Cryptanalysis of Block Ciphers

Jiqiang Lu

Technical Report
RHUL–MA–2008–19
30 July 2008

Department of Mathematics
Royal Holloway, University of London
Egham, Surrey TW20 0EX, England
http://www.rhul.ac.uk/mathematics/techreports

CRYPTANALYSIS OF BLOCK CIPHERS

JIQIANG LU

Thesis submitted to the University of London
for the degree of Doctor of Philosophy

Information Security Group
Department of Mathematics
Royal Holloway, University of London

2008

Declaration

These doctoral studies were conducted under the supervision of Prof. Chris Mitchell.

The work presented in this thesis is the result of original research carried out by myself, in collaboration with others, whilst enrolled in the Information Security Group of Royal Holloway, University of London as a candidate for the degree of Doctor of Philosophy. This work has not been submitted for any other degree or award in any other university or educational establishment.

Jiqiang Lu
July 2008

Acknowledgements

First of all, I thank my supervisor Prof. Chris Mitchell for suggesting block cipher cryptanalysis as my research topic when I began my Ph.D. studies in September 2005. I had never done research in this challenging field before, but I soon found it to be really interesting. Every time I finished a manuscript, Chris would give me detailed comments on it, both editorial and technical, which not only benefitted my research, but also improved my written English. Chris' comments are fantastic, and it is straightforward to follow them to make revisions.

I thank my advisor Dr. Alex Dent for his constructive suggestions, although we work in very different fields.

I thank Prof. Kenny Paterson, who is neither my supervisor nor my advisor, but who gave me many helpful suggestions, and shared useful information with me, including job opportunities.

I thank Prof. Keith Martin for giving me some suggestions on writing a thesis, and Prof. Peter Wild for providing me some funding information.

I thank my co-authors Orr Dunkelman, Nathan Keller, Jongsung Kim, and Changhoon Lee for many fruitful discussions, and my colleagues and friends for the happy time spent together and the help provided.

I thank my PhD examiners Prof. Simon Blackburn and Prof. Lars R. Knudsen for their comments on the thesis.

I thank my master's supervisor Prof. Xinmei Wang, Prof. Yumin Wang and Prof. Guozhen Xiao at Xidian University for initiating me into the field of cryptography during my master studies.

Many thanks go to the administrative and technical staff at the department and the university for their support. I am highly impressed by their understanding and high-quality services.

Special thanks go to my wife Xiaoyan Yan for her support, who had to get accustomed to a rather different culture, has experienced and is still to experience every moment of my happiness and sadness.

Lastly, I am grateful for the British Chevening / Royal Holloway Scholarship awarded

to me before I started study, which removed all my concerns about living, and enabled me to concentrate solely on my research; more interestingly, it entitled me to a 30-day interrail pass around Europe. Without its generous sponsorship, I guess that my PhD study would have been unlikely. I also thank both the department and the ECRYPT (European Network of Excellence for Cryptology) project of the European Commission for supporting my travel around the world to attend a number of academic events, including conferences, workshops and summer schools.

Abstract

The block cipher is one of the most important primitives in modern cryptography, information and network security; one of the primary purposes of such ciphers is to provide confidentiality for data transmitted in insecure communication environments. To ensure that confidentiality is robustly provided, it is essential to investigate the security of a block cipher against a variety of cryptanalytic attacks.

In this thesis, we propose a new extension of differential cryptanalysis, which we call the impossible boomerang attack. We describe the early abort technique for (related-key) impossible differential cryptanalysis and rectangle attacks. Finally, we analyse the security of a number of block ciphers that are currently being widely used or have recently been proposed for use in emerging cryptographic applications; our main cryptanalytic results are as follows.

- An impossible differential attack on 7-round AES when used with 128 or 192 key bits, and an impossible differential attack on 8-round AES when used with 256 key bits. An impossible boomerang attack on 6-round AES when used with 128 key bits, and an impossible boomerang attack on 7-round AES when used with 192 or 256 key bits. A related-key impossible boomerang attack on 8-round AES when used with 192 key bits, and a related-key impossible boomerang attack on 9-round AES when used with 256 key bits, both using two keys.
- An impossible differential attack on 11-round reduced Camellia when used with 128 key bits, an impossible differential attack on 12-round reduced Camellia when used with 192 key bits, and an impossible differential attack on 13-round reduced Camellia when used with 256 key bits.
- A related-key rectangle attack on the full Cobra-F64a, and a related-key differential attack on the full Cobra-F64b.
- A related-key rectangle attack on 44-round SHACAL-2.
- A related-key rectangle attack on 36-round XTEA.
- An impossible differential attack on 25-round reduced HIGHT, a related-key rectangle attack on 26-round reduced HIGHT, and a related-key impossible differential attack on 28-round reduced HIGHT.

In terms of either the attack complexity or the numbers of attacked rounds, the attacks presented in the thesis are better than any previously published cryptanalytic results for the block ciphers concerned, except in the case of AES; for AES, the presented impossible differential attacks on 7-round AES used with 128 key bits and 8-round AES used with 256 key bits are the best currently published results on AES in a single key attack scenario, and the presented related-key impossible boomerang attacks on 8-round AES used with 192 key bits and 9-round AES used with 256 key bits are the best currently published results on AES in a related-key attack scenario involving two keys.

Contents

List of Figures

List of Tables

Abbreviations

ACPC	Adaptive Chosen Plaintexts and Ciphertexts
AES	Advanced Encryption Standard
AES-128/192/256	AES when used with a key length of 128, 192 or 256 bits
Camellia-128/192/256	Camellia when used with a key length of 128, 192 or 256 bits
CP	Chosen Plaintexts
CRYPTREC	Cryptography Research and Evaluation Committees *http://www.cryptrec.jp/english*
DES	Data Encryption Standard
IEEE	The Institute of Electrical and Electronics Engineers *http://www.ieee.org*
IETF	The Internet Engineering Task Force *http://www.ietf.org*
IPsec	Internet Protocol security
ISO	International Organization for Standardization *http://www.iso.org*
KP	Known Plaintexts
MA	Memory Accesses
MAC	Message Authentication Code
NESSIE	New European Schemes for Signatures, Integrity, and Encryption *https://www.cosic.esat.kuleuven.be/nessie*
NIST	National Institute of Standards and Technology, U.S.A. *https://www.nist.gov*
RFID	Radio-Frequency IDentification
RK-CP	Related-Key Chosen Plaintexts
SPN	Substitution-Permutation Network

Notation

Throughout this thesis, a number without a prefix is in decimal (base 10) notation, a number with prefix $0x$ is in hexadecimal (base 16) notation, and a number preceded and followed by $\langle$ and $\rangle_2$ is in binary (base 2) notation. The bits of an n-bit value are numbered from 1 to n from left to right. We use the following notation.

$\oplus$	bitwise logical exclusive OR (XOR) of two bit strings of the same length
$\&$	bitwise logical AND of two bit strings of the same length
$\neg$	bitwise logical complement of a bit string
$\odot$	dot product of two bit strings of the same length
$\boxplus$	addition modulo 2^n
$\boxminus$	subtraction modulo 2^n
$\boxtimes$	multiplication modulo 2^n
$<<$ $(>>)$	left (right) shift of a bit string
$\lll$ $(\ggg)$	left (right) rotation of a bit string
$\|\|$	string concatenation
Δ	difference with respect to the $\oplus$ operation
$\circ$	functional composition. When composing functions $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{X} \circ \mathbf{Y}$ denotes the function obtained by first applying $\mathbf{X}$ and then applying $\mathbf{Y}$
e	the base of the natural logarithm, $(\mathsf{e} = 2.71828\ldots)$
$\|X\|$	the number of elements in a set X
$\lfloor x \rfloor$	the largest integer that is less than or equal to x
e_j	an n-bit value with zeros everywhere except for bit position j, $(1 \leq j \leq n)$
$e_{i_1,\cdots,i_j}$	the n-bit word equal to $e_{i_1} \oplus \cdots \oplus e_{i_j}$, $(1 \leq i_1, \cdots, i_j \leq n)$
$e_{j,\sim}$	an n-bit value that has zeros in bit positions 1 to $j-1$, a one in bit position j and indeterminate values in bit positions $(j+1)$ to n, $(1 \leq j \leq n-1)$
$\overline{e}_{j,\sim}$	an n-bit value that has zeros in bit positions 1 to j and indeterminate values in bit positions $(j+1)$ to n, $(1 \leq j \leq n-1)$
$\star$	an arbitrary n-bit value, where two values represented by the $\star$ symbol may be different
$X \sim \text{Poi}(\lambda)$	a random variable X follows the Poisson distribution with parameter λ, where λ is the expected value of X. See [72] for the details of Poisson distribution
$X \sim \text{Bin}(N, p)$	a random variable X follows the binomial distribution with parameters N and p, where N is the number of trials, and p is the success rate for each trial. See [72] for the details of binomial distribution

Chapter 1

Introduction

In this chapter, we give the motivation for our research. We also describe the contributions of this thesis, present its overall structure, and give the notation used throughout this thesis.

Contents

1.1 Motivation

Since the first computer network was established in 1956, Internet technologies have developed very quickly. A wide variety of communications networks, including Public Switched Telephone Networks (PSTNs), Public Switched Data Networks (PSDNs), Integrated Service Networks (ISNs) and mobile communication systems, are becoming ever more important in our daily lives, and they have greatly changed the way we live. As a consequence, as the science of secure communications, cryptology has received considerable attention.

Cryptology has two main branches — cryptography and cryptanalysis. Cryptography is the study of how to design algorithms that provide confidentiality, authenticity, integrity and other security-related services for data transmitted in insecure communication environments. Confidentiality protects data from leaking to unauthorised users. Authenticity provides assurance regarding the identity of a communicating party, which protects against impersonation. Integrity protects data against being modified (or at least enables modifications to be detected).

Modern cryptography involves secret-key (symmetric) cryptography and public-key (asymmetric) cryptography. In secret-key cryptography, when using a secret-key encryption algorithm, the sender and receiver of a message use the same secret key; the sender uses the secret key to encrypt the message, and the receiver uses the same secret key to decrypt the message. In public-key cryptography, introduced in 1976 by Diffie and Hellman [23], each participating party has a pair of keys, one called the public key and the other called the private key; the public key is typically published in a trusted directory, while the private key is kept secret. When using a public-key encryption algorithm, the sender uses the public key of the receiver to encrypt the message, and the receiver uses his/her private key to decrypt the message.

Cryptanalysis studies how to evaluate or break cryptographic algorithms. This helps to enable more secure algorithms to be designed.

The block cipher is an important primitive in secret-key cryptography; one main purpose of a block cipher is to provide confidentiality for data transmitted in insecure communication environments. A block cipher can also be used to build other secret-key cryptographic primitives, such as stream ciphers, hash functions, message authentication codes (MACs), and cryptographically secure pseudorandom number generators. Block ciphers are also widely used as a fundamental component in public-key cryptography, information security, network security, computer security, and other security applications. It is thus of great importance to investigate the security of a block cipher algorithm against a variety of cryptanalytic attacks.

1.2 Contributions

In this thesis we propose a new extension of differential cryptanalysis, which we call the impossible boomerang attack. We describe the early abort technique for (related-key) impossible differential cryptanalysis and rectangle attacks. Finally, we analyse the security of a number of block ciphers that are currently being widely used or have recently been proposed for use in emerging cryptographic applications; our main cryptanalytic results are as follows.

- An impossible differential attack on 7-round AES-128, 7-round AES-192, and

8-round AES-256. An impossible boomerang attack on 6-round AES-128, 7-round AES-192, and 7-round AES-256. A related-key impossible boomerang attack on 8-round AES-192 and 9-round AES-256, using two keys.

- An impossible differential attack on 11-round reduced Camellia-128, 12-round reduced Camellia-192, and 13-round reduced Camellia-256.
- A related-key rectangle attack on the full Cobra-F64a, and a related-key differential attack on the full Cobra-F64b.
- A related-key rectangle attack on 44-round SHACAL-2.
- A related-key rectangle attack on 36-round XTEA.
- An impossible differential attack on 25-round reduced HIGHT, a related-key rectangle attack on 26-round reduced HIGHT, and a related-key impossible differential attack on 28-round reduced HIGHT.

In terms of either the attack complexity or the numbers of attacked rounds, the attacks presented in the thesis are better than any previously published cryptanalytic results for the block ciphers concerned, except in the case of AES; for AES, the presented impossible differential attacks on 7-round AES-128 and 8-round AES-256 are the best currently published results on AES in a single key attack scenario, and the presented related-key impossible boomerang attacks on 8-round AES-192 and 9-round AES-256 are the best currently published results on AES in a related-key attack scenario using two keys.

