at each round. After each block is encrypted, they are combined together to make one encrypted ciphertext.

Generally, size of the block length does not affect the security of the block cipher, but security is directly affected by the length of the key. As Kerckhoffs' principle state that crytographic algorithm should be secure even if the design of the algorithm is known by public except only the key must be secure. Thus, the design of the block cipher including size of the block is known by public but the key is kept as secure.

Today, there are many block ciphers and all have different designs. However, basically they can be categorized as Feistel Networks and Substitution Permutation Networks (SPNs).

In Feistel networks, blocks are divided into two parts as can be seen in Figure 1.3. One of the parts is encrypted with round function and XOR'ed with the other part and the result is replaced with the second part. At each round a different subkey is used that is derived from the master key. Substitution and permutation operations are completed inside the round function. DES algorithm is one of the example of Feistel Network block ciphers.

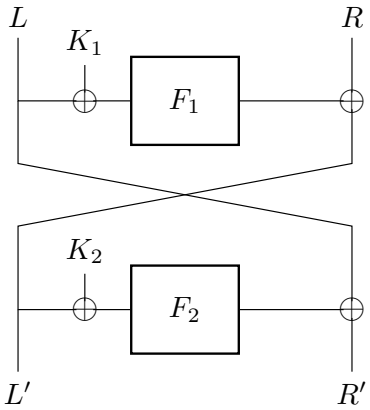

Figure 1.3: Feistel Networks

SPN type block ciphers contain key addition, substitution and permutation layers as can be seen in Figure 1.4. At the key addition layer, round key is XOR'ed with the plaintext. At the Substitution layer, generally S-boxes are used which substitute the *n* bits of the message with different *m* bits to perform confussion. Finally, at the permutation layer generally bit positions are replaced to perform diffusion. Alternative to bit-level permutation, matrix multiplication or shift-row functions are also used for diffusion operation. These operations are completed for each round repeatedly.

In SPN block ciphers a key schedule algorithm is used to derive subkeys from the master key. Thus, at each round different subkey is used at the key addition layer. AES [20] is one of the example of SPN type block ciphers which is accepted as industry standard.

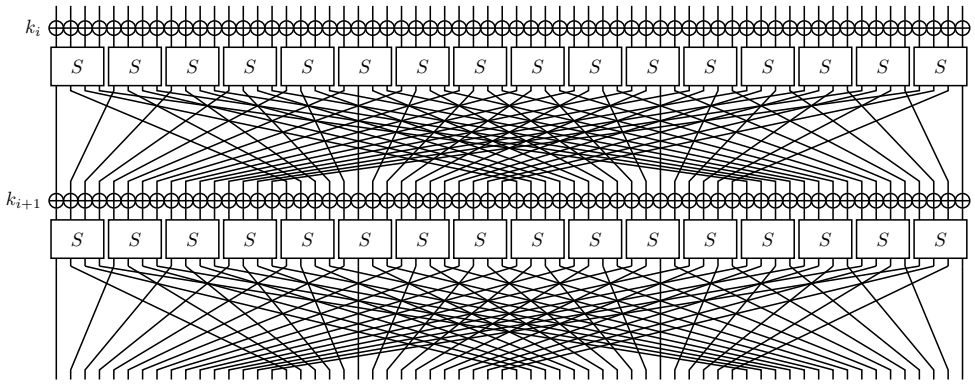

Figure 1.4: PRESENT [14] - SPN Type Block Cipher

1.3 Motivation of Lightweight Block Ciphers

In today's information technology environment we see some evolutionary changes. At the first design, information technology systems were based on mainframe technology in which one super computer was serving many users through thin clients. However, after the mainframe technology times, personal computers became popular and each user had his own computer. This computing model can be called as one user - one computer model and it is still being used today commonly. On the other hand, nowadays we observe some technological improvements that change the computing environment to a new model which can be called as one user - many computers

model. In this model, computers are usually internet connected smart devices embedded with electronics and software. This model is commonly known as Internet of Things (IoT). IoT devices are used in very critical areas like payment cards, contactless keys, biometric information etc. Today, while the number of IoT technologies increases, security of these devices becomes very important.

In the last few decades, security of computing environment was mainly provided by Cryptographic applications like block ciphers. Cryptography is implemented on almost every security intensive application on quite powerful devices like laptops and personal computers. For IoT devices, Cryptography is again the first method coming to mind to provide security. However, generally IoT devices run on platforms with limited resource and limited computing power. Besides, most of the IoT devices are produced in extremely high volumes, which requires them to be cost effective. For all these reasons, cryptographic algorithms that are quite suitable for personal computers, are not so suitable for IoT devices. Therefore, cryptographic systems that are optimized for IoT devices that require less cost, less power, less energy and less memory are highly required in today's industry.

In the last few years, several lightweight cryptographic algorithms are published to satisfy industry needs. They are designed for hardware and software implementations of IoT devices. Therefore, lightweight cryptographic algorithms need less chip area and less energy for hardware implementations and need less memory and less coding for software implementations. However there are still more work required to design and optimize lightweight block ciphers considering some criteria like ease of coding, power consumption, side-channel resistance, and ease of implementation. PRESENT [14], LED [35], PRIDE [3], LBLOCK [74], RECTANGLE [78], TWINE [62], KLEIN [33] are some examples of the most widely known lightweight block ciphers.

1.4 Design of Lightweight Block Ciphers

As mentioned above there are several metrics that are considered when evaluating a block cipher as a lightweight block cipher. These metrics must be carefully inves-

tigated to meet the required design goals. Therefore, designing a lightweight block cipher requires some special phases like Specification, Design, Implement and Cryptanalysis. At the first phase, design criteria of the block cipher are specified with a required threshold value. Some of the examples of design criteria are shown below:

- Memory Consumption
- Power Consumption
- Chip Area
- Cost of one implementation
- Side Channel Resistance
- Latency
- Throughput

At this stage, required threshold values must also be specified for each design criteria. However, this specification is very much dependent on the platform on which the algorithm is implemented. For this reason, for software and hardware implementations the type of embedded micro processors (8bit or 32bit) and FPGAs are considered when determining the threshold values.

At the Design phase, block cipher algorithm is designed with respect to the design criteria specified in the Specification phase. At this stage, new design approaches can be thought without reducing the security of the cipher. The main design criteria that is considered during the design phase is latency. Latency is directly affected by the number of rounds. Therefore, when optimizing a cipher, unnecessary rounds can be removed without reducing the security of the cipher. In fact, instead of using traditional iterative round functions, using unrolled cipher designs decrease latency and the required computational power very much.

At the implementation phase, cipher is implemented as specified in the design phase. In this stage implementation costs are investigated by considering different implementation platforms. Results from this phase feeds back to design phase and some

changes can be applied to the design of the cipher. At the last phase, cryptanalysis studies are performed on implemented cipher. Security of the cipher is tested in this phase. Therefore, results from these tests feed back design phase which may cause design of the cipher to change.

1.5 Recent Studies in Lightweight Block Ciphers

With the emergence of the IoT technologies, new lightweight cryptographic algorithms were needed to suit the low computing resource constraint of IoT devices. For this reason, during last few years, several lightweight block ciphers have been proposed to satisfy this need. The main design objective of these ciphers is to provide enough security and performance with requiring less chip area, less energy consumption, less memory and less cost. We have investigated several lightweight block ciphers in our study and we have summarized them in Table 1.1.

During our research, we have seen that most of the lightweight block ciphers in the literature used SPN and Feistel Network structures. Besides, the rest of them are based on Add-Rotate-XOR (ARX) and NLFSR-based block ciphers. ARXs have only addition and rotation phases without using S-boxes. Although ARXs have fast implementations, their security is not studied as SPN and Feistel Networks. SPECK [8] and LEA [37] are some of the ARX type block ciphers. NLFSR-based ciphers are based on building blocks of stream ciphers and they are mostly used in hardware implementations. KATAN [19], KTANTAN [19] and HALKA [22] are examples of NLFSR based ciphers.

Table 1.1: Recent Lightweight Block Ciphers

Block Cipher	Publication Year	Block Size	Key Size	Rounds	Structure
KATAN [19]	2009	32,48,64	80	254	NLFSR
KTANTAN [19]	2009	32,48,64	80	254	NLFSR
TWIS [53]	2009	128	128	10	FEISTEL
KLEIN [33]	2011	64	64,80,96	12,16,20	SPN
LED [35]	2011	64	64,128	32,48	SPN
TWINE [62]	2011	64	80,128	36	FEISTEL
LBLOCK [74]	2011	64	80	32	FEISTEL
PICCOLO [58]	2011	64	80,128	25,31	FEISTEL
EPCBC [77]	2011	48,96	96	32	SPN
PICARO [54]	2012	128	-	12	SPN
PRINCE[17]	2012	64	128	12	SPN
SIMON [8]	2013	32 48 64 96 128	64 72,96 96,128 96,144 128,192,256	32 36 42,44 52,54 68,69,72	FEISTEL
SPECK [8]	2013	32 48 64 96 128	64 72,96 96,128 96,144 128,192,256	22 22,23 26,27 28,29 32,33,34	ARX
ZORRO [32]	2013	128	128	24	SPN
ITUbee[40]	2013	80	80	-	FEISTEL
LEA [37]	2013	128	128,192,256	24,28,32	ARX
RECTANGLE [78]	2014	64	80,128	25	SPN
FeW [46]	2014	64	80,128	32	FEISTEL
HALKA [22]	2014	64	80	24	NLFSR
ROBIN [34]	2014	128	128	16	SPN
FANTOMAS [34]	2014	128	128	12	FEISTEL
HISEC [2]	2014	64	80	15	FEISTEL
PRIDE [3]	2014	64	128	20	SPN
SIMECK [75]	2015	32,48,64	64,96,128	32,36,44	FEISTEL
MIDORI [6]	2015	64,128	128	16,20	SPN
MYSTERION [38]	2015	128,256		12,16	SPN
ROADRUNNER [7]	2015	64	80,128	10,12	FEISTEL
SKINNY [9]	2016	64 128	64,128,192 128,256,384	32,36,40 40,48,56	SPN
SPARX [24]	2016	64 128	128 128,256	24 32,40	ARX
MANTIS [9]	2016	64	128,64	14	SPN

In our study we have also seen that, lightweight block ciphers can be categorized according to some characteristics like throughput, power consumption, chip size for hardware implementations and code size for software implementations. In [29] software and hardware implementations of lightweight block ciphers are evaluated. In [51] and [50] recent lightweight block ciphers are evaluated according to hardware efficiency, software efficiency and energy consumption. In [25] software implementations are compared in different platforms.

In our study we have seen that PRIDE [3] is one of the best software efficient cipher after SPECK [8] proposed by NSA as shown in [50], [3] and [7]. SPECK [8] does not have linear layer and have more number of rounds to provide security, while PRIDE have very efficient linear layer with 20-rounds. This led us to study the security of PRIDE [3] cipher and in the rest of this thesis PRIDE [3] and its attacks are investigated in detail.

When we study the security of PRIDE block cipher, we have seen three differential cryptanalysis attacks performed on PRIDE in the literature. In the first two attacks [79] and [76] 18-round and 19-round key bits are captured, while in the last attack [21] entire key bits for full of 20-round PRIDE are captured. In this thesis, we have investigated these attacks in detail in Chapter 4.

Besides, during our study we have seen that the first Differential Fault Analysis on the block cipher PRIDE was performed on [47]. Differential Fault Analysis is a kind of Side-Channel attack [45] in which internal information of a chip can be derived by observing some external physical characteristics like power consumption, electromagnetic radiation or calculation time. In the Differential Fault Analysis introduced in [16], regular encryption process is disturbed and altered by injecting some faults by means of light pulses, laser or electromagnetic disruption. In the Differential Fault attacks of [47] authors were able to capture the full key by means of 4 faults which is performed by electromagnetic injection method. Because Differential Fault Attacks are beyond the scope of this thesis, there will not be more information provided about them.

Furthermore, during our research, we have encountered new cryptanalytic time-memory-data tradeoff attacks [26] on PRIDE block cipher. These attacks target the block

ciphers with FX-constructions like PRIDE block cipher. In FX-constructions introduced in [41], encryption keys are XORed with independent keys called whitening keys at the begining and at end of the encryption process. Therefore, total key size becomes sum of the size of the original key and the whitening keys. In these attacks, authors used Hellman's time-memory tradeoff model [36] and showed that the time complexity of the attacks specified in [30] was reduced.

1.6 Cryptanalysis of Block Ciphers

The encryption algorithm of a block cipher is not secret. Actually, the security of the block cipher is provided by keeping the key as secret. Therefore, plaintext information can be obtained from the corresponding ciphertext by decrypting it with the related key. For this reason, attackers try to capture the key material to obtain the plaintext information. However, if they can not access the key material, then they try several different methods to break the block cipher.

The most obvious method to attack the cipher is to try every possible key to decrypt the ciphertext. This method is known as *exhaustive search* or *brute force attack*. This attack can be performed by obtaining some plaintext, ciphertext pairs and encrypting these plaintexts with every possible key. If the ciphertext matches with the previously obtained ciphertext, then that key is identified as the correct key. This key is also tried on different plaintext-ciphertext pairs to be sure about its accuracy. Because the simplicity of this attack, this method can be used for every block cipher. In order to provide the security of a block cipher against exhaustive search attacks key space is kept large. In other words, the bit length of the key is kept as long as the computational power of the current technology is not enough to try every possible key in meaningful time. If the key length of a block cipher is n bits, then 2^n operations are required to perform an exhaustive search and it could be very time consuming.

Another attack method that requires less operations than exhaustive search is *table attack*. In this case, every possible corresponding plaintext - ciphertext pairs for encrypting key are obtained and stored in a database. Then decrypting a ciphertext requires only a database query operation that finds matching plaintext for a cipher-

text. The downside of this attack is, it requires too much space to keep all possible plaintext - ciphertext pairs. If the block size is b bits then 2^b data must be stored.

Because, exhaustive search attack takes very long time and table attack requires too much space, attackers can also use a more advanced attack called *Time-Memory Tradeoff Attack*. The Time-Memory Tradeoff Attack [36] was first suggested by Hellman and it requires less encryptions and less space to capture the key material. The main idea of this attack is to perform the exhaustive search in a clever way and store only a small part of the resulting tables. When performing the attack for k-bit key, attacker performs less than 2^k operations to obtain a value that is already in the table.

If an attack that captures encrypting key for a block cipher with less operations than *exhaustive search* and needed less data than *table attack*, then that block cipher is considered as broken.

In the rest of our thesis, we have studied *Differential Cryptanalysis* which is one of the best known cryptanalysis technique. Differential Cryptanalysis is one of the mostly used attack method that enables cryptanalysts to capture the key by investigating relations between the input differences and the corresponding output differences of a block cipher. However a recent study, *Differential Factors* [67] showed that it may not be possible to fully capture the attacked round key bits when performing a differential attack. Besides, another recent study *Undisturbed Bits* [64] can be used for discovering longer differential characteristics that provides more powerful differential attacks. Differential Cryptanalysis, Differential Factors and Undisturbed Bits are explained in detail in Chapter 2.

1.7 Attack Types

Attacks to block ciphers can be categorized also according to the information that is required to perform the attack:

- **Ciphertext-only attack (CO):** In this attack, attackers have only some ciphertext information. To perform a ciphertext-only attack, the cipher should have significant weaknesses (e.g. A5/1 Stream Cipher)

- **Known-plaintext attack (KP):** In this attack, attackers can get n plaintexts and the corresponding ciphertexts. *Linear Cryptanalysis* can be example of *Known-plaintext attack*.

- **Chosen-plaintext (ciphertext) attack (CP):** In this attack, the attacker is able to request the encryption of n plaintexts of his choosing and captures the corresponding ciphertexts.

- **Adaptive chosen-plaintext (ciphertext) attack (ACP):** In this attack, the attacker is able request encryptions of some plaintexts possibly seeing encryptions of some plaintexts first and making some calculations using them.

Collecting data becomes harder as we move down the list.

1.8 Complexity

Attacks can be expressed according to the resources they require. These resources are data complexity, time complexity and memory complexity.

- **Data Complexity:** The number of plaintext or ciphertext information that is required to perform the attack.

- **Time Complexity:** The amount of time required to perform the attack. It is calculated by the number of encryptions required to perform the attack.

- **Memory Complexity:** The amount of storage required to perform the attack.

In the case of exhaustive search, if the secret key is n bits, then the time complexity is expressed as 2^n encryptions.

In the case of table attack, data complexity and memory complexity can be expressed as 2^b. This attack's time complexity can be negligible.

1.9 Our Contribution and the Structure of the Thesis

In this thesis, we have investigated several lightweight block ciphers for the existence of Differential Factors and Undisturbed Bits. We have also shown how differential factors can be used to reduce the time complexity of differential attacks by summarizing the corrected attacks on PRESENT [14] and SERPENT [10] block ciphers in Chapter 2. Moreover, after investigating the structure of PRIDE [3] in Chapter 3, we have also investigated the 18-round [79], 19-round [76] and 20-round [21] differential attacks on PRIDE [3] block cipher and we have provided some corrections for these attacks considering differential factors in Chapter 4. On the 18-round attack, we showed that authors fail to discover differential factors that exist in the first and 17th round, that increase the time complexity from 2^{66} to 2^{70}. We have also presented that, on the 19-round attack authors fail to discover differential factors that exist in 2nd and 18th round which reduces time complexity from 2^{64} to 2^{63} and on the 20-round attack we have showed that attack needs 2^{52} encryptions exhaustive search not 2^{48}. We have published these corrections in [70] and [71].

CHAPTER 2

DIFFERENTIAL CRYPTANALYSIS OF BLOCK CIPHERS

In this thesis, we have shown that differential attacks performed on PRIDE [3] block cipher was wrong and we have corrected these attacks by using S-box properties *Differential Factors* and *Undisturbed Bits*. In our correction, we have seen that these attacks were required more time complexity values than it was claimed. We have presented corrected attacks of PRIDE in Chapter 4.

Therefore, in this chapter, we first explain the concept of *Differential Cryptanalysis* and mention about the S-box properties *Differential Factors* and *Undisturbed Bits*. Until now, by using these properties attacks performed only on PRESENT [14] and SERPENT [10] block ciphers were corrected and by this means, time complexity of attacks on SERPENT [10] was significantly reduced. We explain these corrected attacks of PRESENT [14] and SERPENT [10] in Section 2.3.1 and Section 2.4.1 in this chapter.

2.1 Differential Crytpanalysis

Differential cryptanalysis [13] was discovered by Biham and Shamir in late 1980s and it is used to attack various block ciphers, stream ciphers and hash functions. Differential cryptanalysis is a chosen plaintext attack and it investigates the relations of input differences and the corresponding output differences of encryption process.

Notation:

- I : Plaintext 1
- I': Plaintext 2
- O : Ciphertext of plaintext 1
- O': Ciphertext of plaintext 2
- N: Number of plaintext and ciphertext pairs
- ΔI : Input Difference ($\Delta I = I \oplus I'$), (Difference values are presented in Hexadecimal format)
- ΔO : Output Difference ($\Delta O = O \oplus O'$), (Difference values are presented in Hexadecimal format)
- p_0 : Probability of difference
- r: Number of Rounds

If we explain more clearly, when two different plaintext information encrypted with the same key, difference values of plaintext information and difference values of corresponding ciphertext information are investigated in differential cryptanalysis. Difference values are calculated by mathematically XOR operation of two values, in this case they are plaintext and ciphertext values. If input difference value leads to some output difference value with some probability greater than expected after r rounds, this is called as differential characteristic. Differential characteristic can be used to guess some parts of the secret key. It can also be used to distinguish block cipher encryption from random permutation. These subjects related to differential cryptanalysis are explained in detail below:

- **Differential Characteristic:** Let two inputs such as I and $I^{'}$ are XORed. The result of this calculation is called as input difference and expressed as ΔI. Corresponding outputs such as O and $O^{'}$ are also XORed and the result of this calculation is called as output difference, and expressed as ΔO. If after r rounds,

ΔI input difference causes ΔO output difference with some probability p_0, it is called as differential characteristic.

If the round operations are linear operations, then we can exactly know their effect on the difference. However, result of the non-linear operations depend on the input and we can trace the difference with some probability. If an S-box is used and substitution operation is performed as a non-linear operation, Difference Distribution Table (DDT) is used to determine which input differences lead to which output difference in what probability. Therefore, probability value of differential characteristic is determined by using Difference Distribution Table (DDT) if S-box is the only non-linear part of the block cipher. DDT is explained in detail in Cryptanalysis of PRESENT example later in this section. The structure of PRESENT can be seen in Figure 1.4.

- **Effects of the Key Addition, Substitution and Permutation layers when Constructing Differential Characteristic:**

  - **Key Addition:** We get $(I \oplus k)$ and $(I' \oplus k)$. Since both inputs are encrypted with the same key, their difference is still the same: $(I \oplus k) \oplus (I' \oplus k) = I \oplus I' = \Delta I$

  - **Substitution:** We don't know the exact values of $S(I \oplus k)$ and $S(I' \oplus k)$. So we can not exactly know the difference $S(I \oplus k) \oplus S(I' \oplus k)$. But we can analyze the S-box to see which input differences provide which output differences.

  - **Permutation:** Permutation layer is the linear layer of the block cipher. This layer can include different operations like changing bit positions or matrix multiplications. Because this operation is linear, it is known that if the input has a nonzero difference at the i-th bit, then the output has a nonzero difference at the P(i)-th bit.

- **Distinguishing Block Cipher:** A differential characteristic can be used to distinguish random encryption from the specific block cipher by encrypting N plaintext with fixed key and comparing the ΔI and ΔO results with the differential characteristic. We expect to observe differential characteristic in N

plaintext encryption with an *expected value* which is calculated by multiplying differential characteristic probability and number of pairs.

- **Guessing Secret Key:** Differential characteristic can also be used to guess some parts of the subkeys. In order to do that, one or more rounds of encryption are added to the before or after of r round differential. For example; let's say we have 4x4 simple S-box and we have 4 rounds differential characteristic such as ΔI = 00000001 and ΔO=00070000. We add one round encryption above the 4 round differential. We must find out possible input differential values for the newly added round by investigating the DDT tables of the related S-boxes. So that we determine the input difference values that causes to ΔI=000000001 after one round. Let's say these differences are (4,7,9,A). For the next phase, we pick random input values that have difference of 4,7,9 and A values. Next, we encrypt these input pairs for five round with every possible key and observe the ΔO values. If we detect that ΔO equals to 00070000, we increase the counter of that key. After trying all possible keys, correct key must have the greatest counter. So that, we can guess some parts of the subkeys. For the next parts of the subkeys, we use exhaustive search.

2.1.1 Types of Differential Cryptanalysis

Differential cryptanalysis is one of the most important techniques to investigate the security of a block ciphers. So that, designers try to develop block ciphers that are more resistant to differential cryptanalysis.

Since its discovery, many variations of differentiial cryptanalysis are introduced.

- **Truncated Differential Cryptanalysis** [44] Truncated Differential Cryptanalysis guesses only part of the difference in a pair of texts after each round of encryption. More than one truncated differential characteristic can be used together to decrease the attack's time complexity. Some of the examples of the block ciphers that Truncated Differential Cryptanalysis is applied are CRYPTON [42] and SAFER [73] block ciphers.

- **Higher Order Differential Cryptanalysis** [44] Differential cryptanalysis investigates difference between two inputs, but Higher Order Differential Cryptanalysis investigates effects of a number of differences between a larger set of inputs. Some of the examples of the block ciphers that Higher Order Differential Cryptanalysis is applied are MISTY1 [5], CAST [52] and SHA-256 [48].

- **Impossible Differential Cryptanalysis** [11] While regular differential cryptanalysis investigates differences that are greater than expected probability, impossible differential cryptanalysis investigates differences that are impossible, in other words some determined differential characteristic can not exist for the correct key. The probability of such differentials should be 0. Therefore, any candidate key that is tried on cryptanalysis operation satisfy the impossible differential characteristic can not be the correct key. CLEFIA [59], SIMON [18], CAMELLIA [18] and AES [49] are some of the block ciphers that had Impossible Differential Cryptanalysis attacks.

- **Improbable Differential Cryptanalysis** [65] Improbable differential cryptanalysis investigates the differences that are less likely exist for the correct key than a wrong key. Impossible differential cryptanalysis is a subset of this attack. Improbable Differential Cryptanalysis is applied on CLEFIA [66] and PRESENT [64] block ciphers.

2.2 Differential Factors

In a differential attack, every possible key are tried on plaintext pairs expecting to satisfy differential characteristic which allows us to guess the correct subkeys. It is expected that correct key must satisfy the differential characteristic more times than any other key, so that we can distinguish the correct key from the wrong ones. However, in certain cases, output difference of the S-box operation may be invariant when the round key is XORed with some specific value. Therefore, some candidate keys can satisfy the differential characteristic for an equal number of times. Such a case would prevent the attacker from fully capturing the round key. Differential Factors [67] have been first described by Tezcan and they are defined as follows:

Definition 1 (Differential Factor [67]) *Let S be a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2^m$. For all $x, y \in \mathbb{F}_2^n$ that satisfy $S(x) \oplus S(y) = \mu$, if we also have $S(x \oplus \lambda) \oplus S(y \oplus \lambda) = \mu$, then we say that the S-box has a* differential factor *λ for the output difference μ. (i.e. μ remains invariant for λ).*

It is useful to note the following additional properties of differential factors.

Theorem 1 *([67]) 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 an output difference λ.*

Theorem 2 *([67]) If λ_1 and λ_2 are differential factors for an ouput 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.*

Differential factors affect differential cryptanalysis when a block cipher contains key XOR process prior to S-box substitution process. This structure is generally used in SPN block ciphers. In order to see the effect of differential factors, there are some conditions that must be satisfied. These conditions are listed as follows:

- There must be a differential factor λ for output difference μ for an S-box activated by the attack.
- The differential being used in the attack requires the output difference of this S-box to be μ.

During regular differential cryptanalysis, candidate keys are tried to discover the correct keys. Correct key k, must satisfy the differential characteristic more times than any other key. In other words, when correct key k is tried during attack, S-box produces μ more times than any other key. The problem arises at this point if there is a differential factor. In other words, for any plaintext/ciphertext pair when k is tried and μ is obtained from S-box substitution, $k \oplus \lambda$ also produce the same result which is μ again. This situation can also be observed from the counter values of candidate keys in that counter value of k and counter value of $k \oplus \lambda$ become equal. This means

that, the correct candidate key k, is indistinguishable from an incorrect candidate key containing $k \oplus \lambda$.

As a result, during key guess step it is not possible to discover the key bits where differential factors exist. These key bits must be discovered with the exhaustive search. This causes that the time complexity of the key guess step is decreased and because additional bits have to be discovered, time complexity of exhaustive search step is increased.

This behavior is not only limited to the encryption process. It is also observed during the decryption process.

Theorem 3 *([67]) In a block cipher let an S-box S contain a differential factor λ for an output difference μ and the partial round key k is XORed with the input of S. If an input pair provides the output difference μ under a partial subkey k, then the same output difference is observed under the partial subkey $k \oplus \lambda$. Therefore, during a differential attack involving the guess of a partial subkey corresponding to the output difference μ, the advantage of the cryptanalyst is reduced by 1 bit and the time complexity of this key guess step is halved.*

Corollary 1 *([67]) 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.*

Corollary 2 *([63]) 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.*

During our research, we have investigated several block ciphers and discovered the existence of differential factors on most of them. The S-boxes of block ciphers and related differential factors are listed in Table 2.1

Table 2.1: Differential Factors of Some Block Ciphers

Block Cipher	S-box	Differential Factors
PRIDE[3]	**S[x]:** 0, 4, 8, F, 1, 5, E, 9, 2, 7, A, C, B, D, 6, 3	λ=1 μ=1 λ=8 μ=8
PRESENT[14]	**S[x]:** C, 5, 6, B, 9, 0, A, D, 3, E, F, 8, 4, 7, 1, 2	λ=1 μ=5 λ=F μ=F
LBLOCK[74]	**s0:** E, 9, F, 0, D, 4, A, B, 1, 2, 8, 3, 7, 6, C, 5 **s1:** 4, B, E, 9, F, D, 0, A, 7, C, 5, 6, 2, 8, 1, 3 **s2:** 1, E, 7, C, F, D, 0, 6, B, 5, 9, 3, 2, 4, 8, A **s3:** 7, 6, 8, B, 0, F, 3, E, 9, A, C, D, 5, 2, 4, 1 **s4:** E, 5, F, 0, 7, 2, C, D, 1, 8, 4, 9, B, A, 6, 3 **s5:** 2, D, B, C, F, E, 0, 9, 7, A, 6, 3, 1, 8, 4, 5 **s6:** B, 9, 4, E, 0, F, A, D, 6, C, 5, 7, 3, 8, 1, 2 **s7:** D, A, F, 0, E, 4, 9, B, 2, 1, 8, 3, 7, 5, C, 6	**s0,** λ=3 μ=4 **s0,** λ=B μ=1 **s1,** λ=3 μ=4 **s1,** λ=B μ=2 **s2,** λ=3 μ=1 **s2,** λ=B μ=2 **s3,** λ=3 μ=8 **s3,** λ=B μ=1 **s4,** λ=3 μ=2 **s4,** λ=B μ=1 **s5,** λ=3 μ=2 **s5,** λ=B μ=1 **s6,** λ=3 μ=4 **s6,** λ=B μ=2 **s7,** λ=3 μ=4 **s7,** λ=B μ=2
NOEKEON	**S[x]:** 7, A, 2, C, 4, 8, F, 0, 5, 9, 1, E, 3, D, B, 6	λ=1 μ=1 λ=B μ=B
PICCOLO[58]	**S[x]:** E, 4, B, 2, 3, 8, 0, 9, 1, A, 7, F, 6, C, 5, D	λ=1 μ=2 λ=2 μ=5
RECTANGLE[78]	**S[x]:** 6, 5, C, A, 1, E, 7, 9, B, 0, 3, D, 8, F, 4, 2	λ=2 μ=4 λ=E μ=C
	CONTINUED ON NEXT PAGE	