Some of the results described in the thesis have previously been presented in [74, 75, 76, 77, 78, 79, 80].

1.3 Organisation of Thesis

The remainder of this thesis is organised as follows.

Literature review: In Chapter 2, we briefly review a number of currently known cryptanalytic methods for block ciphers. The cryptanalytic methods discussed include differential cryptanalysis, linear cryptanalysis, differential-linear cryptanalysis, impossible differential cryptanalysis, boomerang and rectangle attacks, integral cryptanalysis, and related-key cryptanalysis.

Our new cryptanalytic results: In Chapter 3, we propose the (related-key) impossible boomerang attack. In Chapter 4, we give a general description of the early abort technique for impossible differential cryptanalysis and the rectangle attack.

In Chapters 5–10, we present our new cryptanalytic results on AES, Camellia, Cobra-F64a and Cobra-F64b, SHACAL-2, XTEA, and HIGHT, respectively. In each chapter we start with a description of the block cipher concerned, followed by a review of the previously published cryptanalytic results for this cipher. We then present our new cryptanalytic results. Finally, we compare our new cryptanalytic results with the previous state of the art.

Conclusions: In Chapter 11, we provide a summary of the main results in this thesis, and give some possible directions for future research.

1.4 Notation

Throughout this thesis, a number without a prefix is in decimal (base 10) notation, a number with prefix $0x$ is in hexadecimal (base 16) notation, and a number preceded and followed by $\langle$ and $\rangle_2$ is in binary (base 2) notation. The bits of an n-bit value are numbered from 1 to n from left to right. We use the following notation.

- $\oplus$: bitwise logical exclusive OR (XOR) of two bit strings of the same length
- &: bitwise logical AND of two bit strings of the same length
- $\neg$: bitwise logical complement of a bit string
- $\odot$: dot product of two bit strings of the same length
- $\boxplus$: addition modulo 2^n
- $\boxminus$: subtraction modulo 2^n

- $\boxtimes$: multiplication modulo 2^n
- $<<$ ($>>$): left (right) shift of a bit string
- $\lll$ ($\ggg$): left (right) rotation of a bit string
- $||$: string concatenation
- Δ: difference with respect to the $\oplus$ operation
- $\circ$: functional composition. When composing functions $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{X}\circ\mathbf{Y}$ denotes the function obtained by first applying $\mathbf{X}$ and then applying $\mathbf{Y}$
- $\mathtt{e}$: the base of the natural logarithm, ($\mathtt{e} = 2.71828\ldots$)
- $|X|$: the number of elements in a set X
- $\lfloor x \rfloor$: the largest integer that is less than or equal to x
- e_j: an n-bit value with zeros everywhere except for bit position j, $(1 \leq j \leq n)$
- $e_{i_1,\cdots,i_j}$: the n-bit word equal to $e_{i_1} \oplus \cdots \oplus e_{i_j}$, $(1 \leq i_1, \cdots, i_j \leq n)$
- $e_{j,\sim}$: an n-bit value that has zeros in bit positions 1 to $j-1$, a one in bit position j and indeterminate values in bit positions $(j+1)$ to n, $(1 \leq j \leq n-1)$
- $\overline{e}_{j,\sim}$: an n-bit value that has zeros in bit positions 1 to j and indeterminate values in bit positions $(j+1)$ to n, $(1 \leq j \leq n-1)$
- $\star$: an arbitrary n-bit value, where two values represented by the $\star$ symbol may be different
- $X \sim \mathrm{Poi}(\lambda)$: a random variable X follows the Poisson distribution with parameter λ, where λ is the expected value of X. See [72] for the details of Poisson distribution
- $X \sim \mathrm{Bin}(N, p)$: a random variable X follows the binomial distribution with parameters N and p, where N is the number of trials, and p is the success rate for each trial. See [72] for the details of binomial distribution

Chapter 2

Block Cipher Cryptanalysis

In this chapter we first give a definition of a block cipher. We then briefly review a number of cryptanalytic methods for block ciphers, including differential cryptanalysis, linear cryptanalysis, differential-linear cryptanalysis, impossible differential cryptanalysis, boomerang and rectangle attacks, and related-key cryptanalysis. These techniques underlie the results presented in the remainder of this thesis.

Contents

2.1 Introduction

A block cipher is an algorithm that transforms a fixed-length data block, called a plaintext block, into another data block of the same length, called a ciphertext block, under the control of a secret key. Ideally, the set of transformations induced by the set of all possible secret keys should be indistinguishable from a random set of transformations. If a block cipher has a plaintext/ciphertext block length of n bits, then we refer to it as an n-bit block cipher. Currently, the widely used block lengths are 64 and 128 bits, and the key length is typically 128, 192 or 256 bits. Note that in the field of block cipher cryptanalysis the term ‘ciphertext’ is sometimes abused slightly to mean the result of encrypting a plaintext block using a reduced version of the block cipher concerned.

In practice, almost all block ciphers are constructed by repeating a simple function many times, known as the iterated method. The repeated function is called the round function, every iteration is called a round, the key used in every round is called a round subkey, and the number of iterations is called the number of rounds of the block cipher.

An iterated block cipher involves three sub-algorithms — an encryption algorithm, a decryption algorithm and a key schedule algorithm. The encryption algorithm takes a plaintext block as input, and outputs a ciphertext block, under the control of a secret key. The decryption algorithm is the inverse of the encryption algorithm, when under the control of the same secret user key. The key schedule algorithm takes a secret user key as input, and generates the required round subkeys.

Most block ciphers are examples of one of two special types of iterated ciphers, known as Feistel ciphers and Substitution-Permutation Networks (SPNs). In a Feistel cipher, the plaintext is split into two halves. The round function is applied to one half, and the output of the round function is bitwise exored with the other half; finally, the two halves are swapped, and become the two halves of the next round. The Data Encryption Standard (DES) block cipher [91] is an example of a Feistel cipher. In an SPN cipher, the round function is applied to the whole block, and its output becomes the input of the next round. The Advanced Encryption Standard (AES) block cipher [90] is an example of an SPN. There also exist block ciphers with other round structures; one such example is the IDEA block cipher [66]. One major difference between these two approaches is that, for a Feistel cipher, the round function can be chosen arbitrarily, whereas, for an SPN, the round function must be bijective (invertible). A Feistel structure whose round function is bijective is called a Feistel structure with a bijective round function.

A round function for an iterative block cipher is typically made up of a bitwise XOR with a round subkey, followed by sub-block substitutions using non-linear S-boxes, i.e. fixed functions taking a string of bits as input and giving a string of bits as output, and, finally, a bit-level permutation. The S-boxes used need to be bijective for an SPN, but can be arbitrarily chosen for a Feistel cipher. An S-box with an m-bit input and n-bit output is called an $m \times n$ S-box.

The notion of 'branch number' [20] is sometimes used to measure the diffusion power of a linear transformation operating on byte tuples, and is defined as follows.

Definition 2.1 *Suppose that $B = \{0, 1\}^8$ and $\mathbb{N} = \{0, 1, 2, 3, \cdots\}$. Let $W : B^* \rightarrow \mathbb{N}$ be the function returning the number of non-zero bytes of an input byte tuple. The branch number of a linear transformation $L : B^m \rightarrow B^n$ (for specific values of m and n) is defined to equal the minimum value of $W(x)+W(L(x))$, where $x \in B^m - \{0^m\}$.*

2.2 Cryptanalytic Methods

In the most general sense, a cryptanalytic attack is an algorithm that distinguishes a cryptosystem from a random function (that operates on data blocks of the same length). The effectiveness of an attack is usually measured using the following three metrics.

- Data complexity: the numbers of plaintexts and/or ciphertexts required for execution of the attack.
- Memory (storage) complexity: the amount of memory required for execution of the attack.
- Time (computational) complexity: the amount of computation or time required for execution of the attack. In block cipher cryptanalysis, this is usually measured in terms of how many encrytions/decryptions of the block cipher or memory accesses are required.

2.2.1 Cryptanalysis Scenarios

We start this review of cryptanalytic techniques by considering what assumptions are normally made regarding the resources of a cryptanalyst.

It is generally agreed that any cryptosystem should meet Kerckhoffs' principle.

Kerckhoffs' principle [51] *A cryptosystem should be secure even if everything about the system, except the secret key, is public knowledge.*

Kerckhoffs' principle says that the security of a cryptosystem should rely solely on the secret key, rather than on the secrecy of the cryptographic algorithm. In other words, the cryptosystem should be secure even if an attacker knows everything about the cryptographic algorithm except the secret key.

Following from Kerckhoffs' principle, there are four widely discussed attack scenarios, each giving slightly differing resources to a cryptanalyst. Each scenario gives the cryptanalyst more resources than the previous, and, in general, it is highly desirable for any cryptosystem to be secure even in the final scenario.

- Ciphertext-only attack scenario. In this scenario the attacker is assumed to have access to a number of ciphertexts. The attacker is also assumed to have some information about the plaintext, e.g. that conforms to certain formatting constraints or that it is written in a particular natural language. (If no information about the plaintext is available, then it is theoretically impossible to perform cryptanalysis, except to observe that repeated ciphertext blocks correspond to repeated plaintext blocks).
- Known-plaintext attack scenario. Here, the attacker is assumed to have access to a number of ciphertexts and the corresponding plaintexts for at least some of the ciphertexts.
- Chosen-plaintext/cipertext attack scenario. In this case the attacker can choose a number of plaintexts (and/or ciphertexts), and be given the corresponding ciphertexts (and/or plaintexts).

- Adaptive chosen plaintext and ciphertext attack scenario. In this final case the attacker can choose plaintexts (and/or ciphertexts) and be given the corresponding ciphertexts (and/or plaintexts). Based on the information obtained, the attacker can then choose further plaintexts/ciphertexts, and be given the corresponding ciphertexts/plaintexts. This process can be iterated.

2.2.2 Elementary Techniques

We next describe three fundamental cryptanalytic techniques that can be applied to any block cipher. In the description below (and throughout the thesis) we assume the use of an n-bit block cipher with a k-bit user key, i.e. the cipher is a function $\mathbf{E} : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$, where we write the key input as a subscript, i.e. if $K \in \{0,1\}^k$ is a key, and $P \in \{0,1\}^n$ is a plaintext block, then the ciphertext is denoted by $\mathbf{E}_K(P)$; sometimes, if there is no ambiguity about the key in use, we simply write $\mathbf{E}(P)$. Note that, for any fixed K, the restricted function $\mathbf{E}_K$ acts as a permutation on the set $\{0,1\}^n$, since otherwise unique decryption will not be possible.

- A dictionary attack involves an attacker building and maintaining a table containing all 2^k possible ciphertext blocks corresponding to a particular plaintext block, with one entry in the table for each possible key. If the attacker obtains an enciphered version of the particular plaintext block, then he can deduce the key from the table with high probability, as long as $n \geq k$ (if $n < k$, then the expected number of possible keys will be reduced to 2^{k-n}). This attack has a data complexity of 2^k ciphertexts, a 2^k n-bit memory complexity and a negligible time complexity. Moreover, it requires a one-off precomputation to generate the table, which has a time complexity of 2^k encryptions; however, the time complexity of the precomputation is typically not counted as part of the time complexity of an attack, since it can be performed at the cryptanalyst's leisure [36].

- A codebook attack requires an attacker to build and maintain a table of the 2^n ciphertexts for the 2^n plaintexts encrypted using one particular (unknown) key. The table is sorted by plaintext, thus and only the 2^n ciphertexts need to be stored in the table. When the attacker gets a ciphertext, he can deduce the corresponding plaintext from the table, provided that the particular key is used. Such an attack has a data complexity of 2^n plaintext/ciphertext pairs, a 2^n n-bit memory complexity and a negligible time complexity.

- An exhaustive key search (or brute force search) attack involves an attacker trying every possible key, given a known plaintext/ciphertext pair. The correct key will yield the correct correspondence between plaintext and ciphertext; if more than one candidate key is produced, then the incorrect candidates can be eliminated using one or more additional pairs. Such an attack has a time complexity of 2^k encryptions and negligible data and memory complexities.

An attack is commonly regarded as effective if it is faster (i.e. it has lower time complexity) than an exhaustive key search. In recent years, a variety of cryptanalytic methods have been proposed, of which differential cryptanalysis [12] and linear cryptanalysis [83] are probably the best known. All of these techniques are tradeoffs between data, time and/or memory complexities [36], compared with the above three elementary cryptanalytic techniques.

In this chapter, we briefly review a range of cryptanalytic techniques, including differential cryptanalysis, linear cryptanalysis, differential-linear cryptanalysis, impossible differential cryptanalysis, boomerang and rectangle attacks, and related-key cryptanalysis. These methods all exploit statistical relationships between a block cipher's inputs and outputs, in particular between the inputs and outputs of the nonlinear S-boxes.

2.2.3 Mathematical Background

We first review several types of discrete probability distributions, including Bernoulli distribution, binomial distribution and Poisson distribution, which are often used in the statistical cryptanalysis methods to be described below. See [72] for their detailed introduction.

2.2.3.1 Fundamental Notions

A sample space represents the individual, distinct outcomes in which a random experiment can terminate. An event is any set of outcomes in a sample space. The probability of an event, A say, is the sum of the probabilities assigned to the outcomes that make up A, denoted by $\Pr(A)$. A probability space, denoted by $(\Omega, \mathcal{F}, P)$, involves a sample space Ω, an non-empty collection $\mathcal{F}$ of subsets of Ω, and a probability function P defined on $\mathcal{F}$. A discrete (real-valued) random variable X on a probability space $(\Omega, \mathcal{F}, P)$ is a function X with domain Ω and range a finite or countably infinite subset $(x_1, x_2, \cdots)$ of the real numbers $\mathbb{R}$ such that $\{\omega \in \Omega | X(w) = x_i\}$ is an event for all i. The expected value (or mathematical expectation) of a discrete random variable is the sum of the probability of each possible outcome of the experiment multiplied by the outcome value. A probability mass function, denoted by $f_X(x)$, is a function that gives the probability that a discrete random variable X is equal to some value x. A probability distribution function, denoted by $F_X(x)$, is a function that describes the probability distribution of a discrete random variable X, which is defined to equal $\Pr(X \leq x)$.

2.2.3.2 Bernoulli Distribution

Bernoulli distribution is a finite discrete probability distribution where a random variable can take on only two values. The two values are usually 0 and 1, where 0 and 1 are artificial; for example, we can let 0 and 1 respectively denote failure and success of a test.

Let $\Pr(X = 1) = p$, then we have $\Pr(X = 0) = 1 - \Pr(X = 1) = 1 - p$. The mathematical expectation of such a Bernoulli distribution is $p \times 1 + (1 - p) \times 0 = p$.

2.2.3.3 Binomial Distribution

Binomial distribution is a finite discrete probability distribution where a random variable can be represented to be the sum of the successive results of independent trials of a Bernoulli experiment. Suppose that $X_1, X_2, \cdots, X_N$ are the results of N

independent trials of a Bernoulli experiment (as described in Section 2.2.3.2). Let a random variable Y be the sum of $X_1, X_2, \cdots, X_N$, (i.e. $Y = X_1 + X_2 + \cdots + X_N$), then the distribution of Y is defined by the following probability function, where $k = 0, 1, \cdots, N$.

$$\begin{aligned}\Pr(Y = k) &= \Pr(X_1 + X_2 + \cdots + X_N = k) \\ &= \binom{N}{k} p^k (1-p)^{N-k}.\end{aligned}$$

This distribution is called the binomial distribution with parameters n and p, written $Y \sim \mathrm{Bin}(N, p)$. The mathematical expectation of a binomial distribution $Y \sim \mathrm{Bin}(N, p)$ is Np.

2.2.3.4 Poisson Distribution

Poisson distribution is a countably infinite discrete probability distribution. A random variable X is said to have a Poisson distribution with parameter λ if and only if

$$\Pr(X = k) = \mathsf{e}^{-\lambda}\lambda^k/k!, \quad k = 0, 1, 2, \cdots,$$

where $\mathsf{e}(= 2.71828\cdots)$ is the base of the natural logarithm.

We write $X \sim \mathrm{Poi}(\lambda)$ for a random variable X having a Poisson distribution with parameter λ. The mathematical expectation of a Poisson distribution $X \sim \mathrm{Poi}(\lambda)$ is λ.

2.2.3.5 Relationship between Binomial Distribution and Poisson Distribution

A Binomial distribution $Y \sim \mathrm{Bin}(N, p)$ can be approximated with a Poisson distribution $Y \sim \mathrm{Poi}(Np)$ when N is large and p is small. That is,

$$\binom{N}{k} p^k (1-p)^{N-k} \approx \mathsf{e}^{-Np}(Np)^k/k!, \quad k = 0, 1, 2, \cdots, N.$$

A proof of this relationship is given in [72].

2.2.4 Differential Cryptanalysis

Differential cryptanalysis was introduced in 1990 by Biham and Shamir [12]; it was the first cryptanalytic method more effective than an exhaustive key search to be proposed for the full DES [13, 14]. A similar method was used a little earlier by Murphy [87] to analyse the FEAL block cipher [97].

Differential cryptanalysis takes advantage of how a specific difference in a pair of inputs of a cipher or function can affect a difference in the pair of outputs of the cipher or function, where the pair of outputs are obtained by encrypting the pair of inputs using the same key. The notion of difference can be defined in several ways; the most widely discussed is with respect to the XOR operation. The difference between the inputs is called the input difference, the difference between the outputs of a function is called the output difference, and the difference between internal values is called an intermediate difference. The combination of the input difference and the output difference is called a differential. The probability of a differential for an S-box is defined as follows.

Definition 2.2 *Suppose* $\mathbf{T}$ *is an* $m \times n$ *S-box. If* γ *is an* m*-bit block and* δ *is an* n*-bit block, then the probability of the differential* (γ, δ) *for* $\mathbf{T}$*, written* $\Delta\gamma \rightarrow \Delta\delta$*, is defined to be*

$$\mathrm{Pr}_{\mathbf{T}}(\Delta\gamma \rightarrow \Delta\delta) = \Pr_{P \in \{0,1\}^m}(\mathbf{T}(P) \oplus \mathbf{T}(P \oplus \gamma) = \delta).$$

The following result follows trivially from Definition 2.2:

Proposition 2.1 *If* $\mathbf{T}$ *is an* $m \times n$ *S-box, then*

$$\mathrm{Pr}_{\mathbf{T}}(\Delta\gamma \rightarrow \Delta\delta) = \frac{|\{x \in \{0,1\}^m | \mathbf{T}(x) \oplus \mathbf{T}(x \oplus \gamma) = \delta\}|}{2^m}.$$

The (XOR) difference distribution table for an $m \times n$ S-box $\mathbf{T}$ is a table storing all possible pairs of input and output differences (γ, δ) and the numbers of m-bit blocks x ($\in \{0,1\}^m$) such that $\mathbf{T}(x) \oplus \mathbf{T}(x \oplus \gamma) = \delta$.

The probability of a differential for a block cipher using a particular key is defined as follows.

Definition 2.3 *Suppose* $\mathbf{E}$ *is an* n*-bit block cipher and* $K \in \{0,1\}^k$ *is a key for* $\mathbf{E}$*. If* α *and* β *are* n*-bit blocks, then the probability of the differential* (α, β) *for* $\mathbf{E}_K$*, written* $\Delta\alpha \to \Delta\beta$*, is defined to be*

$$\Pr_{\mathbf{E}_K}(\Delta\alpha \to \Delta\beta) = \Pr_{P \in \{0,1\}^n}(\mathbf{E}_K(P) \oplus \mathbf{E}_K(P \oplus \alpha) = \beta).$$

The following result follows trivially from Definition 2.3:

Proposition 2.2 *If* $\mathbf{E}$ *is an* n*-bit block cipher, and* $K \in \{0,1\}^k$ *is a key for* $\mathbf{E}$*, and* α *and* β *are* n*-bit blocks. Then*

$$\Pr_{\mathbf{E}_K}(\Delta\alpha \to \Delta\beta) = \frac{|\{x|\mathbf{E}_K(x) \oplus \mathbf{E}_K(x \oplus \alpha) = \beta, x \in \{0,1\}^n\}|}{2^n}.$$

Sometimes we refer to the differential of a block cipher without specifying the key. For most currently studied block ciphers the differential probabilities do not depend on the key used, and so this is reasonable practice.

There may be a number of different intermediate differences that give rise to the same differential. A sequence of intermediate differences that give rise to a particular differential is called a differential characteristic. That is, a differential is the set of all the differential characteristics with the same input difference and output difference.

A differential (characteristic) for r consecutive rounds is often called an r-round differential (characteristic). An r-round differential (characteristic) that has a probability of p is often called an r-round differential (characteristic) with probability p.

Given a set of $\frac{c}{\Pr_{\mathbf{E}_K}(\Delta\alpha\to\Delta\beta)}$ pairs of plaintexts with difference α, (for some $c > 1$), then, if they are all encrypted using the key K, the expected number of pairs of ciphertexts with a difference of β is equal to $\frac{c}{\Pr_{\mathbf{E}_K}(\Delta\alpha\to\Delta\beta)} \cdot \Pr_{\mathbf{E}_K}(\Delta\alpha \to \Delta\beta) = c$. If, on the other hand, these pairs are input to a randomly chosen function, then the expected number of pairs of outputs with a difference of β is equal to $\frac{c}{\Pr_{\mathbf{E}_K}(\Delta\alpha\to\Delta\beta)} \cdot 2^{-n} = \frac{c}{2^n \cdot \Pr_{\mathbf{E}_K}(\Delta\alpha\to\Delta\beta)}$. Therefore, if $\Pr_{\mathbf{E}_K}(\Delta\alpha \to \Delta\beta)$ is larger than 2^{-n}, we can use the differential to distinguish the block cipher from a randomly chosen function, given a sufficient number of chosen plaintext pairs.

Definition 2.4 *With respect to a particular differential characteristic, an active S-box is defined to be an S-box that has a non-zero input difference, and an inactive S-box is defined to be an S-box that has a zero input difference.*

Several extensions to differential cryptanalysis have been proposed, including high-order differential cryptanalysis [56, 65], truncated differential cryptanalysis [56], impossible differential cryptanalysis [4, 57], and the boomerang and rectangle attacks [6, 48, 103]. In this chapter we restrict our attention to impossible differential cryptanalysis and the boomerang and rectangle attacks.

2.2.5 Linear Cryptanalysis

Linear cryptanalysis was introduced in 1992 by Matsui and Yamagishi [84], who used it to analyse the FEAL cipher. In 1993, Matsui [83] presented a linear cryptanalysis attack on the full DES.

Linear cryptanalysis exploits correlations between a particular linear function of the input blocks and a second linear function of the output blocks. The most widely used linear function involves computing the bitwise dot product operation of the block with a specific binary vector (the specific value combined with the input blocks may be different from the value applied to the output blocks). The combination of the two linear functions is called a linear approximation. The probability of a linear approximation is defined as follows.

Definition 2.5 *Suppose* $\mathbf{E}$ *is an* n*-bit block cipher and* $K \in \{0,1\}^k$ *is a key for* $\mathbf{E}$*. If* α *and* β *are* n*-bit blocks, then the probability of the linear approximation* (α, β)*, written* $\Gamma\alpha \to \Gamma\beta$*, is defined to be*

$$\Pr_{\mathbf{E}_K}(\Gamma\alpha \to \Gamma\beta) = \Pr_{P \in \{0,1\}^n}(P \odot \alpha = \mathbf{E}_K(P) \odot \beta),$$

where $\odot$ *represents the dot product of two bit strings regarded as binary vectors.*

We refer to the dot product $P \odot \alpha$ as the input parity, and the dot product $\mathbf{E}_K(P) \odot \beta$ as the output parity.

The following result follows trivially from Definition 2.5:

Proposition 2.3 *If* $\mathbf{E}$ *is an* n*-bit block cipher, and* $K \in \{0,1\}^k$ *is a key for* $\mathbf{E}$*, and* α *and* β *are* n*-bit blocks. Then*

$$\mathrm{Pr}_{\mathbf{E}_K}(\Gamma\alpha \to \Gamma\beta) = \frac{|\{x|x \odot \alpha = \mathbf{E}_K(x) \odot \beta, x \in \{0,1\}^n\}|}{2^n}.$$

For a randomly chosen function, the expected probability of a linear approximation for any pair (α, β) is $\frac{1}{2}$.

Definition 2.6 *The bias of a linear approximation* $\Gamma\alpha \to \Gamma\beta$*, denoted by* ϵ*, is defined to be*

$$\epsilon = |\mathrm{Pr}_{\mathbf{E}_K}(\Gamma\alpha \to \Gamma\beta) - \frac{1}{2}|.$$

Thus, if the bias ϵ is sufficiently large, we can use the linear approximation to distinguish a block cipher from a randomly chosen function, given a sufficient number of matching plaintext/ciphertext pairs.

Several extensions to linear cryptanalysis have been proposed, including bilinear cryptanalysis [17], linear cryptanalysis using multiple approximations [46], linear cryptanalysis using nonlinear approximations [59] and linear cryptanalysis using chosen plaintexts [58].

2.2.6 Differential-Linear Cryptanalysis

Differential-linear cryptanalysis was introduced in 1994 by Langford and Hellman [67]; it is a combination of differential and linear cryptanalysis. In 2002, Biham, Dunkelman and Keller [7] presented an enhanced version.