	Table 2.1 CONTINUED FROM PREVIOUS PAGE	
SARMAL[80]	Hash Function	λ=F μ=4 λ=A μ=9
SPONGENT[15]	Hash Function	**s1,** λ=F μ=9 **s2,** λ=1 μ=F
GOST [28]	**s0:** 4, A, 9, 2, D, 8, 0, E, 6, B, 1, C, 7, F, 5, 3 **s1:** E, B, 4, C, 6, D, F, A, 2, 3, 8, 1, 0, 7, 5, 9 **s2:** 5, 8, 1, D, A, 3, 4, 2, E, F, C, 7, 6, 0, 9, B **s3:** 7, D, A, 1, 0, 8, 9, F, E, 4, 6, C, B, 2, 5, 3 **s4:** 6, C, 7, 1, 5, F, D, 8, 4, A, 9, E, 0, 3, B, 2 **s5:** 4, B, A, 0, 7, 2, 1, D, 3, 6, 8, 5, 9, C, F, E **s6:** D, B, 4, 1, 3, F, 5, 9, 0, A, E, 7, 6, 8, 2, C **s7:** 1, F, D, 0, 5, 7, A, 4, 9, 2, 3, E, 6, B, 8, C	**s0,** λ=5 μ=3 **s3,** λ=D μ=5 **s5,** λ=9 μ=B **s7,** λ=7 μ=5 **s7,** λ=E μ=6
HAMSI[43]	**S[x]:**8, 6 , 7, 9, 3, C, A, F, D, 1, E, 4, 0, B, 5, 2	λ=2 μ=1 λ=4 μ=D
LED[35]	**S[x]:**C, 5, 6, B, 9, 0, A, D, 3, E, F, 8, 4, 7, 1, 2	λ=1 μ=5 λ=F μ=F
JOLTIK V1	**S[x]:**E, 4, B, 2, 3, 8, 0, 9, 1, A, 7, F, 6 C, 5, D	λ=1 μ=2 λ=2 μ=5
LAC V1	**S[x]:**E, 9, F, 0, D, 4, A, B, 1, 2, 8, 3, 7, 6, C, 5	λ=B μ=1 λ=3 μ=4
PROST V1	**S[x]:**0, 4, 8, F, 1, 5, E, 9, 2, 7, A, C, B, D, 6, 3	λ=1 μ=1 λ=8 μ=8
FOX [39]	**s0:** 2, 5, 1, 9, E, A, C, 8, 6, 4, 7, F, D, B, 0, 3 **s1:** B, 4, 1, F, 0, 3, E, D, A, 8, 7, 5, C, 2, 9, 6 **s2:** D, A, B, 1, 4, 3, 8, 9, 5, 7, 2, C, F, 0, 6, E	**s2,** λ=5 μ=8 **s2,** λ=1 μ=13

As can be seen from the Table 2.1, differential factors exist on most of the block ciphers and the existence of differential factors depend on S-box design. Although, differential factors are not applicable to all differential attacks, they directly affect the time and memory complexity of them when they exist. Therefore, differential factors should be considered during differential attacks.

In the next part of this Section, we have explained the examples of Differential Cryptanalysis and Differential Factors on PRESENT and SERPENT block ciphers. We have also shown corrected attacks of PRESENT and SERPENT by considering differential factors.

2.3 Example: Differential Cryptanalysis of PRESENT

PRESENT [14] is a 31-round SPN (Substitution Permutation Network) type block cipher with block size of 64 bits that supports 80 and 128-bit secret key. At each round, as can be seen in Figure 2.1 64-bit input of the round function is XORed with the subkey, then 16 4×4-bit S-boxes are used to provide confusion and then finally permutation is performed to provide diffussion.

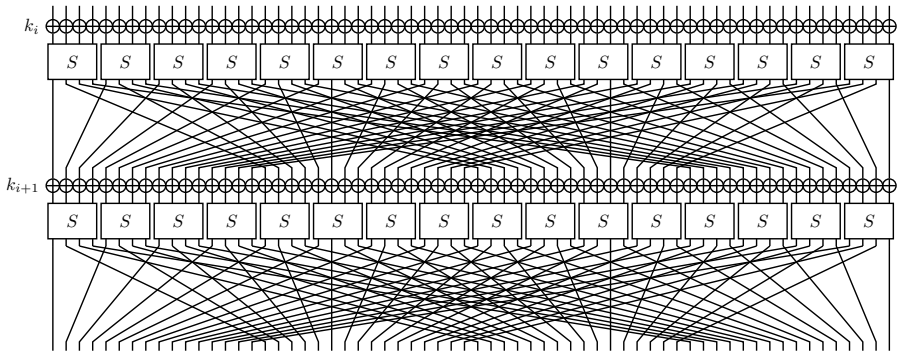

Figure 2.1: Round function of PRESENT

PRESENT's S-BOX is shown in Table 2.2.

Table 2.2: S-box of PRESENT

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

Difference Distribution Table (DDT) of an S-box can be prepared by considering every possible (x; y) input pairs with $x \oplus y = i$ and count the output differences where $S(x) \oplus S(y) = j$ and construct a table with this count as the ij-th entry. DDT of PRESENT's S-box can be seen in Table 2.3.

Table 2.3: Difference Distribution Table of PRESENT

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

The highest values in the DDT (except the first entry) is called differential uniformity. High differential uniformity provides characteristics with high probability, hence lower is better. Values in DDT is even because the pairs (x; y) and (y; x) provides the same difference. Thus, the theoretically best achievable differential uniformity is 2. PRESENT's S-box is differentially 4 uniform.

The best known differential attack on PRESENT is provided in [72] which breaks 16 round by adding two rounds to the bottom of 14 round differential characteristic that is shown in Table 2.4.

Δ_1: 0700000000000700 $\rightarrow_{14r}$ 0000000900000009 with probability: 2^{-62}.

Table 2.4: 14-round Differential Characteristic of PRESENT

Rounds		**Differences**	**Probability**
Input	I	$x_2 = 7$, $x_{14} = 7$	
R1	S	$x_2 = 1$, $x_{14} = 1$	2^{-4}
R1	P	$x_0 = 4$, $x_3 = 4$	1
R2	S	$x_0 = 5$, $x_3 = 5$	2^{-4}
R2	P	$x_0 = 9$, $x_8 = 9$	1
R3	S	$x_0 = 4$, $x_8 = 4$	2^{-4}
R3	P	$x_8 = 1$, $x_{10} = 1$	1
R4	S	$x_8 = 9$, $x_{10} = 9$	2^{-4}
R4	P	$x_2 = 5$, $x_{14} = 5$	1
R5	S	$x_2 = 1$, $x_{14} = 1$	2^{-6}
R5	P	$x_0 = 4$, $x_3 = 4$	1
R6	S	$x_0 = 5$, $x_3 = 5$	2^{-4}
R6	P	$x_0 = 9$, $x_8 = 9$	1
R7	S	$x_0 = 4$, $x_8 = 4$	2^{-4}
R7	P	$x_8 = 1$, $x_{10} = 1$	1
R8	S	$x_8 = 9$, $x_{10} = 9$	2^{-4}
R8	P	$x_2 = 5$, $x_{14} = 5$	1
R9	S	$x_2 = 1$, $x_{14} = 1$	2^{-6}
R9	P	$x_0 = 4$, $x_3 = 4$	1
R10	S	$x_0 = 5$, $x_3 = 5$	2^{-4}
R10	P	$x_0 = 9$, $x_8 = 9$	1
R11	S	$x_0 = 4$, $x_8 = 4$	2^{-4}
R11	P	$x_8 = 1$, $x_{10} = 1$	1
R12	S	$x_8 = 9$, $x_{10} = 9$	2^{-4}
R12	P	$x_2 = 5$, $x_{14} = 5$	1
R13	S	$x_2 = 1$, $x_{14} = 1$	2^{-6}
R13	P	$x_0 = 4$, $x_3 = 4$	1
R14	S	$x_0 = 5$, $x_3 = 5$	2^{-4}
R14	P	$x_0 = 9$, $x_8 = 9$	1

Two rounds are added to the bottom of this characteristic, the output difference should be in the form 0?0?0?0?0?0?0000. Overall attack procedure is shown below:

- **Data Collection:** Gather N plaintext ciphertext pairs with input difference 0700000000000700 and output difference 0?0?0?0?0?0?0000 after 16 rounds.
- **Key Guess:** Partially decrypt these pairs with every possible key bits of rounds 16 and 15 that correspond to S-boxes with nonzero difference. Correct key should have the highest counter.
- **Exhaustive Search:** Remaining key bits are obtained by exhaustive search.

This attack captures 32 bits of the key with $2^{33.18}$ 2-round encryptions. Remaining 48 bits require 2^{48} 16-round encryptions.

2.3.1 Corrected Attacks on PRESENT

In the attack [72] authors claimed to capture 32 bits with $2^{33.18}$ 2-round encryptions, and remaining 48 bits are captured with 2^{48} 16-round encryptions via exhaustive search.

Table 2.5: Differential Factors of PRESENT

Differential Factor	**Output Difference**
$\lambda = 1$	$\mu = 5$
$\lambda = F$	$\mu = F$

However this attack is corrected in [63] that authors fail to discover 6 differential factors which are shown in Table 2.6.

Therefore, as explained in [63], the number of bits that are actually captured is 26 bits not 32 bits which require $2^{27.18}$ 2-round encryptions and remaining 54 bits require 2^{54} 16-round encryptions. Thus, the time complexity of this attack is 2^{54} not 2^{48}.

Table 2.6: 16-round differential-linear attack of [72]. Values that need to be obtained are shown in bold.

	Differences in bits															
Rounds	x_{15}	x_{14}	x_{13}	x_{12}	x_{11}	x_{10}	x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0
$X_{1,I}$	0000	0111	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0111	0000	0000
14-Round Differential Δ_1																
$X_{14,P}$	0000	0000	0000	0000	0000	0000	0000	1001	0000	0000	0000	0000	0000	0000	0000	1001
$X_{15,S}$	0000	0000	0000	0000	0000	0000	0000	???0	0000	0000	0000	0000	0000	0000	0000	???0
$X_{15,P}$	0000	**000?**	0000	**000?**	0000	**000?**	0000	**000?**	0000	**000?**	0000	**000?**	0000	0000	0000	0000
$X_{16,S}$	0000	????	0000	????	0000	????	0000	????	0000	????	0000	????	0000	0000	0000	0000

We have also provided further correction for this attack in [70] by considering overlooked Undisturbed Bits.

2.4 Example: Differential Cryptanalysis of SERPENT

SERPENT [10] was designed by Anderson, Biham and Knudsen as the candidate for Advanced Encryption Standard in 1998 and became one of the finalists in the contest. It is an SPN type block cipher with 128-bit block size, 32 rounds and 256-bit key length. Key length of Serpent can be any size between 64 and 256 bits. Keys that are shorter than 256 bits are completed to 256 bit length, by having "1" bit to the top and as many "0" bits as required. The S-boxes of SERPENT is shown below:

Table 2.7: S-boxes of SERPENT

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S0	3	8	15	1	10	6	5	11	14	13	4	2	7	0	9	12
S1	15	12	2	7	9	0	5	10	1	11	14	8	6	13	3	4
S2	8	6	7	9	3	12	10	15	13	1	14	4	0	11	5	2
S3	0	15	11	8	12	9	6	3	13	1	2	4	10	7	5	14
S4	1	15	8	3	12	0	11	6	2	5	4	10	9	14	7	13
S5	15	5	2	11	4	10	9	12	0	3	14	8	13	6	7	1
S6	7	2	12	5	8	4	6	11	14	9	1	15	13	3	10	0
S7	1	13	15	0	14	8	2	11	7	4	12	10	9	3	5	6

The general design of Serpent can be specified as follows:

The 128-bit input value contains four 32-bit words X_0, X_1, X_2, X_3 and it is represented as $\hat{B}_i$ where $i \in \{0, ..., 31\}$. At each 32 rounds of Serpent Key Mixing, Substitution and Linear Transformation operations are performed.

- Key Mixing: At each round R_i, subkey generated by the key schedule K_i is XORed with the input value of $\hat{B}_i$.
- Substitution: S-box operation are applied on four 32-bit words X_0, X_1, X_2, X_3 that had key mixing operation. 32 copies of S-Box operation are executed simultaneously for each 4-bit inputs and 128-bit output value is created. This operation can be represented as $S_i(B_i \oplus K_i)$.
- Linear Transformation: The four 32-bit words X_0, X_1, X_2, X_3 are linearly mixed by the following operations:

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

where $\lll$ denotes rotation to the left, $\ll$ denotes shift to the left for the specified number of bits. Formal description of design of Serpent is shown as follows:

$$
\begin{aligned}
\hat{B}_0 &= IP(P) \\
\hat{B}_{i+1} &= R_i(\hat{B}_i) \\
C &= FP(\hat{B}_r)
\end{aligned}
$$

where;

IP = Initial Permutation

FP = Final Permutation

$P = \hat{B}_0$

$C = \hat{B}_{32}$

$$R_i(X) = L(\hat{S}_i(X \oplus \hat{K}_i)) \qquad i = 0, ..., r-2$$
$$R_i(X) = \hat{S}_i(X \oplus \hat{K}_i) \oplus \hat{K}_r \qquad i = r-1$$

where $\hat{S}_i$ represents S-box operation and L represents Linear Transformation.

The differential-linear attack on SERPENT was provided in [12] in which 11-round of SERPENT was attacked. The attack was based on a 3-round differential characteristic with probability of 2^{-7} which is shown below.

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

In the attack [12] 48 bits of the key are captured with the time complexity of $2^{139.2}$ 11-round SERPENT encryptions.

The time complexity of the attack [12] was further improved in [27] and 48 bits of the key was captured with the time complexity of $2^{135.7}$. Moreover, in the same publication first 12-round attack on SERPENT was provided. In the 12-round attack 160 bits of the key are captured with the time complexity of $2^{249.4}$.

2.4.1 Corrected Attacks on SERPENT

In the attack [27], authors could not consider the existing differential factors which can be seen in Table 2.8. Therefore, 11-round attack is corrected in [68] by using differential factors. In the corrected attack it was shown that 5 S-boxes are activated by 3-round differential characteristic and 2 of them had a output differences of 4_x and E_x which have differential factors as it is shown in 2.8. Therefore, at each round only 18 bits of the subkey can actually be captured, instead of 20 bits as it was claimed in [27]. Therefore, totally only 46 bits of the key can be captured instead of 48 bits and corrected time complexity is $2^{133.7}$ not $2^{135.7}$. Besides, in the same publication 12-

round attack was also corrected by using the same differential factors in that 157 bits of the key could actually be captured instead of 160 bits and correct time complexity is $2^{249.4}$ not $2^{246.4}$.