Proposition 2.4 *Suppose block cipher* $\mathbf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ *is represented as a cascade of two sub-ciphers* $\mathbf{E} = \mathbf{E}^0 \circ \mathbf{E}^1$ *and* $K \in \{0,1\}^k$ *is a key for* $\mathbf{E}$*. Suppose also that there exists a differential* $\Delta\alpha \to \Delta\beta$ *with probability* p *for* $\mathbf{E}^0_K$ *and a linear approximation* $\Gamma\gamma \to \Gamma\delta$ *with bias* ϵ *for* $\mathbf{E}^1_K$*. If* P' *is chosen uniformly at random from* $\{0,1\}^n$*, and* $P^* = P' \oplus \alpha$*, then (under an assumption about the* *random behaviour of* **E***)*

$$\Pr(\delta \odot \mathbf{E}_K(P') \oplus \delta \odot \mathbf{E}_K(P^*) = \gamma \odot \beta) = \frac{1}{2} + 2p\epsilon^2.$$

Proof. Given a plaintext pair $(P', P^* = P' \oplus \alpha)$, where P' is chosen uniformly at random from $\{0,1\}^n$, we obtain $\mathbf{E}^0_K(P') \oplus \mathbf{E}^0_K(P^*) = \beta$ with probability p, $\gamma \odot \mathbf{E}^0_K(P') = \delta \odot \mathbf{E}_K(P')$ with bias ϵ, and $\gamma \odot \mathbf{E}^0_K(P^*) = \delta \odot \mathbf{E}_K(P^*)$ with bias ϵ. We therefore obtain $\delta \odot \mathbf{E}_K(P') \oplus \delta \odot \mathbf{E}_K(P^*) = \gamma \odot \beta$ with a probability of

$$p \cdot [(\frac{1}{2} + \epsilon) \cdot (\frac{1}{2} + \epsilon) + (\frac{1}{2} - \epsilon) \cdot (\frac{1}{2} - \epsilon)] = p(\frac{1}{2} + 2\epsilon^2).$$

If $\mathbf{E}^0_K(P') \oplus \mathbf{E}^0_K(P^*) \neq \beta$, we assume that the value of $\delta \odot \mathbf{E}_K(P') \oplus \delta \odot \mathbf{E}_K(P^*)$ is distributed uniformly. Hence $\delta \odot \mathbf{E}_K(P') \oplus \delta \odot \mathbf{E}_K(P^*) = \gamma \odot \beta$ with probability

$$p(\frac{1}{2} + 2\epsilon^2) + (1-p) \cdot \frac{1}{2} = \frac{1}{2} + 2p\epsilon^2.$$

Therefore, Proposition 2.4 holds. □

If, by contrast, **E** is a randomly chosen function, then the expected probability that $\delta \odot \mathbf{E}_K(P') \oplus \delta \odot \mathbf{E}_K(P^*) = \gamma \odot \beta$ is $\frac{1}{2}$. Therefore, if the bias $2p\epsilon^2$ is sufficiently large, we can distinguish the block cipher from a randomly chosen function, given a sufficient number of matching plaintext/ciphertext pairs.

2.2.7 Impossible Differential Cryptanalysis

Impossible differential cryptanalysis was independently introduced by Knudsen [57] in 1998 and Biham, Biryukov and Shamir [4] in 1999.

An impossible differential is a differential with a probability of zero. Such a differential is typically constructed in a miss-in-the-middle manner [5]; that is, a differential with probability 1 is concatenated with another differential with probability 1, where the intermediate differences of the two differentials contradict one another.

Impossible differential cryptanalysis uses one or more impossible differentials, written $\Delta\alpha \nrightarrow \Delta\beta$, and it usually treats a block cipher $\mathbf{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$

as a cascade of three sub-ciphers $\mathbf{E} = \mathbf{E}^a \circ \mathbf{E}^0 \circ \mathbf{E}^b$, where $\mathbf{E}^0$ denotes the rounds for which $\alpha \nrightarrow \beta$ holds, $\mathbf{E}^a$ denotes a number of rounds before $\mathbf{E}^0$, and $\mathbf{E}^b$ denotes a number of rounds after $\mathbf{E}^0$.

Given a guess for the subkeys used in $\mathbf{E}^a$ and $\mathbf{E}^b$, if a plaintext pair produces a difference of α just after $\mathbf{E}^a$, and its corresponding ciphertext pair produces a difference of β just before $\mathbf{E}^b$, then this guess for the subkeys must be incorrect. Thus, given a sufficient number of matching plaintext/ciphertext pairs, an attacker can find the correct subkey by discarding the wrong guesses.

2.2.8 Boomerang and Rectangle Attacks

The boomerang attack was introduced in 1999 by Wagner [103]. Such an attack uses two differentials on two different parts of the cipher, instead of a single differential on the entire cipher.

A boomerang attack uses something called a boomerang distinguisher. To define a boomerang distinguisher we need to treat a block cipher $\mathbf{E} : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ as a cascade of two sub-ciphers $\mathbf{E}^0$ and $\mathbf{E}^1$, where $\mathbf{E} = \mathbf{E}^0 \circ \mathbf{E}^1$. Suppose $K \in \{0,1\}^k$ is a key for $\mathbf{E}$. A boomerang distinguisher is then defined to be a pair of differentials $(\Delta\alpha \rightarrow \Delta\beta, \Delta\gamma \rightarrow \Delta\delta)$, where $\Delta\alpha \rightarrow \Delta\beta$ is a differential for $\mathbf{E}^0_K$ with probability p, $\Delta\gamma \rightarrow \Delta\delta$ is a differential for $\mathbf{E}^1_K$ with probability q, and $p \cdot q > 2^{-\frac{n}{2}}$.

Suppose we choose N pairs of plaintext blocks (P, P^*) where $P^* = P \oplus \alpha$. We denote respectively by C and C^* the ciphertext blocks for the plaintext blocks P and P^* encrypted using the block cipher $\mathbf{E}$ under key K.

Then, if we apply $\mathbf{E}^0_K$ to each of these pairs, we will obtain approximately Np pairs $(\mathbf{E}^0_K(P), \mathbf{E}^0_K(P^*))$ with the property that $\mathbf{E}^0_K(P) \oplus \mathbf{E}^0_K(P^*) = \beta$.

Next, we choose N pairs of ciphertext blocks $(C' = C \oplus \delta, C'^* = C^* \oplus \delta)$. If we apply $(\mathbf{E}^1_K)^{-1}$ to each of the pairs (C, C'), we get that $(\mathbf{E}^1_K)^{-1}(C) \oplus (\mathbf{E}^1_K)^{-1}(C') = \gamma$ with probability q; if we apply $(\mathbf{E}^1_K)^{-1}$ to each of the pairs (C^*, C'^*), we get that $(\mathbf{E}^1_K)^{-1}(C^*) \oplus (\mathbf{E}^1_K)^{-1}(C'^*) = \gamma$ with probability q. Therefore, we will ob-

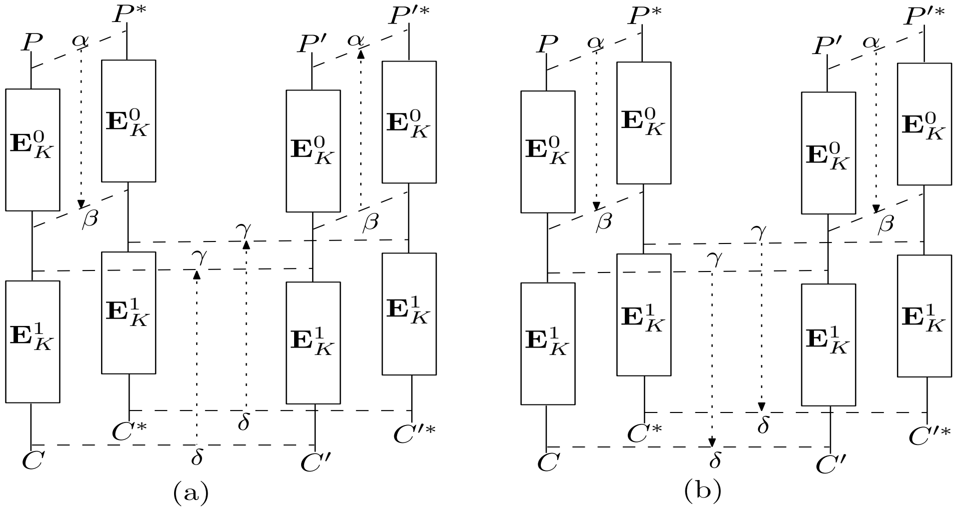


Figure 2.1: The boomerang and amplified boomerang distinguishers

tain approximately Npq^2 pairs $((\mathbf{E}^1_K)^{-1}(C'), (\mathbf{E}^1_K)^{-1}(C'^*))$ with the property that $(\mathbf{E}^1_K)^{-1}(C') \oplus (\mathbf{E}^1_K)^{-1}(C'^*) = \beta$. This is because

$$\begin{aligned}
&(\mathbf{E}^1_K)^{-1}(C') \oplus (\mathbf{E}^1_K)^{-1}(C'^*)\\
&= (\mathbf{E}^1_K)^{-1}(C) \oplus (\mathbf{E}^1_K)^{-1}(C^*) \oplus (\mathbf{E}^1_K)^{-1}(C) \oplus (\mathbf{E}^1_K)^{-1}(C') \oplus (\mathbf{E}^1_K)^{-1}(C^*) \oplus\\
&\quad (\mathbf{E}^1_K)^{-1}(C'^*)\\
&= \beta \oplus \gamma \oplus \gamma\\
&= \beta.
\end{aligned}$$

Therefore, we will get Np^2q^2 pairs of plaintext blocks $((\mathbf{E}_K)^{-1}(C'), (\mathbf{E}_K)^{-1}(C'^*))$ with the property that $(\mathbf{E}_K)^{-1}(C') \oplus (\mathbf{E}_K)^{-1}(C'^*) = \alpha$. Figure 2.1(a) depicts the boomerang distinguisher.

However, for a randomly chosen function, the expected number of plaintext pairs (P', P'^*) with the property that $P' \oplus P'^* = \alpha$ is approximately $N \cdot 2^{-n}$.

Therefore, if $p \cdot q > 2^{-\frac{n}{2}}$, the boomerang distinguisher can effectively distinguish between $\mathbf{E}$ and a randomly chosen function, given a sufficient number of adaptive chosen plaintexts and ciphertexts.

In 2000, Kelsey, Kohno and Schneier [48] presented a variant of the boomerang attack, known as the amplified boomerang attack.

An amplified boomerang attack uses something called an amplified boomerang distinguisher. To define an amplified boomerang distinguisher we also need to treat a block cipher $\mathbf{E}: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ as a cascade of two sub-ciphers $\mathbf{E}^0$ and $\mathbf{E}^1$, where $\mathbf{E} = \mathbf{E}^0 \circ \mathbf{E}^1$. Suppose $K \in \{0,1\}^k$ is a key for $\mathbf{E}$. An amplified boomerang distinguisher is then defined to be a pair of differentials $(\Delta\alpha \rightarrow \Delta\beta, \Delta\gamma \rightarrow \Delta\delta)$, where $\Delta\alpha \rightarrow \Delta\beta$ is a differential for $\mathbf{E}^0_K$ with probability p, $\Delta\gamma \rightarrow \Delta\delta$ is a differential for $\mathbf{E}^1_K$ with probability q, and $p \cdot q > 2^{-\frac{n}{2}}$.

A right quartet consists of two pairs of plaintext blocks $(P, P^* = P \oplus \alpha)$ and $(P', P'^* = P' \oplus \alpha)$ satisfying the following three conditions; see Figure 2.1(b).

C1: $\mathbf{E}^0_K(P) \oplus \mathbf{E}^0_K(P^*) = \mathbf{E}^0_K(P') \oplus \mathbf{E}^0_K(P'^*) = \beta$;

C2: $\mathbf{E}^0_K(P) \oplus \mathbf{E}^0_K(P') = \mathbf{E}^0_K(P^*) \oplus \mathbf{E}^0_K(P'^*) = \gamma$;

C3: $\mathbf{E}_K(P) \oplus \mathbf{E}_K(P') = \mathbf{E}_K(P^*) \oplus \mathbf{E}_K(P'^*) = \delta$.

Suppose we choose N pairs of plaintext blocks (P, P^*) where $P^* = P \oplus \alpha$. These pairs yield $\binom{N}{2} = \frac{N(N-1)}{2}$ candidate quartets $((P, P^*), (P', P'^*))$, where $(P', P'^*) \neq (P, P^*) \in \{(P, P^*)\}$.

Then, if we apply $\mathbf{E}^0_K$ to each of these quartets, we will obtain approximately Np^2 quartets $((P, P^*), (P', P'^*))$ with the property that $\mathbf{E}^0_K(P) \oplus \mathbf{E}^0_K(P^*) = \mathbf{E}^0_K(P') \oplus \mathbf{E}^0_K(P'^*) = \beta$.

Assuming that the intermediate values after $\mathbf{E}^0_K$ are distributed uniformly over all possible values, we get $\mathbf{E}^0_K(P) \oplus \mathbf{E}^0_K(P') = \gamma$ with probability 2^{-n}. Once this occurs, $\mathbf{E}^0_K(P^*) \oplus \mathbf{E}^0_K(P'^*) = \gamma$ holds as well, as

$$\begin{aligned}
&\mathbf{E}^0_K(P^*) \oplus \mathbf{E}^0_K(P'^*) \\
&= \mathbf{E}^0_K(P) \oplus \mathbf{E}^0_K(P^*) \oplus \mathbf{E}^0_K(P') \oplus \mathbf{E}^0_K(P'^*) \oplus \mathbf{E}^0_K(P) \oplus \mathbf{E}^0_K(P') \\
&= \gamma.
\end{aligned}$$

Therefore, the expected number of candidate quartets $((P, P^*), (P', P'^*))$ with the property that $\mathbf{E}_K(P) \oplus \mathbf{E}_K(P') = \mathbf{E}_K(P^*) \oplus \mathbf{E}^0_K(P'^*) = \delta$ is approximately

$$\frac{N(N-1)}{2} \cdot 2^{-n} \cdot p^2 \cdot q^2.$$

However, for a randomly chosen function, the expected number of candidate quartets $((P, P^*), (P', P'^*))$ with the property that $\mathbf{E}_K(P) \oplus \mathbf{E}_K(P') = \mathbf{E}_K(P^*) \oplus \mathbf{E}^0_K(P'^*) = \delta$ is approximately $\frac{N(N-1)}{2} \cdot 2^{-2n}$.

Therefore, if $p \cdot q > 2^{-\frac{n}{2}}$, the amplified boomerang distinguisher can effectively distinguish between $\mathbf{E}$ and a randomly chosen function, given a sufficient number of chosen plaintexts.

In 2001, Biham, Dunkelman and Keller [6] presented an improvement of the amplified boomerang attack, known as the rectangle attack.

The rectangle attack improves over an amplified boomerang attack by allowing β to take any possible value β' in $\mathbf{E}^0_K$ and γ to take any possible value γ' in $\mathbf{E}^1_K$, as long as $\beta' \neq \gamma'$. As a result, given the same number of plaintext pairs as described in the above amplified boomerang attack, the expected number of candidate quartets $((P, P^*), (P', P'^*))$ with the property that $\mathbf{E}_K(P) \oplus \mathbf{E}_K(P') = \mathbf{E}_K(P^*) \oplus \mathbf{E}^0_K(P'^*) = \delta$ is approximately

$$\frac{N(N-1)}{2} \cdot (\widehat{p} \cdot \widehat{q})^2 \cdot 2^{-n},$$

where $\widehat{p} = (\sum_{\beta'} \Pr^2_{\mathbf{E}^0_K}(\Delta\alpha \to \Delta\beta'))^{\frac{1}{2}}$ and $\widehat{q} = (\sum_{\gamma'} \Pr^2_{\mathbf{E}^1_K}(\Delta\gamma' \to \Delta\delta))^{\frac{1}{2}}$.

Other extensions to the boomerang attack include the differential-linear boomerang attack [8] and the differential-bilinear boomerang attack [8].

2.2.9 Related-Key Cryptanalysis

Related-key cryptanalysis was independently introduced by Knudsen [55] in 1992 and Biham [3] in 1993.

Related-key cryptanalysis takes advantage of how a specific difference in a pair of inputs of a cipher or function can affect a difference in the pair of outputs of the cipher or function, where the pair of outputs are obtained by encrypting the pair of inputs using two different keys with a specific difference. The notion of difference can be defined in several ways; the most widely discussed is with respect to the XOR operation. The difference between the inputs is called the input difference, the difference between the outputs of a function is called the output difference, the difference between internal values is called an intermediate difference, and the difference between the user keys is called the user key difference. If we denote by K, K' the two related keys, then the combination of the input difference and the output difference is called a related-key differential under keys K and K'.

Related-key cryptanalysis assumes that the attacker knows or can choose the key difference. This assumption means that it is difficult or even infeasible to conduct such an attack in many applications. Anyway, as demonstrated in [49, 50], certain current real-world applications may allow for practical related-key attacks, including key-exchange protocols and hash functions.

The probability of a related-key differential under keys K and K', written $\Delta\alpha \rightarrow \Delta\beta$, is defined as the probability that the input difference propagates to the output difference under K and K'; more formally, it is defined as follows.

Definition 2.7 *Suppose* $\mathbf{E}$ *is a block cipher and* $K, K' \in \{0,1\}^k$ *are keys for the cipher. If* α *and* β *are* n*-bit blocks, then the probability of the related-key differential for the pair* (α, β) *under the related keys* K *and* K'*, written* $\Delta\alpha \rightarrow \Delta\beta$*, is defined to be*

$$\mathrm{Pr}_{\mathbf{E}_K,\mathbf{E}_{K'}}(\Delta\alpha \rightarrow \Delta\beta) = \Pr_{P\in\{0,1\}^n}(\mathbf{E}_K(P) \oplus \mathbf{E}_{K'}(P \oplus \alpha) = \beta).$$

The following result follows trivially from Definition 2.7:

Proposition 2.5 *If* $\mathbf{E}$ *is an* n*-bit block cipher,* $K, K' \in \{0,1\}^k$ *are keys for the cipher, and* α *and* β *are* n*-bit blocks, then*

$$\mathrm{Pr}_{\mathbf{E}_K,\mathbf{E}_{K'}}(\Delta\alpha \rightarrow \Delta\beta) = \frac{|\{x|\mathbf{E}_K(x) \oplus \mathbf{E}_{K'}(x \oplus \alpha) = \beta, x \in \{0,1\}^n\}|}{2^n}.$$

Sometimes we refer to the related-key differential of a block cipher without specifying the related keys. For most currently studied block ciphers the differential probabilities do not depend on the keys used, and so this is reasonable practice.

There may be a number of different intermediate related-key differences that give rise to the same related-key differential. A sequence of intermediate related-key differences that give rise to a particular related-key differential is called a related-key differential characteristic. That is, a related-key differential is the set of all the related-key differential characteristics with the same input difference and output difference under the same related keys.

A related-key differential (characteristic) for r consecutive rounds is often called an r-round related-key differential (characteristic). An r-round related-key differential (characteristic) that has a probability of p is often called an r-round related-key differential (characteristic) with probability p.

In the following, we briefly describe the related-key rectangle attack, which is a combination of the related-key arrack and the rectangle attack.

The related-key rectangle attack [9, 40, 53] uses something called a related-key rectangle distinguisher. Like a rectangle distinguisher, a related-key rectangle distinguisher treats a block cipher $\mathbf{E} : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ as a cascade of two sub-ciphers $\mathbf{E}^0$ and $\mathbf{E}^1$, where $\mathbf{E} = \mathbf{E}^0 \circ \mathbf{E}^1$. Typically, such a related-key rectangle distinguisher works in a related-key attack scenario involving four related keys K_A, K_B, K_C, K_D satisfying $K_A \oplus K_B = K_C \oplus K_D = \Delta K_0$ and $K_A \oplus K_C = K_B \oplus K_D = \Delta K_1$, where ΔK_0 and ΔK_1 are two known differences, and is made up of four groups of related-key differentials:

- all the possible related-key differentials $\Delta\alpha \to \Delta\beta$ for $\mathbf{E}^0$ under related keys K_A and K_B, where β is any possible output difference;
- all the possible related-key differentials $\Delta\alpha \to \Delta\beta$ for $\mathbf{E}^0$ under related keys K_C and K_D, where β is any possible output difference;
- all the possible related-key differentials $\Delta\gamma \to \Delta\delta$ for $\mathbf{E}^1$ under related keys K_A and K_C, where γ is any possible input difference;
- all the possible related-key differentials $\Delta\gamma \to \Delta\delta$ for $\mathbf{E}^1$ under related keys K_B and K_D, where γ is any possible input difference.

A right quartet consists of two pairs of plaintexts $(P, P^* = P \oplus \alpha)$ and $(P', P'^* =$

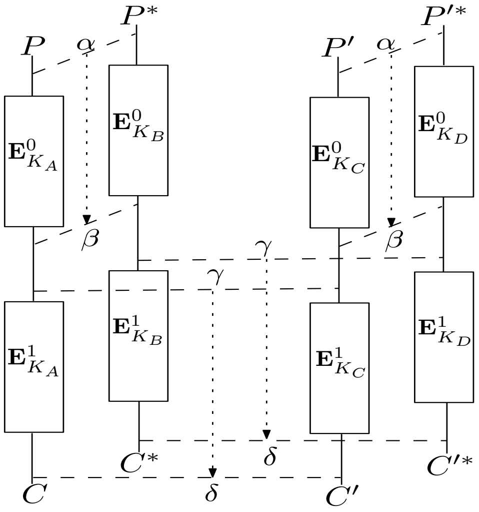


Figure 2.2: A related-key rectangle distinguisher

$P' \oplus \alpha$) satisfying the following three conditions; see Figure 2.2.

C1: $\mathbf{E}^0_{K_A}(P) \oplus \mathbf{E}^0_{K_B}(P^*) = \mathbf{E}^0_{K_C}(P') \oplus \mathbf{E}^0_{K_D}(P'^*) = \beta$,

C2: $\mathbf{E}^0_{K_A}(P) \oplus \mathbf{E}^0_{K_C}(P') = \mathbf{E}^0_{K_B}(P^*) \oplus \mathbf{E}^0_{K_D}(P'^*) = \gamma$,

C3: $\mathbf{E}_{K_A}(P) \oplus \mathbf{E}_{K_C}(P') = \mathbf{E}_{K_B}(P^*) \oplus \mathbf{E}_{K_D}(P'^*) = \delta$.

Assuming that the intermediate values after $\mathbf{E}^0$ are distributed uniformly over all possible values, then we can get $\mathbf{E}^0_{K_A}(P) \oplus \mathbf{E}^0_{K_C}(P') = \gamma$ with probability 2^{-n}. Once this occurs, by C1 we know that $\mathbf{E}^0_{K_B}(P^*) \oplus \mathbf{E}^0_{K_D}(P'^*) = \gamma$ holds with probability 1, for

$$\begin{aligned}
&\mathbf{E}^0_{K_B}(P^*) \oplus \mathbf{E}^0_{K_D}(P'^*) \\
&= (\mathbf{E}^0_{K_A}(P) \oplus \mathbf{E}^0_{K_B}(P^*)) \oplus (\mathbf{E}^0_{K_C}(P') \oplus \mathbf{E}^0_{K_D}(P'^*)) \oplus (\mathbf{E}^0_{K_A}(P) \oplus \mathbf{E}^0_{K_C}(P')) \\
&= \beta \oplus \beta \oplus \gamma \\
&= \gamma.
\end{aligned}$$

As a result, the probability that the quartet satisfies C3 is expected to be approximately

$$\sum_{\beta,\gamma} (\Pr_{\mathbf{E}^0_{K_A},\mathbf{E}^0_{K_B}}(\Delta\alpha \to \Delta\beta))^2 \cdot 2^{-n} \cdot (\Pr_{\mathbf{E}^1_{K_A},\mathbf{E}^1_{K_C}}(\Delta\gamma \to \Delta\delta))^2 = 2^{-n} \cdot (\widehat{p} \cdot \widehat{q})^2,$$

where $\widehat{p} = (\sum_{\beta'} \Pr^2_{\mathbf{E}^0_{K_A},\mathbf{E}^0_{K_B}}(\Delta\alpha \to \Delta\beta'))^{\frac{1}{2}}$ and $\widehat{q} = (\sum_{\gamma'} \Pr^2_{\mathbf{E}^1_{K_A},\mathbf{E}^1_{K_C}}(\Delta\gamma' \to \Delta\delta))^{\frac{1}{2}}$.

For a random function, the probability that the quartet satisfies C3 is approximately $2^{-n\times 2} = 2^{-2n}$.

Therefore, if $\widehat{p} \cdot \widehat{q} > 2^{-\frac{n}{2}}$, the related-key rectangle distinguisher can distinguish between $\mathbf{E}$ and a random function given a sufficient number of chosen plaintext pairs.

Note that there exist three types of related-key rectangle attacks, which correspond to the following three cases.