Table 2.8: Differential Factors of SERPENT

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

2.5 Undisturbed Bits

Undisturbed bits [64] have been first described by Tezcan and they are defined as follows:

Definition 2 (Undisturbed Bits [64]) *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.*

Undisturbed Bits are observed if some bits of the output differences of an corresponding input difference does not change. For example, when we investigate the Difference Distribution Table of PRIDE block cipher which is presented in Chapter 4, we can see that input difference of 1_x leads to output differences of 4_x (0100), 5_x (0101), 6_x (0110) and 7_x (0111). We can also see from these results that the first two bits of (10??) are the same for all output differences. In this case, the first two bits (10??) are called as Undisturbed Bits.

Undisturbed bits can be used for discovering longer differential characteristics which leads to more effective differential attacks. When we review the literature, we see that Undisturbed Bits are used in the differential attacks of PRESENT [64] and SERPENT [69] block ciphers.

We have analyzed several block ciphers such as PRESENT, PRIDE, LBLOCK, LUFFA, NOEKEON, PICCOLO, RECTANGLE, SARMAL, SERPENT, SPONGENT, GOST, HAMSI, LED, JOLTIKv1, LACv1, PROSTv1 and FOX for the existence of undisturbed bits and they can be seen in Table 2.9.

Table 2.9: Undisturbed Bits of Some S-boxes

Block Cipher	S-box	**Input**	**Output**
PRIDE	0,4,8,F,1,5,E,9,2,7,A,C,B,D,6,3	1_x	01??
		2_x	1???
		3_x	1???
		8_x	?0??
		9_x	01??
PRESENT	C,5,6,B,9,0,A,D,3,E,F,8,4,7,1,2	1_x	???1
		9_x	???0
LBLOCK	E,9,F,0,D,4,A,B,1,2,8,3,7,6,C,5	1_x	???1
		2_x	???1
		3_x	??10
		8_x	??1?
		B_x	??0?
LBLOCK	4,B,E,9,F,D,0,A,7,C,5,6,2,8,1,3	1_x	??1?
		2_x	??1?
		3_x	??01
		8_x	???1
		B_x	???0
LBLOCK	1,E,7,C,F,D,0,6,B,5,9,3,2,4,8,A	1_x	??1?
		2_x	??1?
		3_x	1?0?
		8_x	1???
		B_x	0???
LBLOCK	7,6,8,B,0,F,3,E,9,A,C,D,5,2,4,1	1_x	???1
		2_x	???1
		3_x	???0
		8_x	?1??
		B_x	?0??
LBLOCK	E,5,F,0,7,2,C,D,1,8,4,9,B,A,6,3	1_x	???1
		2_x	???1
		3_x	1??0
		8_x	1???
		B_x	0???
LBLOCK	2,D,B,C,F,E,0,9,7,A,6,3,1,8,4,5	1_x	???1
		2_x	???1
		3_x	?1?0
		8_x	?1??
		B_x	?0??
	CONTINUED ON NEXT PAGE		

	Table 2.9 **CONTINUED FROM PREVIOUS PAGE**		
LBLOCK	B,9,4,E,0,F,A,D,6,C,5,7,3,8,1,2	1_x	??1?
		2_x	??1?
		3_x	??01
		8_x	???1
		B_x	???0
LBLOCK	D,A,F,0,E,4,9,B,2,1,8,3,7,5,C,6	1_x	??1?
		2_x	??1?
		3_x	??01
		8_x	???1
		B_x	???0
NOEKEON	7,A,2,C,4,8,F,0,5,9,1,E,3,D,B,6	1_x	11??
		8_x	0???
		9_x	1???
		A_x	?1??
		B_x	?0??
PICCOLO	E,4,B,2,3,8,0,9,1,A,7,F,6,C,5,D	1_x	10??
		2_x	0???
		3_x	1???
		8_x	?1??
		9_x	?1??
RECTANGLE	6,5,C,A,1,E,7,9,B,0,3,D,8,F,4,2	1_x	??1?
		4_x	??11
		5_x	??0?
		8_x	???1
		C_x	???0
SERPENT	C,5,6,B,9,0,A,D,3,E,F,8,4,7,1,2	1_x	???1
		8_x	???1
		9_x	???0
GOST	4,A,9,2,D,8,0,E,6,B,1,C,7,F,5,3	-	-
GOST	E,B,4,C,6,D,F,A,2,3,8,1,0,7,5,9	-	-
GOST	5,8,1,D,A,3,4,2,E,F,C,7,6,0,9,B	-	-
GOST	7,D,A,1,0,8,9,F,E,4,6,C,B,2,5,3	8_x	1???
GOST	6,C,7,1,5,F,D,8,4,A,9,E,0,3,B,2	-	-
GOST	4,B,A,0,7,2,1,D,3,6,8,5,9,C,F,E	2_x	??1?
GOST	D,B,4,1,3,F,5,9,0,A,E,7,6,8,2,C	9_x	???1
GOST	1,F,D,0,5,7,A,4,9,2,3,E,6,B,8,C	-	-
HAMSI	8,6,7,9,3,C,A,F,D,1,E,4,0,B,5,2	2_x	???1
		8_x	???1
		A_x	???0
	CONTINUED ON NEXT PAGE		

	Table 2.9 **CONTINUED FROM PREVIOUS PAGE**		
LED	C,5,6,B,9,0,A,D,3,E,F,8,4,7,1,2	1_x 8_x 9_x	???1 ???1 ???0
JOLTIK V1	E,4,B,2,3,8,0,9,1,A,7,F,6,C,5,D	1_x 2_x 3_x 8_x 9_x	10?? 0??? 1??? ?1?? ?1??
LAC V1	E,9,F,0,D,4,A,B,1,2,8,3,7,6,C,5	1_x 2_x 3_x 8_x B_x	???1 ???1 ??10 ??1? ??0?
PROST V1	0,4,8,F,1,5,E,9,2,7,A,C,B,D,6,3	1_x 2_x 3_x 8_x 9_x	01?? 1??? 1??? ?0?? ?1??
FOX	2,5,1,9,E,A,C,8,6,4,7,F,D,B,0,3	-	-
FOX	B,4,1,F,0,3,E,D,A,8,7,5,C,2,9,6	1_x 4_x 5_x	??1? ??1? ??0?
FOX	D,A,B,1,4,3,8,9,5,7,2,C,F,0,6,E	-	-

CHAPTER 3

OVERVIEW OF PRIDE

In this thesis, we have corrected differential attacks performed on PRIDE [3] block cipher by considering differential factors. We have presented these attack details and our corrections in Chapter 4. Therefore, in this chapter we have analyzed the design features of PRIDE [3] block cipher.

3.1 PRIDE

3.1.1 Description

PRIDE [3] is a lightweight block cipher designed by Albrecht in CRYPTO 2014. The designers proposed a method to develop a good linear layer, which provides optimal design between security and efficiency. Therefore, it performs well in software and hardware implementations.

PRIDE is an SPN type block cipher with 64-bit block size, 128-bit key, and 20 rounds. Except the last round, round function R is used for the first 19 rounds which consists of successive key addition, substitution and linear layers. The last round function R' omits the linear layer. The overall structure of PRIDE is shown in Figure 3.1.

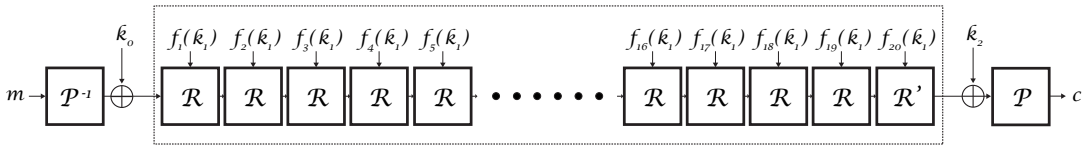

Figure 3.1: Overall structure of PRIDE

3.1.2 Linear Layer

Inside the round function of PRIDE as can be seen in Figure 3.2, at first 64-bit input XORed with the round key, then split into 16 4-bit nibbles and put into the S-box. And then the result is permuted and processed by the linear layer.

The linear layer L of PRIDE can be divided into three sub-layers, a permutation layer P, a matrix layer M and another permutation layer P^{-1} which is the inverse of P. L and M can be explained as:

$$L : P^{-1} \circ \mathrm{M} \circ \mathrm{P}$$

$$M\text{: } L_0 \text{ x } L_1 \text{ x } L_2 \text{ x } L_3$$

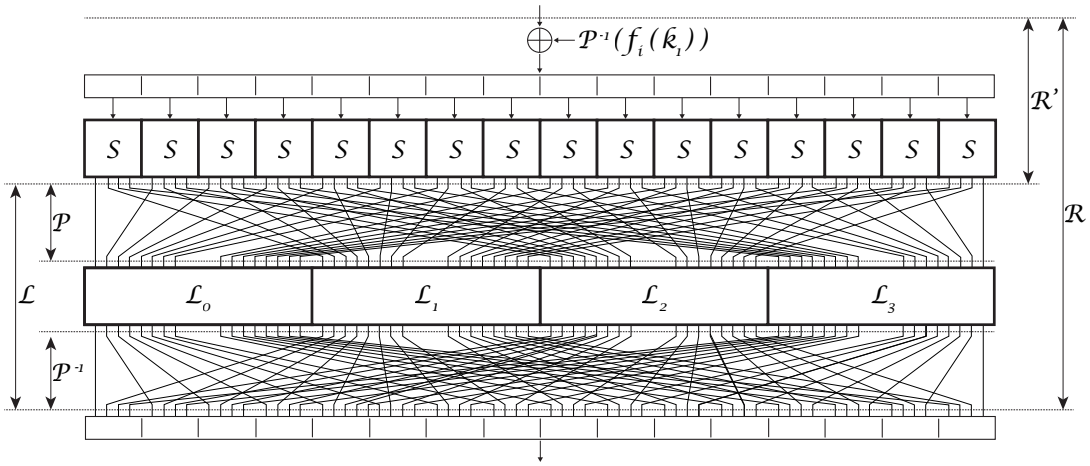

Figure 3.2: Round Function of PRIDE

In matrix layer M, 64-bit block is divided by 4 groups, and each group is multiplied by separate constant matrix. Designers searched the optimal matrices in [3] to have a very efficient linear layer. As a result, PRIDE outperforms recent lightweight block ciphers both in terms of code size and cycle count as can be seen in Section 3.1.5.

Linear layer matrices L_0, L_1, L_2, L_3 and their inverses are defined as follows:

$$L_0 = L_0^{-1} = \begin{pmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0
\end{pmatrix}$$

$$L_3 = L_3^{-1} = \begin{pmatrix}
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{pmatrix}$$

$$L_1=\begin{pmatrix}
1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&1&1&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&1&0&1&0&0&0&0&0&0\\
1&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&1&1&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&1\\
0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1\\
0&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&1&1&0&0&0&0
\end{pmatrix}$$

$$L_2=\begin{pmatrix}
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&1&1&0&1&0&0&0&0&0&0&0\\
0&0&0&0&0&0&1&1&0&1&0&0&0&0&0&0\\
1&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0\\
1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0\\
0&1&1&0&0&0&0&0&0&0&0&0&1&0&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&1&0&0\\
0&0&0&1&1&0&0&0&0&0&0&0&0&0&1&0\\
0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&1\\
0&0&0&0&0&1&0&0&1&1&0&0&0&0&0&0\\
0&0&0&0&0&0&1&0&0&1&1&0&0&0&0&0\\
0&0&0&0&0&0&0&1&0&0&1&1&0&0&0&0\\
1&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0\\
0&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0\\
0&0&1&0&0&0&0&0&0&0&0&0&0&1&1&0\\
0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&1
\end{pmatrix}$$

$$L_1^{-1}=\begin{pmatrix}
0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&0\\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1\\
1&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&1&1&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&1&1&0&0&0&0&0&0&1&0&0\\
0&0&0&1&0&0&0&0&0&0&0&1&1&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&1&1&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&1&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&1\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1\\
1&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&1&1&0&0&0&0
\end{pmatrix}$$

$$L_2^{-1}=\begin{pmatrix}
0&0&1&1&0&0&0&0&0&0&1&0&0&0&0&0\\
0&0&0&1&1&0&0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&1&1&0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&1&1&0&0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&0\\
1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1\\
1&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0\\
0&1&1&0&0&0&0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&1\\
1&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&1&1&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0&1&1&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0&0&1&1&0&0&0\\
0&0&0&0&1&0&0&0&0&0&0&0&1&1&0&0\\
0&0&0&0&0&1&0&0&0&0&0&0&0&1&1&0\\
0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&1
\end{pmatrix}$$

Permutation layers of P and P^{-1} are defined as follows:

Table 3.1: Permutation $P(x)$ of PRIDE

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$P(x)$	1	17	33	49	2	18	34	50	3	19	35	51	4	20	36	52
x	17	18	19	20	21	22	23	24	25	27	27	28	29	30	31	32
$P(x)$	5	21	37	53	6	22	38	54	7	23	39	55	8	24	40	56
x	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
$P(x)$	9	25	41	57	10	26	42	58	11	27	43	59	12	28	44	60
x	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
$P(x)$	13	29	45	61	14	30	46	62	15	31	47	63	16	32	48	64

Table 3.2: Permutation $P^{-1}(x)$ of PRIDE

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$P^{-1}(x)$	1	5	9	13	17	21	25	29	33	37	41	45	49	53	57	61
x	17	18	19	20	21	22	23	24	25	27	27	28	29	30	31	32
$P^{-1}(x)$	2	6	10	14	18	22	26	30	34	38	42	46	50	54	58	62
x	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
$P^{-1}(x)$	3	7	11	15	19	23	27	31	35	39	43	47	51	55	59	63
x	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
$P^{-1}(x)$	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64

3.1.3 Key Schedule

Key schedule of PRIDE [3] is defined as follows:

The 128-bit master key K of block cipher PRIDE is divided into two 64-bit parts ($k_0 || \mathrm{k}_1$). k_0 is used for pre-whitening and post-whitening, while k_1 is divided into 8 8-bit words

$$k_1 = k_{1,1} \,||\, \mathrm{k}_{1,2} \,||\, \mathrm{k}_{1,3} \,||\, \mathrm{k}_{1,4} \,||\, \mathrm{k}_{1,5} \,||\, \mathrm{k}_{1,6} \,||\, \mathrm{k}_{1,7} \,||\, \mathrm{k}_{1,8}$$

and used to generate the subkeys $f_r(k_1)$ which is defined as follows:

$$f_r(k_1) = k_{1,1} \,||\, \mathrm{g}_r^{(1)}(\mathrm{k}_{1,2}) \,||\, \mathrm{k}_{1,3} || \,\mathrm{g}_r^{(2)}(\mathrm{k}_{1,4}) \,||\, \mathrm{k}_{1,5} \,||\, g_r^{(3)}(\mathrm{k}_{1,6}) \,||\, \mathrm{k}_{1,7} \,||\, g_r^{(4)}(\mathrm{k}_{1,8})$$

as the subkey derivation function with four byte-local modifiers of the key as

$g_r^{(1)}(x) = (x + 193r) \bmod 256$

$g_r^{(2)}(x) = (x + 165r) \bmod 256$

$g_r^{(3)}(x) = (x + 81r) \bmod 256$

$g_r^{(4)}(x) = (x + 197r) \bmod 256$

which simply add one of four constants to every other byte of k_1.