- TYPE 1: $\Delta K_0 \neq 0, \Delta K_1 \neq 0$, (four keys);
- TYPE 2: $\Delta K_0 = 0, \Delta K_1 \neq 0$, (two keys);
- TYPE 3: $\Delta K_0 \neq 0, \Delta K_1 = 0$, (two keys).

2.3 Summary

In this chapter we have briefly reviewed a number of cryptanalytic methods for block ciphers. In subsequent chapters we use and extend these techniques to obtain new cryptanalytic results for a range of block ciphers.

It is very worthy to note that the statistical methods described above generally treat a basic unit of input (i.e. a chosen-plaintext pair for differential cryptanalysis, differential-linear cryptanalysis, and impossible differential cryptanalysis; a known-plaintext for linear cryptanalysis; and a quartet of (adaptive) chosen plaintexts for boomerang and (related-key) rectangle attacks) as a Bernoulli random variable, and assume that given a set of inputs of the basic unit, the inputs that satisfy the required property have (or can be approximated by) a binomial distribution.

The methods we consider here are all statistical in nature; they typically require assuming that the output of one intermediate round is uniformly distributed, and is independent from that of previous rounds. As Handschuh and Naccache [31] mention, this is “most often not exactly the case, but as often it is a good approximation”. This means that, in some cases, the success probability of the attack may be overestimated. However, in the absence of any evidence one way or the other, it seems reasonable to take the worst case assumption from the point of the user of the cipher. As a result we make use of assumptions regarding uniform distributions at various places in this thesis.

Other cryptanalytic methods not considered here include integral cryptanalysis [20, 60, 82] and algebraic cryptanalysis [18]. These techniques are different in nature from those described above.

Chapter 3

The Impossible Boomerang Attack

In this chapter we propose a new extension of differential cryptanalysis, named the impossible boomerang attack. We also describe a variant of this attack which applies in a related-key attack scenario.

Contents

3.1 Introduction

Most modern block ciphers are designed to be provably secure against differential cryptanalysis and linear cryptanalysis [94, 95]. Thus proposing new cryptanalytic techniques is always desirable in the sense that it provides a better evaluation of the security of a block cipher and also enables more secure ciphers to be designed.

Impossible differential cryptanalysis and the boomerang-type attacks (including the boomerang, amplified boomerang and rectangle attacks as well as their related-key variants) have been used to yield the best currently published cryptanalytic results for a number of state-of-the-art block ciphers [2, 9, 52, 64, 76, 111]. These techniques are thus clearly of importance.

In this chapter, inspired by the ideas that impossible differential cryptanalysis and the boomerang attack use, we propose a new extension of differential cryptanalysis, which we call the impossible boomerang attack. Such an attack is based on the use of a so-called impossible boomerang distinguisher, which, like a boomerang attack, treats a block cipher $\mathbf{E}$ as two sub-ciphers $\mathbf{E}^0 \circ \mathbf{E}^1$. It uses two (or more) differentials with probability 1 for $\mathbf{E}^0$ and two (or more) differentials with probability 1 for $\mathbf{E}^1$, where the XOR of the intermediate differences of these differentials is not equal to zero. We then describe a variant of this attack that applies in a related-key scenario, giving rise to what we call a related-key impossible boomerang attack.

The rest of this chapter is organised as follows. In Section 3.2 we propose the impossible boomerang attack. In Section 3.3 we briefly describe the variant of the impossible boomerang attack in a related-key attack scenario. In Section 3.4 we compare the impossible boomerang attack with impossible differential cryptanalysis and the boomerang-type attacks. Section 3.5 summarises this chapter.

3.2 The Impossible Boomerang Attack

Typically, when formulating a differential cryptanalysis attack, it is desirable to use a differential operating on as many rounds of the cipher as possible. Of course, the more rounds the differential operates on, the smaller its probability is likely to be. As described in Section 3.2.1, the boomerang attack is based on a somewhat different idea, namely of using two short differentials with relatively large probabilities, instead of using a differential operating on as many rounds as possible with a small probability. Impossible differential cryptanalysis involves using a differential that will never happen under any situation. The attack we describe in this chapter, i.e. what we call the impossible boomerang attack, combines the boomerang attack with impossible differential cryptanalysis. Possible combinations of cryptanalytic techniques have been proposed in the past, and have proved effective [8, 9, 34, 40, 53, 67]; a good example is provided by differential-linear cryptanalysis [7, 67].

3.2.1 The Basic Impossible Boomerang Attack

As mentioned earlier, an impossible boomerang attack is constructed on an impossible boomerang distinguisher.

3.2.1.1 Distinguisher Using Two Tuples

An impossible boomerang distinguisher is defined as follows. Like a boomerang distinguisher, an impossible boomerang distinguisher treats a block cipher $\mathbf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ as two sub-ciphers $\mathbf{E}^0 \circ \mathbf{E}^1$. Such a distinguisher is made up of four related differentials (or truncated differentials [56]), two for $\mathbf{E}^0$ and two for $(\mathbf{E}^1)^{-1}$, all of which must have probability 1. That is, an impossible boomerang distinguisher consists of:

- a differential $\Delta\alpha \to \Delta\beta$ with probability 1 for $\mathbf{E}^0$;
- a differential $\Delta\alpha' \to \Delta\beta'$ with probability 1 for $\mathbf{E}^0$;
- a differential $\Delta\delta \to \Delta\gamma$ with probability 1 for $(\mathbf{E}^1)^{-1}$;
- a differential $\Delta\delta' \to \Delta\gamma'$ with probability 1 for $(\mathbf{E}^1)^{-1}$,

where $\alpha, \alpha', \beta, \beta', \gamma, \gamma', \delta$ and δ' are all n-bit blocks, and β, β', γ and γ' meet the condition $\beta \oplus \beta' \oplus \gamma \oplus \gamma' \neq 0$. An impossible boomerang distinguisher is shown pictorially in Figure 3.1(a).

The following theorem provides the theoretical basis for the impossible boomerang attack.

Theorem 3.1 *Suppose that X and X' are n-bit blocks and K is a key for an n-bit block cipher $\mathbf{E}$, where $\mathbf{E} = \mathbf{E}^0 \circ \mathbf{E}^1$ for some $\mathbf{E}^0$ and $\mathbf{E}^1$. Suppose that $\Delta\alpha \to \Delta\beta$ and $\Delta\alpha' \to \Delta\beta'$ are differentials with probability 1 for $\mathbf{E}^0_K$, and $\Delta\delta \to \Delta\gamma$ and $\Delta\delta' \to \Delta\gamma'$ are differentials with probability 1 for $(\mathbf{E}^1_K)^{-1}$, where $\beta \oplus \beta' \oplus \gamma \oplus \gamma' \neq 0$.*

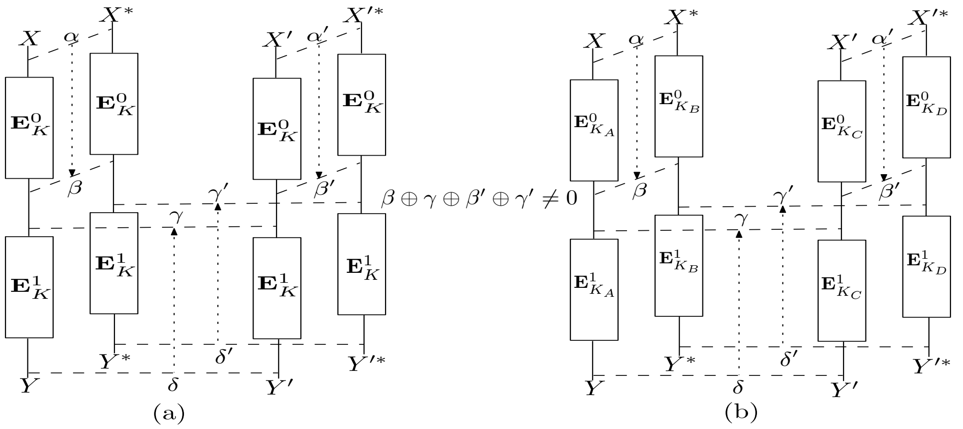


Figure 3.1: Impossible boomerang and related-key impossible boomerang distinguishers

Then the following pair of equations cannot both hold:

$$\mathbf{E}_K(X) \oplus \mathbf{E}_K(X') = \delta, \tag{3.1}$$

$$\mathbf{E}_K(X \oplus \alpha) \oplus \mathbf{E}_K(X' \oplus \alpha) = \delta'. \tag{3.2}$$

Proof. Suppose that equations (3.1) and (3.2) both hold for some X, X' and K. Since both the differentials $\Delta\alpha \to \Delta\beta$ and $\Delta\alpha' \to \Delta\beta'$ for $\mathbf{E}^0_K$ hold with probability 1, we have

$$\begin{aligned} \mathbf{E}^0_K(X) \oplus \mathbf{E}^0_K(X \oplus \alpha) &= \beta, \\ \mathbf{E}^0_K(X') \oplus \mathbf{E}^0_K(X' \oplus \alpha') &= \beta'. \end{aligned}$$

As both the differentials $\Delta\delta' \to \Delta\gamma'$ and $\Delta\delta \to \Delta\gamma$ for $(\mathbf{E}^1_K)^{-1}$ hold with probability 1, we can get the following equation with probability 1:

$$\begin{aligned} &\mathbf{E}^0_K(X') \oplus \mathbf{E}^0_K(X' \oplus \alpha) \\ &= (\mathbf{E}^0_K(X') \oplus \mathbf{E}^0_K(X)) \oplus (\mathbf{E}^0_K(X) \oplus \mathbf{E}^0_K(X \oplus \alpha)) \oplus (\mathbf{E}^0_K(X \oplus \alpha) \oplus \mathbf{E}^0_K(X' \oplus \alpha)) \\ &= ((\mathbf{E}^1_K)^{-1}(\mathbf{E}_K(X')) \oplus (\mathbf{E}^1_K)^{-1}(\mathbf{E}_K(X))) \oplus (\mathbf{E}^0_K(X) \oplus \mathbf{E}^0_K(X \oplus \alpha)) \oplus \\ &\quad ((\mathbf{E}^1_K)^{-1}(\mathbf{E}_K(X \oplus \alpha)) \oplus (\mathbf{E}^1_K)^{-1}(\mathbf{E}_K(X' \oplus \alpha))) \\ &= \gamma \oplus \beta \oplus \gamma'. \end{aligned}$$

Hence, from the above discussion we get that $\mathbf{E}^0_K(X')\oplus\mathbf{E}^0_K(X'\oplus\alpha) = \beta' = \gamma\oplus\beta\oplus\gamma'$ holds. However, this contradicts with the condition that $\beta \oplus \beta' \oplus \gamma \oplus \gamma' \neq 0$. Therefore, Theorem 3.1 holds. □

From Theorem 3.1 we know that a distinguisher of the form shown in Figure 3.1(a) can never occur; we call it an impossible boomerang distinguisher, written $(\Delta\alpha, \Delta\alpha') \nrightarrow (\Delta\delta, \Delta\delta')$.

Note that the two differentials for $\mathbf{E}^0$ or $\mathbf{E}^1$ may be identical, as long as the condition $\beta \oplus \beta' \oplus \gamma \oplus \gamma' \neq 0$ holds.

3.2.1.2 A Key Recovery Attack

An impossible boomerang attack involves treating a block cipher $\mathbf{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ as a cascade of four sub-ciphers $\mathbf{E} = \mathbf{E}^a \circ \mathbf{E}^0 \circ \mathbf{E}^1 \circ \mathbf{E}^b$, where $\mathbf{E}^0 \circ \mathbf{E}^1$ denotes the rounds for which the impossible boomerang distinguisher $(\Delta\alpha, \Delta\alpha') \nrightarrow (\Delta\delta, \Delta\delta')$ holds, $\mathbf{E}^a$ denotes a number of rounds before $\mathbf{E}^0$, and $\mathbf{E}^b$ denotes a number of rounds after $\mathbf{E}^1$.

In a chosen plaintext attack scenario, given a guess for the subkeys used in $\mathbf{E}^a$ and $\mathbf{E}^b$, the impossible boomerang attack involves checking whether a candidate quartet consisting of two pairs of plaintext blocks meets the differential conditions required by the impossible boomerang distinguisher. Specifically, suppose K_a is the guess for the subkey used in $\mathbf{E}^a$, and K_b is the guess for the subkey used in $\mathbf{E}^b$, then the attacker checks whether a candidate quartet of known plaintext/ciphertext pairs $(((P,C),(P^*,C^*)),((P',C'),(P',C'^*)))$ satisfies the following four conditions:

$$\mathbf{E}^a_{K_a}(P) \oplus \mathbf{E}^a_{K_a}(P^*) = \alpha, \tag{3.3}$$

$$\mathbf{E}^a_{K_a}(P') \oplus \mathbf{E}^a_{K_a}(P'^*) = \alpha', \tag{3.4}$$

$$(\mathbf{E}^b_{K_b})^{-1}(C) \oplus (\mathbf{E}^b_{K_b})^{-1}(C') = \delta, \tag{3.5}$$

$$(\mathbf{E}^b_{K_b})^{-1}(C^*) \oplus (\mathbf{E}^b_{K_b})^{-1}(C'^*) = \delta'. \tag{3.6}$$

If there exists a candidate quartet satisfying equations (3.3)–(3.6), then the subkey guess (K_a, K_b) must be incorrect, and can be discarded. Thus, given a sufficient number of chosen plaintext pairs, the attacker can find the correct subkeys used in $\mathbf{E}^a$ and $\mathbf{E}^b$ by discarding the wrong guesses.

3.2.2 The Impossible Boomerang Attack Using More Tuples

The impossible boomerang distinguisher described above uses two tuples, i.e. $(X, X^* = X \oplus \alpha)$ and $(X', X'^* = X' \oplus \alpha')$. In fact, we can construct an impossible boomerang distinguisher using more tuples.

For example, suppose we have a third tuple $(X'', X''^* = X'' \oplus \alpha'')$, and we have two additional differentials $\Delta\alpha'' \to \Delta\beta''$ and $\Delta\delta'' \to \Delta\gamma''$ for $\mathbf{E}^0$ and $\mathbf{E}^1$, respectively, both with probability 1. Suppose also that $\beta \oplus \beta' \oplus \beta'' \oplus \gamma \oplus \gamma' \oplus \gamma'' \neq 0$. Then we can construct a 6-fold impossible boomerang distinguisher, which can be used to construct an attack, given a sufficient number of plaintext pairs.

3.3 The Related-Key Impossible Boomerang Attack

In a related-key attack scenario [3, 49, 55], the attacker is assumed to know the specific differences between one or more pairs of unknown keys.

A related-key impossible boomerang distinguisher treats a block cipher $\mathbf{E} : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ as two sub-ciphers $\mathbf{E}^0 \circ \mathbf{E}^1$. Typically, such a distinguisher works in a related-key attack scenario involving four related keys K_A, K_B, K_C, K_D, and is made up of four related-key differentials, two for $\mathbf{E}^0$ and two for $(\mathbf{E}^1)^{-1}$, all of which must have probability 1. That is, a related-key impossible boomerang distinguisher consists of:

- a related-key differential $\Delta\alpha \to \Delta\beta$ with probability 1 for $\mathbf{E}^0$ under keys K_A and K_B;
- a related-key differential $\Delta\alpha' \to \Delta\beta'$ with probability 1 for $\mathbf{E}^0$ under keys K_C and K_D;

- a related-key differential $\Delta\delta \to \Delta\gamma$ with probability 1 for $(\mathbf{E}^1)^{-1}$ under keys K_A and K_C;
- a related-key differential $\Delta\delta' \to \Delta\gamma'$ with probability 1 for $(\mathbf{E}^1)^{-1}$ under keys K_B and K_D,

where $\alpha, \alpha', \beta, \beta', \gamma, \gamma', \delta$ and δ' are all n-bit blocks, and β, β', γ and γ' meet the condition $\beta \oplus \beta' \oplus \gamma \oplus \gamma' \neq 0$. A related-key impossible boomerang distinguisher is depicted in Figure 3.1(b).

Similarly we can learn that such a related-key distinguisher is impossible and allows us to conduct a related-key impossible boomerang attack given a sufficient number of chosen plaintext pairs.

3.4 A Comparison

From an impossible boomerang distinguisher we can always obtain an impossible differential for the same number of rounds. Consider an impossible boomerang distinguisher using two tuples; from the condition $\beta \oplus \beta' \oplus \gamma \oplus \gamma' \neq 0$ we have $\beta \oplus \gamma \neq \beta' \oplus \gamma'$, which implies that the values $\beta \oplus \gamma$ and $\beta' \oplus \gamma'$ cannot both be equal to zero. The above result applies when using two tuples, since the four differentials required by the impossible boomerang distinguisher have a probability of one. A similar result holds when using more tuples.

However, this relationship does not hold for their variants in a related-key attack scenario. When formulating a related-key impossible differential, choosing the subkey difference for $\mathbf{E}^0$ usually incurs a fixed subkey difference for $\mathbf{E}^1$, and vice versa; but when formulating a related-key impossible boomerang distinguisher we have more flexibility in choosing the subkey differences for $\mathbf{E}^0$ and $\mathbf{E}^1$: we can use a subkey difference for $\mathbf{E}^0$ and use a completely irrelevant subkey difference for $\mathbf{E}^1$, and even more flexibly, we can use two different subkey differences for $\mathbf{E}^0$ or $\mathbf{E}^1$. These flexibilities in choosing the key differences may enable us to break more rounds of a block cipher using a related-key impossible boomerang attack.

The (related-key) impossible boomerang attack gives us different choices on the (related-key) differentials used as well as the plaintexts required, and can be treated as multi-dimensional (related-key) impossible differential cryptanalysis.

The advantages of the (related-key) impossible boomerang attack over the boomerang-type attacks are analogous to those of (related-key) impossible differential cryptanalysis over (related-key) differential cryptanalysis. A block cipher resistant to boomerang-type attacks will not necessarily resist a (related-key) impossible boomerang attack. In boomerang-type distinguishers, one generally assumes that the output of one intermediate round of the cipher is uniformly distributed and is independent from that of previous rounds, while an impossible boomerang distinguisher does not require this assumption, which is often observed to be not the truth [105]. Therefore, a (related-key) impossible boomerang distinguisher is more reasonable than boomerang-type distinguishers.

3.5 Summary

In this chapter, inspired by the notions of impossible differential cryptanalysis and the boomerang attack, we have proposed a new extension of differential cryptanalysis, called the impossible boomerang attack. We have also described a variant of this attack that applies in a related-key attack scenario.

In Chapter 5 we have applied the impossible boomerang attack to break 6-round AES-128, 7-round AES-192 and 7-round AES-256 in a single key attack scenario, and 8-round AES-192 and 9-round AES-256 in a related-key attack scenario involving two keys.

The (related-key) impossible boomerang attack is a general cryptanalytic technique and can potentially be used to cryptanalyse other block ciphers. It is likely to be particularly useful in cryptanalysing ciphers with a simple key schedule in a related-key attack scenario.

Chapter 4

The Early Abort Technique

In this chapter we give a general description of early abort techniques for (related-key) impossible differential cryptanalysis and rectangle attacks. In some circumstances these techniques can be used to improve the efficiency of such attacks.

Contents

4.1 Introduction

Certain cryptanalytic techniques for block ciphers involve exhaustive searches over one or more subkeys (some of the bits of which may already be known). The means used to eliminate possibilities for the subkeys are dependent on the cryptanalysis method. Nevertheless, we can identify a general approach, which we call the early abort technique, which applies to more than one type of cryptanalysis.

This technique involves taking advantage of special properties of the block cipher round function. In some cases the structure of the round function itself, when combined with the particular approach to eliminating subkey possibilities, enables the subkey search to be partitioned, so that some subkey bits can be tested independently of the values of other subkey bits. This has the potential to significantly reduce the size of the exhaustive search.

In this chapter we consider in detail two examples of the early abort technique as applied to specific cryptanalytic techniques — (related-key) impossible differential cryptanalysis and (related-key) rectangle attacks.

Before proceeding, we observe that a similar technique was previously used in differential cryptanalysis of DES [13]. As the permutation function of the DES round structure is just a reordering of the output bits of the substitution layer, and not a diffusion function, one can determine the output of an S-box without inverting a diffusion function. More recently, the term 'early abort' has been extensively used in the cryptanalysis of hash functions, where the technique is also sometimes known as "early stop".

The rest of this chapter is organised as follows. In Section 4.2 we describe an early abort technique for (related-key) impossible differential cryptanalysis. In Section 4.3 we describe an early abort technique for the rectangle attack. In Section 4.4 we describe an early abort technique for the related-key rectangle attack. Section 4.5 summarises the chapter.

4.2 Early Abort for (Related-Key) Impossible Differential Cryptanalysis

As discussed in Section 2.2.7, impossible differential cryptanalysis uses one or more impossible differentials, written $\Delta\alpha \nrightarrow \Delta\beta$. Such an attack involves treating a block cipher $\mathbf{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ as a cascade of three sub-ciphers: $\mathbf{E} = \mathbf{E}^a \circ \mathbf{E}^0 \circ \mathbf{E}^b$, where $\mathbf{E}^0$ denotes the rounds for which $\Delta\alpha \nrightarrow \Delta\beta$ holds, $\mathbf{E}^a$ denotes the rounds before $\mathbf{E}^0$, and $\mathbf{E}^b$ denotes the rounds after $\mathbf{E}^0$.

Given a candidate for the subkeys used in $\mathbf{E}^a$ and $\mathbf{E}^b$, if a plaintext pair produces a difference of α immediately after $\mathbf{E}^a$, and the corresponding known ciphertext pair produces a difference of β immediately before $\mathbf{E}^b$, then this candidate for the subkey must be incorrect. More specifically, suppose K_a is the guess for the subkeys used in $\mathbf{E}^a$, and K_b is the guess for the subkeys used in $\mathbf{E}^b$. Then the candidate (K_a, K_b) for the subkeys used in $\mathbf{E}^a$ and $\mathbf{E}^b$ is impossible if there is a pair of known plaintext/ciphertext pairs $((P, P'), (C, C'))$ satisfying the following two conditions:

$$\mathbf{E}^a_{K_a}(P) \oplus \mathbf{E}^a_{K_a}(P') = \alpha, \tag{4.1}$$

$$(\mathbf{E}^b_{K_b})^{-1}(C) \oplus (\mathbf{E}^b_{K_b})^{-1}(C') = \beta. \tag{4.2}$$

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

When checking whether a plaintext pair produces a difference of α just after $\mathbf{E}^a$, as in equation 4.1, (or the corresponding ciphertext pair produces a difference of β just before $\mathbf{E}^b$, as in equation 4.2), the 'standard' approach is to guess all the unknown bits of the relevant round subkey necessary to partially encrypt (or decrypt) the pair. Finally, one can check whether the pair produces the expected difference just after (or before) the round.

As an example, consider a Feistel round function structure similar to that used in Camellia, as shown in Figure 4.1. We assume that the round function $\mathbf{F}$ uses an nonlinear substitution consisting of m parallel S-boxes and a linear diffusion function $\mathbf{D}$. Suppose that (P, P') is a pair of plaintexts, (L_i, R_i) is the input to the ith round of the encryption of P, (L'_i, R'_i) is the input to the ith round of the encryption of P', (L_{i+1}, R_{i+1}) is the output of the ith round of the encryption of P, and (L'_{i+1}, R'_{i+1}) is the output of the ith round of the encryption of P'. For simplicity, we assume that the final round of $\mathbf{E}^a$ is round i, and the current task for the attacker is to check whether $(\mathbf{F}(L_i) \oplus R_i \oplus \mathbf{F}(L'_i) \oplus R'_i)||(L_i \oplus L'_i) = \alpha$.

When using the attack procedure described in Section 2.2.7, because of the use of the function $\mathbf{D}$, the attacker will need to guess all the required unknown bits of the subkey K (i.e. those corresponding to the active S-boxes). The attacker must then encrypt L_i and L'_i through the substitution layer to get the values $\mathbf{F}(L_i \oplus K)$ and $\mathbf{F}(L'_i \oplus K)$, and compute the difference $\mathbf{F}(L_i \oplus K) \oplus \mathbf{F}(L'_i \oplus K)$. Finally, the attacker XORs this difference with the difference $\Delta R_i (= R_i \oplus R'_i)$ to check whether the pair (P, P') has the difference α after round i.