3.1.4 Sbox

PRIDE's Sbox is shown in Table 3.3.

Table 3.3: Sbox of PRIDE [3]

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

It can be seen from the Table 3.3 that there are 4 fixed points in the PRIDE S-box. These fixed points are S[0x0]=0x0, S[0x5]=0x5, S[0xA]=0xA, S[0xD]=0xD.

3.1.5 Performance Analysis

Designers of PRIDE [3] presented the performance comparison of it with some of the other known block ciphers. Atmel's AVR micro-controller was the implementation platform for all of these block ciphers.

It can be seen that PRIDE [3] has better performance than many of the block ciphers in terms of cycle count and block size. Performance comparison of these block ciphers is shown in Table 3.4.

Table 3.4: Performance Analysis of PRIDE [3]

	AES-128	SERPENT-128	PRESENT-128	CLEFIA-128	SEA-96	NOEKEON-128	PRINCE-128	ITUbee-80	SIMON-64/128	SPECK-64/96	SPECK-64/128	PRIDE
t(cyc)	3159	49314	10792	28648	17745	23517	3614	2607	2000	1152	1200	**1514**
bytes	1570	7220	660	3046	386	364	1108	716	282	182	186	**266**

According to the Table 3.4 only SPECK-64/96 and SPECK-64/128 have better performance than PRIDE.

3.1.6 Testvectors for PRIDE [3]

Testvectors for PRIDE [3] is shown in Table 3.5.

Table 3.5: Testvectors for PRIDE

Plaintext	k_0	k_1	CipherText
0000000000000000	0000000000000000	0000000000000000	82b4109fcc70bd1f
ffffffffffffff	0000000000000000	0000000000000000	d70e60680a17b956
0000000000000000	ffffffffffffff	0000000000000000	28f19f97f5e846a9
0000000000000000	0000000000000000	ffffffffffffff	d123ebaf368fce62
0123456789abcdef	0000000000000000	fedcba9876543210	d1372929712d336e

CHAPTER 4

ATTACKS ON PRIDE

We have published some parts of this section in paper [70] and [71]. In this section, first we explain the 18-Round Attack [79], 19-Round Attack [76] and 20-Round Attack [21] applied on PRIDE, after that we provide corrections for each of these attacks considering differential factors.

4.1 Notation

The notation that is used in this section is presented in Table 4.1.

Table 4.1: PRIDE notation conventions

I_r	the input of the r-th round
X_r	the state after the key addition layer of the r-th round
Y_r	the state after the substitution layer of the r-th round
Z_r	the state after the permutation layer of the r-th round
W_r	the state after the matrix layer of the r-th round
O_r	the output of the r-th round
ΔX	the XOR difference of X and X'
?	a bit with an uncertain value
$X[n_1, n_2, \ldots]$	the n_1, n_2, ...-th nibbles of state X, $1 \leq n_1 < n_2 < \ldots \leq 16$
$X\{b_1, b_2, \ldots\}$	the b_1, b_2, ...-th bits of state X, $1 \leq b_1 < b_2 < \ldots \leq 64$, numbered from left to right.

4.2 Difference Distribution Table of PRIDE

Difference Distribution Table of PRIDE is shown is Table 4.2.

Table 4.2: Difference Distribution Table of PRIDE

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

4.3 Differential Factors of PRIDE

Table 4.3: Differential Factors of PRIDE

Sbox	**Differential Factors**
0, 4, 8, F, 1, 5, E, 9, 2, 7, A, C, B, D, 6, 3	$\lambda = 1$, $\mu = 1$
	$\lambda = 8$, $\mu = 8$

4.4 18 Round Differential Attack on PRIDE

In 18-Round differential attack [79], authors found 16 different 2-round iterative characteristics and they have constructed several 15-round differentials. And then, they have attacked 18 rounds of PRIDE with 2^{60} chosen plaintexts, 2^{66} encryptions and 2^{64} bytes.

4.4.1 Differential Characteristic of 18-Round Attack

In PRIDE Sbox, the input difference of 0x8, leads to the output difference of 0x8 with probability 2^{-2}. Authors have found 2-round iterative differential characteristic holding with probability 2^{-8} as in Table 4.4.

Table 4.4: 2-Round Differential Characteristic for PRIDE

ΔI_r	0x0	0x8	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
ΔX_r	0x0	0x8	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
ΔY_r	0x0	0x8	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
ΔZ_r	0x4	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
ΔW_r	0x0	0x4	0x4	0x4	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
ΔI_{r+1}	0x0	0x0	0x0	0x0	0x0	0x8	0x0	0x0	0x0	0x8	0x0	0x0	0x0	0x8	0x0	0x0
ΔX_{r+1}	0x0	0x0	0x0	0x0	0x0	0x8	0x0	0x0	0x0	0x8	0x0	0x0	0x0	0x8	0x0	0x0
ΔY_{r+1}	0x0	0x0	0x0	0x0	0x0	0x8	0x0	0x0	0x0	0x8	0x0	0x0	0x0	0x8	0x0	0x0
ΔZ_{r+1}	0x0	0x4	0x4	0x4	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
ΔW_{r+1}	0x4	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
ΔI_{r+2}	0x0	0x8	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0

Authors have found 16 2-round iterative differential characteristics listed in Table 4.5.

Table 4.5: 16 2-Round Differential Characteristics for PRIDE

$(8000000000000000) \rightarrow_{1r} (0000800080008000) \rightarrow_{1r} (8000000000000000)$
$(0800000000000000) \rightarrow_{1r} (0000080008000800) \rightarrow_{1r} (0800000000000000)$
$(0080000000000000) \rightarrow_{1r} (0000800000800080) \rightarrow_{1r} (0080000000000000)$
.
.
$(0000000000000800) \rightarrow_{1r} (0800080008000000) \rightarrow_{1r} (0000000000000800)$
$(0000000000000080) \rightarrow_{1r} (0080008000800000) \rightarrow_{1r} (8000000000000080)$
$(0000000000000008) \rightarrow_{1r} (0008000800080000) \rightarrow_{1r} (0000000000000008)$

Authors have iterated the 2-round differential 7 times and add one round below it to obtain 15-round differential with the probability of 2^{-58}. 15-round differential characteristic can be seen in Table 4.6.

Table 4.6: 15 Round Differential Characteristic for PRIDE

ΔI_r	0x0	0x8	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0
ΔX_{r+15}	0x0	0x0	0x0	0x0	0x0	0x8	0x0	0x0	0x0	0x8	0x0	0x0	0x0	0x8	0x0	0x0

Authors have added one round to the top and two rounds to the bottom of 15-round differential to attack 18-round of PRIDE block cipher as it is shown in Table 4.7.

Table 4.7: 18 Round Differential Attack of PRIDE

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
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

4.4.2 Data Collection Phase

In the data collection phase 2^n structures are chosen in that for each plaintext nibbles $1, 2, 3, 4, 5, 7, 8, 9, 11, 12, 13, 15, 16$ are fixed and nibbles $6, 10, 14$ are traversed. In other words, for each plaintext pairs nibbles $6, 10, 14$ are chosen different and the other nibbles are kept the same. For these nibbles there are 2^{23} plaintext and ciphertext pairs. For the chosen plaintexts at the end of 18-round, ciphertext difference should satisfy that the difference of nibbles $4, 8, 9, 12, 13, 16$ must be zero. Because, there are 2^4 possibility for each nibble, for the nibbles $4, 8, 9, 12, 13, 16$ the probability of having zero difference is 2^{-24}. This means that only 2^{-1} pairs left.

4.4.3 Key Recovery Phase

In this attack, 2^n structures are chosen and 64 bits of the key are guessed that are shown in Table 4.8. At the first round key bits of $X_1[6]$, $X_1[10]$, $X_1[14]$ and at the last round $Y_{18}[1]$, $Y_{18}[2]$, $Y_{18}[3]$, $Y_{18}[5]$, $Y_{18}[6]$, $Y_{18}[7]$, $Y_{18}[10]$, $Y_{18}[11]$, $Y_{18}[14]$, $Y_{18}[15]$ and at the 17th round $Y_{17}[6]$, $Y_{17}[10]$, $Y_{17}[14]$ are guessed.

- **Step1:** Encrypt the nibbles $X_1[6]$ and distinguish the pairs whose output difference $\Delta Y_1[6]$ is equal to 1000 and guess the key bits of $X_1[6]$. This means that 2^{-5} pairs remain.
- **Step2:** Encrypt the nibbles $X_1[10]$ and distinguish the pairs whose output difference $\Delta Y_1[10]$ is equal to 1000 and guess the key bits of $X_1[10]$. This means that 2^{-9} pairs remain.
- **Step3:** Encrypt the nibbles $X_1[14]$ and distinguish the pairs whose output difference $\Delta Y_1[14]$ is equal to 1000 and guess the key bits of $X_1[14]$. This means that 2^{-13} pairs remain.
- **Step4:** Decrypt the nibbles $Y_{18}[1]$, $Y_{18}[2]$, $Y_{18}[3]$, $Y_{18}[5]$, $Y_{18}[6]$, $Y_{18}[7]$, $Y_{18}[10]$, $Y_{18}[11]$, $Y_{18}[14]$, $Y_{18}[15]$ and guess the corresponding key bits by distinguishing pairs by factors 2^{-3}, 2^{-3}, 2^{-1}, 2^{-2}, 2^{-3}, 2^{-1}, 2^{-2}, 2^{0}, 2^{-1}, 2^{0}, 2^{-3} respectively. After this step 2^{-29} pairs remain.
- **Step5:** In this step other remaining pairs are decrypted without guessing key by distinguishing 2^{-12} pairs. This results in 2^{-41} pairs left.
- **Step6:** Decrypt the nibbles $Y_{17}[6]$ and distinguish the pairs whose output difference $\Delta X_{17}[6]$ is equal to 1000 and guess the key bits of $Y_{17}[6]$. 2^{-45} pairs remain after this step.
- **Step7:** Decrypt the nibbles $Y_{17}[10]$ and distinguish the pairs whose output difference $\Delta X_{17}[10]$ is equal to 1000 and guess the key bits of $Y_{17}[10]$. This results in 2^{-49} pairs remain.
- **Step8:** Decrypt the nibbles $Y_{17}[14]$ and distinguish the pairs whose output difference $\Delta X_{17}[14]$ is equal to 1000 and guess the key bits of $Y_{17}[14]$. 2^{-53} pairs remain after this step.

- **Step9:** Remaining 64 bits of the key are decrypted with the exhaustive search.

In this attack n is chosen to be 48, therefore 2^{48+23} pairs are used to guess the keys.

Table 4.8: Guessed key bits in 18 Round Differential Attack of PRIDE

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
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

4.4.4 Attack Complexity of 18-Round Attack

The time, data and memory complexity of 18-round attack are shown as follows:

- **Time Complexity:** 2^{66}
  - **Step1:** Encrypting nibbles $X_1[6]$ requires $2 \times 2^{47} \times 2^4 \times \frac{1}{16} \times \frac{1}{18} \approx 2^{43}$ encryptions.
  - **Step2:** This step is similar to Step1 and it requires $2 \times 2^{43} \times 2^4 \times 2^4 \times \frac{1}{16} \times \frac{1}{18} \approx 2^{43}$ encryptions.
  - **Step3:** This step is also similar to Step1 and it requires $2 \times 2^{39} \times 2^8 \times 2^4 \times \frac{1}{16} \times \frac{1}{18} \approx 2^{43}$ encryptions.
  - **Step4:** Decrypting the nibbles $Y_{18}[1]$, $Y_{18}[2]$, $Y_{18}[3]$, $Y_{18}[5]$, $Y_{18}[6]$, $Y_{18}[7]$, $Y_{18}[10]$, $Y_{18}[11]$, $Y_{18}[14]$, $Y_{18}[15]$ requires $2 \times 2^{19} \times 2^{48} \times 2^4 \times \frac{1}{16} \times \frac{1}{18} \approx 2^{62}$ encryptions.
  - **Step5:** Decrypting remaining pairs requires $2 \times 2^{19} \times 2^{52} \times \frac{1}{4} \times \frac{1}{18} \approx 2^{66}$ encryptions.
  - **Step6:** This step is also similar to Step1 and it requires $2 \times 2^7 \times 2^{52} \times 2^4 \times \frac{1}{16} \times \frac{1}{18} \approx 2^{55}$ encryptions.

- **Step7:** This step is also similar to Step1 and it requires $2 \times 2^3 \times 2^{56} \times 2^4 \times \frac{1}{16} \times \frac{1}{18} \approx 2^{55}$ encryptions.
- **Step8:** This step is also similar to Step1 and it requires $2 \times 2^{-1} \times 2^{60} \times 2^4 \times \frac{1}{16} \times \frac{1}{18} \approx 2^{55}$ encryptions.
- **Step9:** 2^{64} encryptions.

- **Data Complexity:** 2^{60}
- **Memory Complexity:** 2^{64}

4.4.5 Our Correction

In the provided attack [79] authors claim that they capture 64 bits of round keys with 2^{66} 18-round PRIDE encryptions. 40-bit round key is captured in the key addition layer of round 18, 12-bit round key is captured in the key addition layer of round 17 and 12-bit round key is captured in the key addition layer of the first round. And they also claim that, remaining 64-bit key is captured via exhaustive search with time complexity of 2^{64} encryptions.

However, we have corrected this attack in [70] as authors fail to discover differential factors that exist in the first and 17th round which are shown bold in Table 4.9. This shows that, it is not possible to capture the 6 bits of the key in the first part of the attack, actually, only 58 bits can be captured not 64 bits which require 2^{60} 18-round PRIDE encryptions. This also affects the exhaustive search part of the attack in that, the correct time complexity is 2^{70} 18-round PRIDE encryptions not 2^{66}.

Table 4.9: 18-round differential attack of [79]. Differences $\mu = 8$ which have differential factors $\lambda = 8$ are shown in bold.

	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_2																
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

4.5 19 Round Differential Attack on PRIDE

The 19-round attack [76] of PRIDE is based on a 15-round differential path that is obtained by iterating 1-round differential 15 times. Authors have used automatic search methods [60], [61] to find differential characteristics and they found 24 1-round iterative differential characteristics and 32 2-round iterative differential characteristics. So that, they have used one of the 1-round iterative differential characteristics to construct a 15-round differential path with the probability of 2^{-60}. Finally, they have attacked 19-round of PRIDE by adding two rounds to the bottom and two rounds to the top of 15-round differential path. The data, time and memory complexity of the attack is 2^{62}, 2^{63} and 2^{71} respectively.

4.5.1 Differential Characteristic of 19-Round Attack

Authors have found 24 different 1-round iterative differential characteristic which is shown in Table 4.10. They have used the 4th differential characteristic in Table 4.10 to obtain a 15-round differential characteristics with the probability of 2^{-60}.

Table 4.10: 1-Round Iterative Characteristics of PRIDE

1	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
2	1000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000
3	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000
4	**0000**	**0000**	**0000**	**0000**	**1000**	**0000**	**0000**	**0000**	**1000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**	**0000**
5	0000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000
6	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	1000	0000	0000	0000
7	0000	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
8	0000	1000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000
9	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000
10	0000	0000	0000	0000	0000	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000
11	0000	0000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000
12	0000	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	1000	0000	0000
13	0000	0000	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000
14	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	0000	0000
15	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000
16	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000
17	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	1000	0000
18	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	1000	0000
19	0000	0000	0000	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000
20	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	0000
21	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	1000
22	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	1000	0000	0000	0000	0000
23	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	1000
24	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	1000

Authors have attacked 19-round by adding two rounds to the top and two rounds to the bottom of a 15-round characteristics as it is shown in Table 4.11.

Table 4.11: 19-Round Attack on PRIDE

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
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

4.5.2 Data Collection Phase

In the data collection phase $2^{25,65}$ structures are chosen in that for each plaintext nibbles $4, 6, 8, 11, 12, 15, 16$ are fixed and nibbles $1, 2, 3, 5, 7, 9, 10, 13, 14$ are traversed. In other words, for each plaintext pairs nibbles $1, 2, 3, 5, 7, 9, 10, 13, 14$ are chosen different and the other nibbles are kept the same. For these nibbles there are 2^{71} plaintext and ciphertext pairs. So that, the total number of pairs can be calculated as $2^{71+25,65}$.

For the chosen plaintexts at the end of 19-round, ciphertext difference should satisfy that the difference of nibbles $3, 4, 6, 7, 11, 12, 14, 15, 16$ must be zero. There are 109 situations that satisfy this condition and thus, only $2^{60,65}$ pairs remain.

4.5.3 Key Recovery Phase

In this attack, 68 bits of the key are guessed as they are shown in Table 4.13. At the first round key bits of $X_1[1]$, $X_1[2]$, $X_1[3]$, $X_1[5]$, $X_1[7]$, $X_1[9]$, $X_1[10]$, $X_1[13]$, $X_1[14]$ and at the last round $Y_{19}[1]$, $Y_{19}[2]$, $Y_{19}[5]$, $Y_{19}[8]$, $Y_{19}[9]$, $Y_{19}[10]$, $Y_{19}[13]$ and at the 18th round $Y_{18}[5]$, $Y_{18}[9]$ and at the 2nd round $X_2[5]$, $X_2[9]$ are guessed.

Authors have chosen one of the 109 situations presented below to determine the right number of pairs to attack. Because, 13th nibble of ΔX_{19} is chosen to be 0000, $2^{60,65} \times 2^{-4} = 2^{56,65}$ pairs remain.

Table 4.12: Chosen pair of 19-Round Attack

ΔY_1	1000	0010	0010	0000	0001	0000	0010	0000	1011	0010	0000	0000	1001	0010	0000	0000
ΔX_{19}	0010	0010	0000	0000	1000	0000	0000	0010	1000	0010	0000	0000	0000	0000	0000	0000

By investigating the Difference Distribution Table of PRIDE on Table 4.2, authors sieved the corresponding pairs in Table 4.12 with the proability of 4/16, 6/16, 6/16, 4/16, 6/16, 6/16, 6/16, 8/16, 6/16 for ΔY_1 and 6/16, 6/16, 4/16, 6/16, 4/16, 6/16 for ΔX_{19}. Therefore, $2^{56,65} \times (4/16)^4 \times (6/16)^{10} \times 8/16 \approx 2^{33.50}$ pairs remain.

Table 4.13: Guessed Key Bits in 19-Round Attack on PRIDE

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																
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

4.5.4 Attack Complexity of 19-Round Attack

The time, data and memory complexity of 19-round attack are shown as follows:

- **Time Complexity:** 2^{63}
  - **Step1:** Considering the situation in Table 4.12, encrypt the nibbles $X_1[1]$ and distinguish the pairs whose output difference $\Delta Y_1[1]$ is equal to 1000 and store the values in a Table. The time complexity for this step is $2^{33,50} \times 1/16 \times 1/19 \approx 2^{25,25}$
  - **Step2:** Considering again the same situation in Table 4.12, encrypt the nibbles $X_1[13]$ and distinguish the pairs whose output difference $\Delta Y_1[13]$ is equal to 1001 and store the values in the Table. The time complexity for this step is $2^{33,50} \times 4 \times 1/16 \times 1/19 \approx 2^{27,25}$
  - **Step3:** Considering again the same situation in Table 4.12, by investigating the Difference Distribution Table of PRIDE 4.2 distinguish the candidate pairs for $\Delta X_1[2]$, $\Delta X_1[3]$, $\Delta X_1[5]$, $\Delta X_1[7]$, $\Delta X_1[9]$, $\Delta X_1[10]$,$\Delta X_1[14]$ and $\Delta Y_{19}[1]$, $\Delta Y_{19}[2]$, $\Delta Y_{19}[5]$, $\Delta Y_{19}[8]$, $\Delta Y_{19}[9]$, $\Delta Y_{19}[10]$. These pairs again are stored in the same table. The time complexity for this step is $2^{54,65} \times 1/16 \times 1/19 \approx 2^{46,40}$
  - **Step4:** For ΔX_2 and ΔY_{18} all the pairs determined in the previous steps looked up four times.
  - **Step5:** Previous steps are repeated for 109 other situations and candidate table contains $2^{56,65} \times 2^8 \times 3 \times 27 \approx 2^{70,99}$ pairs for 68 bits of the key. The most frequently appearing pair in the table is the right key and the time complexity or this step is $2^{56,65} \times 2^8 \times 3 \times 27 \times 1/16 \times 1/19 \approx 2^{62,74}$
  - **Step6:** For the rest 60 bits of the key are guessed with the exhaustive search with 2^{60} encryptions.
- **Data Complexity:** 2^{62}
- **Memory Complexity:** 2^{71}

4.5.5 Our Correction

19-round attack captures 68 bits of the round keys with time complexity of 2^{63} encryptions and remaining 60 bits are captured via exhaustive search.

However, we have corrected this attack in [70] as authors fail to discover the differential factors existing in the 2nd and 18th round that can be seen in Table 4.14. This shows that 4 bits of the round keys are not actually captured. Therefore, only 64 bits of the keys are captured which requires 2^{59} encryptions and remaining 64 bits are found via exhaustive search. Thus, time complexity is 2^{64} not 2^{63}.

Table 4.14: 19-round differential attack of [76]. Differences $\mu = 8$ which have differential factors $\lambda = 8$ are shown in bold.

	Differences in bits															
Rounds	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
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																
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

Besides, we have noticed that authors have used undisturbed bits of PRIDE Sbox as input difference **1000** yields to output difference **?0??** which can be seen in Table 4.14 $\Delta X_2[5]$, $\Delta X_2[9]$, $\Delta Y_{18}[5]$ and $\Delta Y_{18}[9]$. By this property, this attack has been improved to cover 19-round. However, key bits corresponding to undisturbed bits with 0 difference may also need to be captured. This can further increase the time complexity value. Although, we have mentioned the effects of differential factors in this correction, final time complexity of this attack may need further correction because of the need for guessing these key bits corresponding to undisturbed bits.

4.6 20 Round Differential Attack on PRIDE

In 20-Round differential attack [21], authors have used 18-round and 17-round related-key differential characteristics to attack 20-round of PRIDE. They have also improved 18-round attack by using multiple related-key differentials. The details and the complexity values of these attacks are provided in Table 4.15.

Table 4.15: 20-round attack details of PRIDE

Differential Characteristics	Attacked Rounds	Data Complexity	Time Complexity
18-Round Related-key	20	2^{39}	2^{60}
18-Round Multiple Related-key	20	$2^{41,4}$	2^{44}
17-Round Related-key	20	2^{34}	$2^{53,7}$

4.6.1 Related-Key Differential Characteristic of 20-Round Attack

Authors provided 17-round and 18-round related-key differential characteristics by using 2-round iterative related-key differential characteristics. They have found 8 different 2-round iterative related-key differential characteristic and each of them can be used to construct 17-round or 18-round differential path to attack full PRIDE. One of the 2-round iterative related-key differential characteristics that can be observed when $\Delta k_1 = 8800000000000000$ is provided in Table 4.16 and it has the probability of 2^{-4}.

$$(8000800080000000) \rightarrow_{1r} (8000800000008000) \rightarrow_{1r} (8000800080000000)$$

Table 4.16: 2-round iterative related-key differential characteristics

ΔI_r	1000	0000	0000	0000	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000
ΔX_r	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000
ΔY_r	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000
ΔZ_r	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔW_r	1000	1000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔI_{r+1}	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000
ΔX_{r+1}	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000
ΔY_{r+1}	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	1000	0000	0000	0000
ΔZ_{r+1}	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔW_{r+1}	1000	1000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
ΔI_{r+2}	1000	0000	0000	0000	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000

For 4 different Δk_1 values authors have found 8 different 2-round iterative related key differential characteristics which are shown in Table 4.17.

Table 4.17: 8 2-round iterative characteristics

2-round characteristics	$\Delta P^{-1}(\Delta f_r(k_1))$	$\Delta f_r(k_1)$
8000800080000000 $\rightarrow_{2r}$ 8000800080000000	8000 8000 0000 0000	8800 0000 0000 0000
0800080008000000 $\rightarrow_{2r}$ 0800080008000000	0800 0800 0000 0000	4400 0000 0000 0000
0080008000800000 $\rightarrow_{2r}$ 0080008000800000	0080 0080 0000 0000	2200 0000 0000 0000
0008000800080000 $\rightarrow_{2r}$ 0008000800080000	0008 0008 0000 0000	1100 0000 0000 0000
8000800000000000 $\rightarrow_{2r}$ 8000800000000000	8000 8000 0000 0000	8800 0000 0000 0000
0800080000000000 $\rightarrow_{2r}$ 0800080000000000	0800 0800 0000 0000	4400 0000 0000 0000
0080008000000000 $\rightarrow_{2r}$ 0080008000000000	0080 0080 0000 0000	2200 0000 0000 0000
0008000800000000 $\rightarrow_{2r}$ 0008000800000000	0008 0008 0000 0000	1100 0000 0000 0000

4.6.2 Key Recovery Attack By Using 18-Round Path

Authors have used the following 2-round iterative characteristic to obtain 18-round differential characteristic with propability of 2^{-36}.

$$(8000800080000000 \rightarrow_{2r} 8000800080000000)$$

Authors have added 2-round after the 18-round characteristics to attack full PRIDE with 2^{39} chosen plaintexts and 2^{60} encryptions. Full PRIDE attack is shown in Table 4.18.

Table 4.18: Full PRIDE attack Based on 18-Round Differential

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

This attack requires 2^{39} chosen plaintexts and 2^{60} encryptions and the detailed attack steps are explained below:

- **Step1:** Decrypt the nibbles Y_{20}[1,2,5,6,9,10,13,14] and guess the the related key bits partially by comparing the results if the difference of the decrypted nibbles of ΔX_{20}[1,2,5,6,9,10,13,14] equals to *00*, 00*0, *00*, 00*0, *00*, 00*0, **0*, 0*00 and the probability is 2^{-2}, 2^{-3}, 2^{-2}, 2^{-3}, 2^{-3}, 2^{-3}, 2^{-1}, and 2^{-3} respectively. This step requires $2 \times 2^6 \times 2^{32} \times 1/20 = 2^{35}$ encryptions.

- **Step2:** Exhaustively guess key bits of Y_{20}[3,4,7,8,11,12,15,16]. And guess Y_{19}[9] by checking if ΔX_{19}[9] is 1000. This step requires $2 \times 2^{32} \times 2^{-24} \times 2^{36} \times 1/20 = 2^{41}$ encryptions.

- **Step3:** Exhaustively guess the rest 60-bit keys with 2^{60} encryptions.

  Therefore this attacks requires 2^{60} encryptions.

This attack is improved by using multiple characteristics. This time, attack requires $2^{41,4}$ chosen plaintexts and 2^{44} encryptions.

4.6.3 Key-Recovery Attack By Using 17-Round Path

In this attack, authors have used another 2-round iterative differential characteristic (8000800000000000 $\rightarrow_{2r}$ 8000800000000000) to obtain 17-round differential characteristic with probability of 2^{-32} with Δk_1 = 880000000000000. 17-round differential characteristic is shown as follows:

$$8000800000000000 \rightarrow_{16r} 8000800000000000 \rightarrow_{1r} 0000000000000000$$

Authors have added 1-round before the characteristics and 2-round after the characteristics to attack the full PRIDE. This attack is shown in Table 4.19.

Table 4.19: Full PRIDE attack based on 17-round differential

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
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

This attack requires 2^{34} chosen plaintexts and $2^{53,7}$ encryptions and the detailed attack steps are shown below:

- **Step1:** Guess 8-bit keys by encrypting X_1[1,5] and checking results if ΔY_1[1,5] equal to 1000. This step requires $2 \times 2 \times 28 \times 1/20 = 2^{5,7}$ encryptions.
- **Step2:** Guess the key bits of Y_{20}[1, 4, 5, 8, 9, 13] and checking results if ΔX_{20}[1, 4, 5, 8, 9, 13] equal to ****, 0**0, ****, 0**0, ****, ****. The probability is 1, 2^{-2}, 1, 2^{-2}, 1, and 1 respectively. This step requires $2^8 \times 2 \times 2^{-7} \times 2^{24} \times 1/20$ = $2^{21,7}$ encryptions.
- **Step3:** Guess 40 bits of the key corresponding to Y_{20}[2, 3, 6, 7, 10, 11, 12, 14, 15, 16] and partially decrypt them. Guess 8-bit keys Y_{19}[1, 5] by decrypting and checking results if ΔX_{19}[1, 5] equal to 1000. This step requires $2^{32} \times 2 \times 2^{-23} \times 2^{48} \times 1/20 = 2^{53,7}$ encryptions.
- **Step4:** Other 48-bit keys are guessed by exhaustive search and this step needs 2^{48} encryptions.

  Therefore this attacks requires $2^{53,7}$ encryptions.

4.6.4 Our Correction

In Section 4.6.2, authors claim to capture 68 bits of the key by using an 18-round path with 2^{41} encryptions and perform 2^{60} encryptions to capture the remaining bits. However, we have corrected this attack in [70] as authors fail to discover existing single differential factor shown in Table 4.20. As a result, actual time complexity of this attack is 2^{61} not 2^{60}.

Table 4.20: Full PRIDE attack Based on 18-Round Differential. Differences $\mu = 8$ which have differential factors $\lambda = 8$ are shown in bold.

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

In the second attack presented in Section 4.6.3, one round is added to the top and two rounds are added to the bottom of the 17-round related-key differential characteristics that are also based on the 2-round iterative characteristics Δ_5 with probability of 2^{-32} where

$$\Delta_5: 8880000000000000 \rightarrow 8000800080000000 \rightarrow 0000000000000000$$

Authors claim to capture 80 bits of the key by using a 17-round path with $2^{53.7}$ encryptions and perform 2^{48} encryptions to capture the remaining bits. However, we have corrected this attack also in [70] as authors fail to discover four differential factors which are shown in Table 4.21. This shows that 4 bits of the round keys are not actually captured. Therefore, only 76 bits of the keys are captured which requires $2^{49.7}$ encryptions and remaining 52 bits are found via exhaustive search. Therefore, time complexity for this attack is $2^{52} + 2^{49,7}$ encryptions not $2^{53,7} + 2^{48}$. As a result, by this correction we have shown that this attack is approximately twice faster then it is claimed.

Moreover, since PRIDE [3] is not designed to resist the related key attacks, these 20-round related-key attacks do not contradict the security claims of the designers.

Table 4.21: Full PRIDE attack Based on 17-Round Differential. Differences $\mu = 8$ which have differential factors $\lambda = 8$ are shown in bold.

	Differences in bits															
Rounds	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
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_1	1000	0000	0000	0000	1000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
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

CHAPTER 5

CONCLUSION

5.1 Conclusion

In the last few decades, security of computing environment was mainly provided by Cryptographic applications like block ciphers. Cryptography is implemented on almost every security intensive application. Besides, in recent years with the emergence of IoT technologies, several lightweight block ciphers that are optimized for limited-source devices are published. However, in order to be sure about the security of these lightweight block ciphers, further studies should be done. In fact, if a block cipher does not have a security vulnerability, capturing the secret key is the only way to access the encrypted information. Although, Exhaustive Search or Table Attacks are the easiest attack methods that cryptanalysts can use to capture the secret key, block ciphers are designed today to have enough key length that can not be guessed with the today's computing capabilities. Therefore, different cryptanalysis techniques have evolved over time.

Differential cryptanalysis is one of the most used cryptanalysis technique today, to analyze the block ciphers. It is based on the relations of input differences and the output differences of block ciphers. In order to have a differential attack, cryptanalysts first try to discover a differential characteristic for a block cipher. If, after r rounds, ΔI input difference causes ΔO output difference with probability higher than it is for a random permutation, it is called as differential characteristic. In a differential attack, some rounds are added to before or after of differential characteristic. After that, every possible activated round key bits are tried on plaintext pairs expecting to satisfy the

differential characteristic. It is expected that correct key must satisfy the differential characteristic more times than any other key, which makes possible to capture some parts of the secret key.

Differential Factors that is a recent study discovered by Tezcan showed that it may not be possible to fully capture the attacked round key bits when performing a differential attack. If there exists a differential factor in a block cipher, for any plaintext/ciphertext pair when a key k is tried and S-box output difference μ is obtained from S-box substitution, $k \oplus \lambda$ also produces the same result which is μ again. This situation can also be observed from the counter values of candidate keys in that counter value of k and counter value of $k \oplus \lambda$ become equal. This means that, the correct candidate key containing k is indistinguishable from an incorrect candidate key containing $k \oplus \lambda$. As a result, during key guess step it may not be possible to discover the key bits where differential factors exist. In this case, the advantage of the cryptanalyst is reduced by 1 bit and the time complexity of this key guess step is halved. Therefore, these key bits must be discovered with the exhaustive search.

In this thesis, we have explained the concepts of Differential Cryptanalysis, Differential Factors and Undisturbed Bits. Furthermore, We have also investigated several lightweight block ciphers for the existence of differential factors and presented differential factors and undisturbed bits of these block ciphers. In our research we have also seen that, most of the differential attacks towards block ciphers did not consider differential factors which directly affects the time complexity of these attacks. After that, by following these findings, we have investigated PRIDE block cipher which has very good performance and studied differential attacks towards it in detail. We have investigated 18-round, 19-round and 20-round differential attacks of PRIDE and presented our findings in this study. In the mean time, we have noticed that in 18-round, 19-round and 20-round attacks, authors did not take into account differential factors that exist in PRIDE block cipher. As a result, we have provided some corrections for these attacks by considering differential factors. Finally, we have presented the correct time complexity values for the 18-round, 19-round and 20-round attack of PRIDE which are 2^{70}, 2^{64} and 2^{48} instead of 2^{66}, 2^{63} and 2^{52} as authors claimed.

We have also published some parts of our research in papers [70] and [71].

Bibliography

[1] Carlisle M. Adams. “Constructing Symmetric Ciphers Using the CAST Design Procedure”. In: *Des. Codes Cryptography* 12.3 (1997), pp. 283–316. DOI: 10.1023/A:1008229029587. URL: http://dx.doi.org/10.1023/A:1008229029587.

[2] Sufyan Salim Mahmood AlDabbagh, Imad Fakhri Taha Al Shaikhli, and Mohammad A. Alahmad. “HISEC: A New Lightweight Block Cipher Algorithm”. In: *Proceedings of the 7th International Conference on Security of Information and Networks, Glasgow, Scotland, UK, September 9-11, 2014*. 2014, p. 151. DOI: 10.1145/2659651.2659662. URL: http://doi.acm.org/10.1145/2659651.2659662.

[3] Martin R. Albrecht et al. “Block Ciphers - Focus on the Linear Layer (feat. PRIDE)”. In: *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I*. 2014, pp. 57–76. DOI: 10.1007/978-3-662-44371-2_4. URL: http://dx.doi.org/10.1007/978-3-662-44371-2_4.

[4] Kiyomichi Araki, Takakazu Satoh, and Shinji Miura. “Overview of Elliptic Curve Cryptography”. In: *Public Key Cryptography, First International Workshop on Practice and Theory in Public Key Cryptography, PKC ’98, Pacifico Yokohama, Japan, February 5-6, 1998, Proceedings*. 1998, pp. 29–49. DOI: 10.1007/BFb0054012. URL: http://dx.doi.org/10.1007/BFb0054012.

[5] Steve Babbage and Laurent Frisch. “On MISTY1 Higher Order Differential Cryptanalysis”. In: *Information Security and Cryptology - ICISC 2000, Third International Conference, Seoul, Korea, December 8-9, 2000, Proceedings*. 2000, pp. 22–36. DOI: 10.1007/3-540-45247-8_3. URL: http://dx.doi.org/10.1007/3-540-45247-8_3.

[6] Subhadeep Banik et al. “Midori: A Block Cipher for Low Energy (Extended Version)”. In: *IACR Cryptology ePrint Archive* 2015 (2015), p. 1142. URL: http://eprint.iacr.org/2015/1142.

[7] Adnan Baysal and Sühap Sahin. “RoadRunneR: A Small and Fast Bitslice Block Cipher for Low Cost 8-Bit Processors”. In: *Lightweight Cryptography for Security and Privacy - 4th International Workshop, LightSec 2015, Bochum, Germany, September 10-11, 2015, Revised Selected Papers*. 2015, pp. 58–76. DOI: 10.1007/978-3-319-29078-2_4. URL: http://dx.doi.org/10.1007/978-3-319-29078-2_4.

[8] Ray Beaulieu et al. “The SIMON and SPECK Families of Lightweight Block Ciphers”. In: *IACR Cryptology ePrint Archive* 2013 (2013), p. 404. URL: http://eprint.iacr.org/2013/404.

[9] Christof Beierle et al. “The SKINNY Family of Block Ciphers and its Low-Latency Variant MANTIS”. In: *IACR Cryptology ePrint Archive* 2016 (2016), p. 660. URL: http://eprint.iacr.org/2016/660.

[10] Eli Biham, Ross J. Anderson, and Lars R. Knudsen. “Serpent: A New Block Cipher Proposal”. In: *Fast Software Encryption, 5th International Workshop, FSE ’98, Paris, France, March 23-25, 1998, Proceedings*. 1998, pp. 222–238. DOI: 10.1007/3-540-69710-1_15. URL: http://dx.doi.org/10.1007/3-540-69710-1_15.

[11] Eli Biham, Alex Biryukov, and Adi Shamir. “Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials”. In: *Advances in Cryptology - EUROCRYPT ’99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*. 1999, pp. 12–23. DOI: 10.1007/3-540-48910-X_2. URL: http://dx.doi.org/10.1007/3-540-48910-X_2.

[12] Eli Biham, Orr Dunkelman, and Nathan Keller. “Differential-Linear Cryptanalysis of Serpent”. In: *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*. 2003, pp. 9–

21. DOI: 10.1007/978-3-540-39887-5_2. URL: http://dx.doi.org/10.1007/978-3-540-39887-5_2.

[13] Eli Biham and Adi Shamir. "Differential Cryptanalysis of DES-like Cryptosystems". In: *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*. 1990, pp. 2–21. DOI: 10.1007/3-540-38424-3_1. URL: http://dx.doi.org/10.1007/3-540-38424-3_1.

[14] Andrey Bogdanov et al. "PRESENT: An Ultra-Lightweight Block Cipher". In: *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*. 2007, pp. 450–466. DOI: 10.1007/978-3-540-74735-2_31. URL: http://dx.doi.org/10.1007/978-3-540-74735-2_31.

[15] Andrey Bogdanov et al. "spongent: A Lightweight Hash Function". In: *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*. 2011, pp. 312–325. DOI: 10.1007/978-3-642-23951-9_21. URL: http://dx.doi.org/10.1007/978-3-642-23951-9_21.

[16] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. "On the Importance of Checking Cryptographic Protocols for Faults (Extended Abstract)". In: *Advances in Cryptology - EUROCRYPT '97, International Conference on the Theory and Application of Cryptographic Techniques, Konstanz, Germany, May 11-15, 1997, Proceeding*. 1997, pp. 37–51. DOI: 10.1007/3-540-69053-0_4. URL: http://dx.doi.org/10.1007/3-540-69053-0_4.

[17] Julia Borghoff et al. "PRINCE - A Low-latency Block Cipher for Pervasive Computing Applications (Full version)". In: *IACR Cryptology ePrint Archive* 2012 (2012), p. 529. URL: http://eprint.iacr.org/2012/529.

[18] Christina Boura, María Naya-Plasencia, and Valentin Suder. "Scrutinizing and Improving Impossible Differential Attacks: Applications to CLEFIA, Camellia, LBlock and Simon (Full Version)". In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 699. URL: http://eprint.iacr.org/2014/699.

[19] Christophe De Cannière, Orr Dunkelman, and Miroslav Knezevic. “KATAN and KTANTAN - A Family of Small and Efficient Hardware-Oriented Block Ciphers”. In: *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*. 2009, pp. 272–288. DOI: 10.1007/978-3-642-04138-9_20. URL: http://dx.doi.org/10.1007/978-3-642-04138-9_20.

[20] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002. ISBN: 3-540-42580-2. DOI: 10.1007/978-3-662-04722-4. URL: http://dx.doi.org/10.1007/978-3-662-04722-4.

[21] Yibin Dai and Shaozhen Chen. “Cryptanalysis of Full PRIDE Block Cipher”. In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 987. URL: http://eprint.iacr.org/2014/987.

[22] Sourav Das. “Halka: A Lightweight, Software Friendly Block Cipher Using Ultra-lightweight 8-bit S-box”. In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 110. URL: http://eprint.iacr.org/2014/110.

[23] Whitfield Diffie and Martin E. Hellman. “New directions in cryptography”. In: *IEEE Trans. Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638. URL: http://dx.doi.org/10.1109/TIT.1976.1055638.

[24] Daniel Dinu et al. “Design Strategies for ARX with Provable Bounds: SPARX and LAX (Full Version)”. In: *IACR Cryptology ePrint Archive* 2016 (2016), p. 984. URL: http://eprint.iacr.org/2016/984.

[25] Daniel Dinu et al. “Triathlon of Lightweight Block Ciphers for the Internet of Things”. In: *IACR Cryptology ePrint Archive* 2015 (2015), p. 209. URL: http://eprint.iacr.org/2015/209.

[26] Itai Dinur. “Cryptanalytic Time-Memory-Data Tradeoffs for FX-Constructions with Applications to PRINCE and PRIDE”. In: *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015,*

Proceedings, Part I. 2015, pp. 231–253. DOI: 10.1007/978-3-662-46800-5_10. URL: http://dx.doi.org/10.1007/978-3-662-46800-5_10.

[27] Orr Dunkelman, Sebastiaan Indesteege, and Nathan Keller. "A Differential-Linear Attack on 12-Round Serpent". In: *Progress in Cryptology - INDOCRYPT 2008, 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008. Proceedings*. 2008, pp. 308–321. DOI: 10.1007/978-3-540-89754-5_24. URL: http://dx.doi.org/10.1007/978-3-540-89754-5_24.

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

[29] Thomas Eisenbarth et al. "A Survey of Lightweight-Cryptography Implementations". In: *IEEE Design & Test of Computers* 24.6 (2007), pp. 522–533. DOI: 10.1109/MDT.2007.178. URL: http://dx.doi.org/10.1109/MDT.2007.178.

[30] Pierre-Alain Fouque, Antoine Joux, and Chrysanthi Mavromati. "Multi-user Collisions: Applications to Discrete Logarithm, Even-Mansour and PRINCE". In: *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*. 2014, pp. 420–438. DOI: 10.1007/978-3-662-45611-8_22. URL: http://dx.doi.org/10.1007/978-3-662-45611-8_22.

[31] Taher El Gamal. "A public key cryptosystem and a signature scheme based on discrete logarithms". In: *IEEE Trans. Information Theory* 31.4 (1985), pp. 469–472. DOI: 10.1109/TIT.1985.1057074. URL: http://dx.doi.org/10.1109/TIT.1985.1057074.

[32] Benoît Gérard et al. "Block Ciphers that are Easier to Mask: How Far Can we Go?" In: *IACR Cryptology ePrint Archive* 2013 (2013), p. 369. URL: http://eprint.iacr.org/2013/369.

[33] Zheng Gong, Svetla Nikova, and Yee Wei Law. “KLEIN: A New Family of Lightweight Block Ciphers”. In: *RFID. Security and Privacy - 7th International Workshop, RFIDSec 2011, Amherst, USA, June 26-28, 2011, Revised Selected Papers*. 2011, pp. 1–18. DOI: 10.1007/978-3-642-25286-0_1. URL: http://dx.doi.org/10.1007/978-3-642-25286-0_1.

[34] Vincent Grosso et al. “LS-Designs: Bitslice Encryption for Efficient Masked Software Implementations”. In: *Fast Software Encryption - 21st International Workshop, FSE 2014, London, UK, March 3-5, 2014. Revised Selected Papers*. 2014, pp. 18–37. DOI: 10.1007/978-3-662-46706-0_2. URL: http://dx.doi.org/10.1007/978-3-662-46706-0_2.

[35] Jian Guo et al. “The LED Block Cipher”. In: *IACR Cryptology ePrint Archive* 2012 (2012), p. 600. URL: http://eprint.iacr.org/2012/600.

[36] Martin E. Hellman. “A cryptanalytic time-memory trade-off”. In: *IEEE Trans. Information Theory* 26.4 (1980), pp. 401–406. DOI: 10.1109/TIT.1980.1056220. URL: http://dx.doi.org/10.1109/TIT.1980.1056220.

[37] Deukjo Hong et al. “LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors”. In: *Information Security Applications - 14th International Workshop, WISA 2013, Jeju Island, Korea, August 19-21, 2013, Revised Selected Papers*. 2013, pp. 3–27. DOI: 10.1007/978-3-319-05149-9_1. URL: http://dx.doi.org/10.1007/978-3-319-05149-9_1.

[38] Anthony Journault, François-Xavier Standaert, and Kerem Varici. “Improving the security and efficiency of block ciphers based on LS-designs”. In: *Des. Codes Cryptography* 82.1-2 (2017), pp. 495–509. DOI: 10.1007/s10623-016-0193-8. URL: http://dx.doi.org/10.1007/s10623-016-0193-8.

[39] Pascal Junod and Serge Vaudenay. “FOX : A New Family of Block Ciphers”. In: *Selected Areas in Cryptography, 11th International Workshop, SAC 2004, Waterloo, Canada, August 9-10, 2004, Revised Selected Papers*. 2004, pp. 114–129. DOI: 10.1007/978-3-540-30564-4_8. URL: http://dx.doi.org/10.1007/978-3-540-30564-4_8.

[40] Ferhat Karakoç, Hüseyin Demirci, and A. Emre Harmanci. “ITUbee: A Software Oriented Lightweight Block Cipher”. In: *Lightweight Cryptography for Security and Privacy - Second International Workshop, LightSec 2013, Gebze, Turkey, May 6-7, 2013, Revised Selected Papers*. 2013, pp. 16–27. DOI: 10.1007/978-3-642-40392-7_2. URL: http://dx.doi.org/10.1007/978-3-642-40392-7_2.

[41] Joe Kilian and Phillip Rogaway. “How to Protect DES Against Exhaustive Key Search”. In: *Advances in Cryptology - CRYPTO ’96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*. 1996, pp. 252–267. DOI: 10.1007/3-540-68697-5_20. URL: http://dx.doi.org/10.1007/3-540-68697-5_20.

[42] Jongsung Kim et al. “Truncated Differential Attacks on 8-Round CRYPTON”. In: *Information Security and Cryptology - ICISC 2003, 6th International Conference, Seoul, Korea, November 27-28, 2003, Revised Papers*. 2003, pp. 446–456. DOI: 10.1007/978-3-540-24691-6_33. URL: http://dx.doi.org/10.1007/978-3-540-24691-6_33.

[43] Özgül Küçük. *The Hash Function Hamsi*. Submission to NIST. 2008. URL: http://ehash.iaik.tugraz.at/uploads/9/95/Hamsi.pdf.

[44] Lars R. Knudsen. “Truncated and Higher Order Differentials”. In: *Fast Software Encryption: Second International Workshop. Leuven, Belgium, 14-16 December 1994, Proceedings*. 1994, pp. 196–211. DOI: 10.1007/3-540-60590-8_16. URL: http://dx.doi.org/10.1007/3-540-60590-8_16.

[45] François Koeune and François-Xavier Standaert. “A Tutorial on Physical Security and Side-Channel Attacks”. In: *Foundations of Security Analysis and Design III, FOSAD 2004/2005 Tutorial Lectures*. 2004, pp. 78–108. DOI: 10.1007/11554578_3. URL: http://dx.doi.org/10.1007/11554578_3.

[46] Manoj Kumar, Saibal K. Pal, and Anupama Panigrahi. “FeW: A Lightweight Block Cipher”. In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 326. URL: http://eprint.iacr.org/2014/326.

[47] Benjamin Lac et al. “A First DFA on PRIDE: from Theory to Practice (extended version)”. In: *IACR Cryptology ePrint Archive* 2017 (2017), p. 75. URL: http://eprint.iacr.org/2017/075.

[48] Mario Lamberger and Florian Mendel. “Higher-Order Differential Attack on Reduced SHA-256”. In: *IACR Cryptology ePrint Archive* 2011 (2011), p. 37. URL: http://eprint.iacr.org/2011/037.

[49] Jiqiang Lu et al. “New Impossible Differential Attacks on AES”. In: *IACR Cryptology ePrint Archive* 2008 (2008), p. 540. URL: http://eprint.iacr.org/2008/540.

[50] Charalampos Manifavas et al. “A survey of lightweight stream ciphers for embedded systems”. In: *Security and Communication Networks* 9.10 (2016), pp. 1226–1246. DOI: 10.1002/sec.1399. URL: http://dx.doi.org/10.1002/sec.1399.

[51] Charalampos Manifavas et al. “Lightweight Cryptography for Embedded Systems - A Comparative Analysis”. In: *Data Privacy Management and Autonomous Spontaneous Security - 8th International Workshop, DPM 2013, and 6th International Workshop, SETOP 2013, Egham, UK, September 12-13, 2013, Revised Selected Papers*. 2013, pp. 333–349. DOI: 10.1007/978-3-642-54568-9_21. URL: http://dx.doi.org/10.1007/978-3-642-54568-9_21.

[52] Shiho Moriai, Takeshi Shimoyama, and Toshinobu Kaneko. “Higher Order Differential Attak of CAST Cipher”. In: *Fast Software Encryption, 5th International Workshop, FSE ’98, Paris, France, March 23-25, 1998, Proceedings*. 1998, pp. 17–31. DOI: 10.1007/3-540-69710-1_2. URL: http://dx.doi.org/10.1007/3-540-69710-1_2.

[53] Shrikant Ojha et al. “TWIS - A Lightweight Block Cipher”. In: *Information Systems Security, 5th International Conference, ICISS 2009, Kolkata, India, December 14-18, 2009, Proceedings*. 2009, pp. 280–291. DOI: 10.1007/978-3-642-10772-6_21. URL: http://dx.doi.org/10.1007/978-3-642-10772-6_21.

[54] Gilles Piret, Thomas Roche, and Claude Carlet. “PICARO - A Block Cipher Allowing Efficient Higher-Order Side-Channel Resistance - Extended Version -”. In: *IACR Cryptology ePrint Archive* 2012 (2012), p. 358. URL: http://eprint.iacr.org/2012/358.

[55] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”. In: *Commun. ACM* 21.2 (1978), pp. 120–126. DOI: 10.1145/359340.359342. URL: http://doi.acm.org/10.1145/359340.359342.

[56] Bruce Schneier. “Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish)”. In: *Fast Software Encryption, Cambridge Security Workshop, Cambridge, UK, December 9-11, 1993, Proceedings*. 1993, pp. 191–204. DOI: 10.1007/3-540-58108-1_24. URL: http://dx.doi.org/10.1007/3-540-58108-1_24.

[57] Bruce Schneier et al. “On the Twofish Key Schedule”. In: *Selected Areas in Cryptography ’98, SAC’98, Kingston, Ontario, Canada, August 17-18, 1998, Proceedings*. 1998, pp. 27–42. DOI: 10.1007/3-540-48892-8_3. URL: http://dx.doi.org/10.1007/3-540-48892-8_3.

[58] Kyoji Shibutani et al. “Piccolo: An Ultra-Lightweight Blockcipher”. In: *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*. 2011, pp. 342–357. DOI: 10.1007/978-3-642-23951-9_23. URL: http://dx.doi.org/10.1007/978-3-642-23951-9_23.

[59] Bing Sun et al. “Impossible Differential Cryptanalysis of CLEFIA”. In: *IACR Cryptology ePrint Archive* 2008 (2008), p. 151. URL: http://eprint.iacr.org/2008/151.

[60] Siwei Sun et al. “Automatic Enumeration of (Related-key) Differential and Linear Characteristics with Predefined Properties and Its Applications”. In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 747. URL: http://eprint.iacr.org/2014/747.

[61] Siwei Sun et al. “Automatic Security Evaluation and (Related-key) Differential Characteristic Search: Application to SIMON, PRESENT, LBlock, DES(L) and Other Bit-Oriented Block Ciphers”. In: *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*. 2014, pp. 158–178. DOI: 10.1007/978-3-662-45611-8_9. URL: http://dx.doi.org/10.1007/978-3-662-45611-8_9.

[62] Tomoyasu Suzaki et al. “$\textnormal{\textsc{TWINE}}$: A Lightweight Block Cipher for Multiple Platforms”. In: *Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers*. 2012, pp. 339–354. DOI: 10.1007/978-3-642-35999-6_22. URL: http://dx.doi.org/10.1007/978-3-642-35999-6_22.

[63] Cihangir Tezcan. “Differential Factors Revisited: Corrected Attacks on PRESENT and SERPENT”. In: *Lightweight Cryptography for Security and Privacy - 4th International Workshop, LightSec 2015, Bochum, Germany, September 10-11, 2015, Revised Selected Papers*. 2015, pp. 21–33. DOI: 10.1007/978-3-319-29078-2_2. URL: http://dx.doi.org/10.1007/978-3-319-29078-2_2.

[64] Cihangir Tezcan. “Improbable differential attacks on Present using undisturbed bits”. In: *J. Computational Applied Mathematics* 259 (2014), pp. 503–511. DOI: 10.1016/j.cam.2013.06.023. URL: http://dx.doi.org/10.1016/j.cam.2013.06.023.

[65] Cihangir Tezcan. “Improbable differential cryptanalysis”. In: *The 6th International Conference on Security of Information and Networks, SIN ’13, Aksaray, Turkey, November 26-28, 2013*. 2013, p. 457. DOI: 10.1145/2523514.2523587. URL: http://doi.acm.org/10.1145/2523514.2523587.

[66] Cihangir Tezcan. “The Improbable Differential Attack: Cryptanalysis of Reduced Round CLEFIA”. In: *IACR Cryptology ePrint Archive* 2010 (2010), p. 435. URL: http://eprint.iacr.org/2010/435.

[67] Cihangir Tezcan and Ferruh Özbudak. “Differential Factors: Improved Attacks on SERPENT”. In: *Lightweight Cryptography for Security and Privacy - Third International Workshop, LightSec 2014, Istanbul, Turkey, September 1-2, 2014, Revised Selected Papers*. 2014, pp. 69–84. DOI: 10.1007/978-3-319-16363-5_5. URL: http://dx.doi.org/10.1007/978-3-319-16363-5_5.

[68] Cihangir Tezcan and Ferruh Özbudak. “Differential Factors: Improved Attacks on SERPENT”. In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 860. URL: http://eprint.iacr.org/2014/860.

[69] Cihangir Tezcan, Halil Kemal Taskin, and Murat Demircioglu. “Improbable Differential Attacks on Serpent using Undisturbed Bits”. In: *Proceedings of the 7th International Conference on Security of Information and Networks, Glasgow, Scotland, UK, September 9-11, 2014*. 2014, p. 145. DOI: 10.1145/2659651.2659660. URL: http://doi.acm.org/10.1145/2659651.2659660.

[70] Cihangir Tezcan et al. “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*. 2016, pp. 18–32. DOI: 10.1007/978-3-319-55714-4_2. URL: http://dx.doi.org/10.1007/978-3-319-55714-4_2.

[71] Cihangir Tezcan et al. *On Differential Factors*. 9. Uluslararası Bilgi Güvenliği ve Kriptoloji Konferansı,Siber Güvenlik ve Nesnelerin İnterneti,ISCTurkey 2016, Ankara, Turkey. 2016.

[72] Meiqin 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*. 2008, pp. 40–49. DOI: 10.1007/978-3-540-68164-9_4. URL: http://dx.doi.org/10.1007/978-3-540-68164-9_4.

[73] Hongjun Wu et al. “Improved Truncated Differential Attacks on SAFER”. In: *Advances in Cryptology - ASIACRYPT ’98, International Conference on*

the Theory and Applications of Cryptology and Information Security, Beijing, China, October 18-22, 1998, Proceedings. 1998, pp. 133–147. DOI: 10.1007/3-540-49649-1_12. URL: http://dx.doi.org/10.1007/3-540-49649-1_12.

[74] Wenling Wu and Lei Zhang. "LBlock: A Lightweight Block Cipher". In: *IACR Cryptology ePrint Archive* 2011 (2011), p. 345. URL: http://eprint.iacr.org/2011/345.

[75] Gangqiang Yang et al. "The Simeck Family of Lightweight Block Ciphers". In: *Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings*. 2015, pp. 307–329. DOI: 10.1007/978-3-662-48324-4_16. URL: http://dx.doi.org/10.1007/978-3-662-48324-4_16.

[76] Qianqian Yang et al. "Improved Differential Analysis of Block Cipher PRIDE". In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 978. URL: http://eprint.iacr.org/2014/978.

[77] Huihui Yap et al. "EPCBC - A Block Cipher Suitable for Electronic Product Code Encryption". In: *Cryptology and Network Security - 10th International Conference, CANS 2011, Sanya, China, December 10-12, 2011. Proceedings*. 2011, pp. 76–97. DOI: 10.1007/978-3-642-25513-7_7. URL: http://dx.doi.org/10.1007/978-3-642-25513-7_7.

[78] Wentao Zhang et al. "RECTANGLE: A Bit-slice Ultra-Lightweight Block Cipher Suitable for Multiple Platforms". In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 84. URL: http://eprint.iacr.org/2014/084.

[79] Jingyuan Zhao et al. "Differential Analysis on Block Cipher PRIDE". In: *IACR Cryptology ePrint Archive* 2014 (2014), p. 525. URL: http://eprint.iacr.org/2014/525.

[80] Kerem Vari ci, Onur "Ozen, and cCelebi Kocair. *Sarmal: SHA-3 Proposal*. Submission to NIST. 2008. URL: http://www.metu.edu.tr/~e127761/Supporting_Documentation/SarmaL.pdf.

TEZ FOTOKOPİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü ☐

Sosyal Bilimler Enstitüsü ☐

Uygulamalı Matematik Enstitüsü ☐

Enformatik Enstitüsü ☐

Deniz Bilimleri Enstitüsü ☐

YAZARIN

Soyadı : .....
Adı : .....
Bölümü : .....

TEZİN ADI (İngilizce) : .....
.....
.....
.....
.....

TEZİN TÜRÜ : Yüksek Lisans ☐ Doktora ☐

1. Tezimin tamamı dünya çapında erişime açılsın ve kaynak gösterilmek şartıyla tezimin bir kısmı veya tamamının fotokopisi alınsın. ☐
2. Tezimin tamamı yalnızca Orta Doğu Teknik Üniversitesi kullancılarının erişimine açılsın. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐
3. Tezim bir (1) yıl süreyle erişime kapalı olsun. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

Yazarın imzası ..... Tarih .....