However, if $\mathbf{D}$ is linear, the round structure allows us to partially determine whether

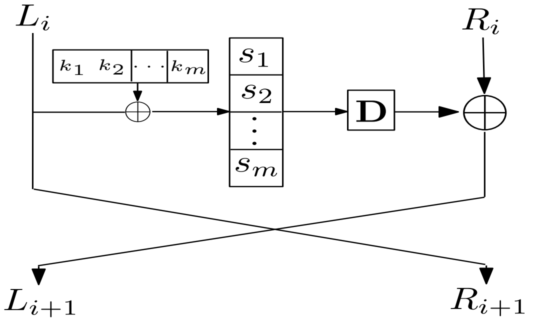


Figure 4.1: A Feistel round structure

the candidate pair (P, P') could produce the expected difference α by guessing only a small fraction of the required round subkey bits at a time, instead of all of them simultaneously. More specifically, since we know the expected difference α and the intermediate values $(L_i||R_i)$ and $(L'_i||R'_i)$ of the pair (P, P') just before the round, we can compute the expected difference $\Delta S = \mathbf{D}^{-1}(R_i \oplus R'_i \oplus \alpha_L)$ just before the **D** function, where α_L is the left half of α, because given our assumption that **D** is linearly invertible. Only if the expected difference ΔS appears after the substitution layer could the pair (P, P') produce the difference α after the round. We next guess only the part of the required unknown subkey bits corresponding to one (or more) active S-box, then encrypt the pair through the S-box, and finally check whether it produces the corresponding partial difference of ΔS. If not, then the pair (P, P') is not a valid candidate, and we can discard it immediately; otherwise, we guess another part of the required round subkey bits corresponding to another active S-box, and check again. A pair is a valid candidate only if it produces the corresponding partial difference of ΔS under each part of the required set of subkey bits. After each guess, some invalid candidate pairs can be discarded. This observation enables us to reduce an attack's computational workload, and, even more significantly, it may be possible to break more rounds of a cipher.

More delicate applications depend on the specific (related-key) impossible differentials used as well as the round function of the block cipher concerned. Examples include the cryptanalyses of AES, Camellia and HIGHT described in Chapters 5, 6 and 10.

The early abort technique can be applied in almost the same way to related-key impossible differential cryptanalysis.

4.3 Early Abort for the Rectangle Attack

The rectangle attack, (like the amplified boomerang attack), involves treating a block cipher $\mathbf{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ as a cascade of four sub-ciphers $\mathbf{E} = \mathbf{E}^a \circ \mathbf{E}^0 \circ \mathbf{E}^1 \circ \mathbf{E}^b$, where $\mathbf{E}^0 \circ \mathbf{E}^1$ denotes the rounds for which the rectangle distinguisher holds, $\mathbf{E}^a$ denotes the rounds before $\mathbf{E}^0$, and $\mathbf{E}^b$ denotes the rounds after $\mathbf{E}^1$.

Given a guess for the subkeys used in $\mathbf{E}^a$ and $\mathbf{E}^b$, the rectangle attack involves checking whether a candidate quartet consisting of two pairs of plaintext blocks meets the differential conditions required by the rectangle distinguisher. Specifically, suppose K_a is the guess for the subkeys used in $\mathbf{E}^a$, and K_b is the guess for the subkeys used in $\mathbf{E}^b$. Then the attacker checks whether a candidate quartet of known plaintext/ciphertext pairs $(((P,C),(P^*,C^*)),((P',C'),(P'^*,C'^*)))$ satisfies the following two conditions:

$$\mathbf{E}^a_{K_a}(P) \oplus \mathbf{E}^a_{K_a}(P^*) = \mathbf{E}^a_{K_a}(P') \oplus \mathbf{E}^a_{K_a}(P'^*) = \alpha, \tag{4.3}$$

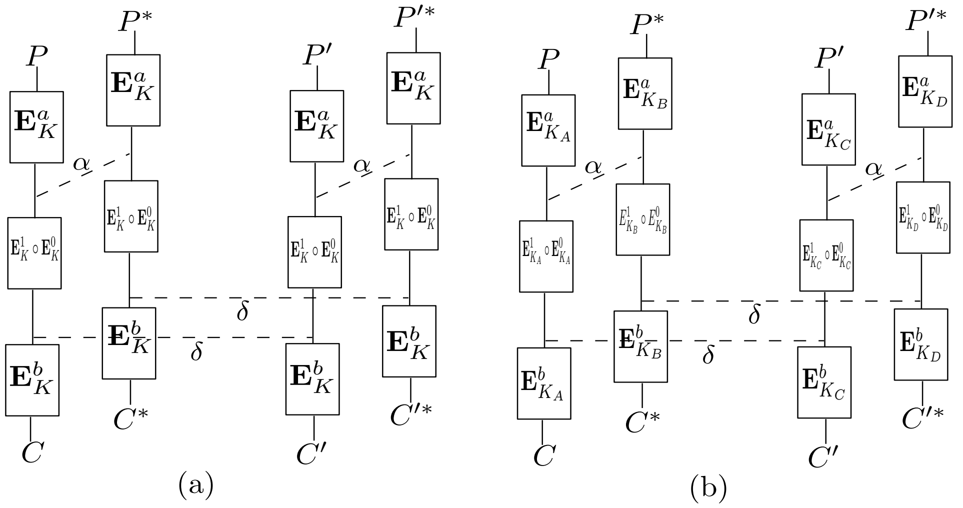
$$(\mathbf{E}^b_{K_b})^{-1}(C) \oplus (\mathbf{E}^b_{K_b})^{-1}(C') = (\mathbf{E}^b_{K_b})^{-1}(C^*) \oplus (\mathbf{E}^b_{K_b})^{-1}(C'^*) = \delta. \tag{4.4}$$

This is shown in Figure 4.2(a).

In a chosen plaintext attack scenario, the attack involves choosing the pairs (P, P^*) and (P', P'^*) in the following way.

1. Choose a plaintext, P say, and encrypt it with $\mathbf{E}^a$ under the guess K_a to obtain $\mathbf{E}^a_{K_a}(P)$.
2. Set $P^* = (\mathbf{E}^a_{K_a})^{-1}(\mathbf{E}^a_{K_a}(P) \oplus \alpha)$.
3. Choose the pair (P', P'^*) in the same way as (P, P^*).

It is straightforward to verify that a quartet $((P, P^*), (P', P'^*))$ selected in the above way meets the conditions described in equation 4.3. The remaining problem is to check whether it also meets the conditions described in equation 4.4.


P
P*
P′
P′*
$\mathbf{E}^a_K$
$\mathbf{E}^1_K \circ \mathbf{E}^0_K$
$\mathbf{E}^b_K$
α
δ
C
C*
C′
C′*
$\mathbf{E}^a_{K_A}$
$\mathbf{E}^a_{K_B}$
$\mathbf{E}^a_{K_C}$
$\mathbf{E}^a_{K_D}$
$\mathbf{E}^1_{K_A} \circ \mathbf{E}^0_{K_A}$
$\mathbf{E}^1_{K_B} \circ \mathbf{E}^0_{K_B}$
$\mathbf{E}^1_{K_C} \circ \mathbf{E}^0_{K_C}$
$\mathbf{E}^1_{K_D} \circ \mathbf{E}^0_{K_D}$
$\mathbf{E}^b_{K_A}$
$\mathbf{E}^b_{K_B}$
$\mathbf{E}^b_{K_C}$
$\mathbf{E}^b_{K_D}$

(a) (b)

Figure 4.2: The rectangle and related-key rectangle attacks

The 'standard' approach to this is to simultaneously decrypt both the pairs (C, C') and (C^*, C'^*) through $\mathbf{E}^b$ by guessing the subkeys used in $\mathbf{E}^b$. However, it may be possible to partially determine whether or not a candidate quartet in a rectangle attack is useful one or more rounds earlier than usual. More specifically, given that we know the expected output difference δ after $\mathbf{E}^1$, we may also know the expected output differences of one or more rounds after $\mathbf{E}^1$. Thus, we only need to guess part of the subkeys in $\mathbf{E}^b$ in order to check whether a candidate quartet produces one of the expected output differences in one or more of the rounds after $\mathbf{E}^1$. If not, we can discard it immediately; otherwise, we then guess part (or all) of the rest of the subkeys used in $\mathbf{E}^b$, and check the quartet in a similar way. Since some candidate quartets are discarded at each stage, this results in a smaller number of computations overall, and may allow us to break more rounds, depending on how many candidate quartets remain and how many subkeys it is necessary to guess.

We also observe that the check can be done a little more efficiently by decrypting the two pairs in a candidate quartet in a staged way. To simplify the explanation, we assume that there is only one round in $\mathbf{E}^b$. If the first pair, (C, C') say, does not meet the condition

$$(\mathbf{E}^b_{K_b})^{-1}(C) \oplus (\mathbf{E}^b_{K_b})^{-1}(C') = \delta,$$

then this candidate quartet is not useful, and we can discard it without decrypting the second pair (C^*, C'^*). If it meets this condition, then we decrypt the other pair

(C^*, C'^*) to check whether it meets the condition

$$(\mathbf{E}^b_{K_b})^{-1}(C^*) \oplus (\mathbf{E}^b_{K_b})^{-1}(C'^*) = \delta.$$

When decrypting either pair from a quartet, we can also apply the technique introduced in Section 4.2.

Generally, using this staged approach to decrypting quartets we can reduce an attack's computation workload by a factor of $O(\frac{1}{2})$. While this improvement is small for a rectangle attack, it may be significant for a related-key rectangle attack.

4.4 Early Abort for the Related-Key Rectangle Attack

A related-key rectangle attack, (like the related-key amplified boomerang attack), treats a block cipher $\mathbf{E} : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ as a cascade of four sub-ciphers $\mathbf{E} = \mathbf{E}^a \circ \mathbf{E}^0 \circ \mathbf{E}^1 \circ \mathbf{E}^b$, where $\mathbf{E}^0 \circ \mathbf{E}^1$ denotes the rounds for which the rectangle distinguisher holds, $\mathbf{E}^a$ denotes the rounds before $\mathbf{E}^0$, and $\mathbf{E}^b$ denotes the rounds after $\mathbf{E}^1$.

Given a guess for the subkeys used in $\mathbf{E}^a$ and $\mathbf{E}^b$, the related-key rectangle attack involves checking whether a candidate quartet consisting of two pairs of known plaintext/ciphertext blocks meets the differential conditions required by the related-key rectangle distinguisher. Specifically, suppose K^a_A, K^a_B, K^a_C and K^a_D are the guesses for the subkeys used in $\mathbf{E}^a$, and K^b_A, K^b_B, K^b_C and K^b_D are the guesses for the subkeys used in $\mathbf{E}^b$. Then the attacker checks whether a candidate quartet of known plaintext/ciphertext pairs $(((P,C),(P^*,C^*)),((P',C'),(P'^*,C'^*)))$ satisfies the following two conditions:

$$\mathbf{E}^a_{K^a_A}(P) \oplus \mathbf{E}^a_{K^a_B}(P^*) = \mathbf{E}^a_{K^a_C}(P') \oplus \mathbf{E}^a_{K^a_D}(P'^*) = \alpha, \tag{4.5}$$

$$(\mathbf{E}^b_{K^b_A})^{-1}(C) \oplus (\mathbf{E}^b_{K^b_C})^{-1}(C') = (\mathbf{E}^b_{K^b_B})^{-1}(C^*) \oplus (\mathbf{E}^b_{K^b_D})^{-1}(C'^*) = \delta. \tag{4.6}$$

This is shown in Figure 4.2(b).

In a chosen plaintext attack scenario, the attack involves choosing the pairs (P, P^*) and (P', P'^*) in a similar way to that described in Section 4.3, as follows.

1. Choose a plaintext, P say, and encrypt it with $\mathbf{E}^a$ under the guess K_A^a to obtain $\mathbf{E}^a_{K_A^a}(P)$.
2. Set $P^* = (\mathbf{E}^a_{K_B^a})^{-1}(\mathbf{E}^a_{K_A^a}(P) \oplus \alpha)$.
3. Choose the pair (P', P'^*) in the same way as (P, P^*).

It is straightforward to verify that a quartet $((P, P^*), (P', P'^*))$ selected in the above way meets the conditions described in equation 4.5. The remaining problem is to check whether it also meets the conditions described in equation 4.6.

The key schedules of some block ciphers make it impossible for us to determine the subkey differences used in $\mathbf{E}^b$ from the user key differences; thus it is necessary to guess the four[1] different unknown subkeys K_A^b, K_B^b, K_C^b and K_D^b used in $\mathbf{E}^b$ to check whether the candidate quartet $((P, P^*), (P', P'^*))$ meets the condition described in equation 4.6.

The 'standard' approach to this is to first guess the four subkeys at once, then decrypt both (C, C') and (C^*, C'^*) to check whether they meet the conditions described in equation 4.6. However, we observe that the check can be performed more efficiently by decrypting the two pairs from a candidate quartet in a staged way. To simplify the explanation, we assume that there is only one round in $\mathbf{E}^b$. We first guess the two subkeys K_A^b and K_C^b connected with the pair (C, C'), and then check whether the pair meets the condition

$$(\mathbf{E}^b_{K_A^b})^{-1}(C) \oplus (\mathbf{E}^b_{K_C^b})^{-1}(C') = \delta.$$

If the pair does not meet this condition, then we can discard the candidate quartet; if it meets the condition, then we guess the other two subkeys K_B^b and K_D^b connected with the other pair (C^*, C'^*), and check if this pair meets the condition

$$(\mathbf{E}^b_{K_B^b})^{-1}(C^*) \oplus (\mathbf{E}^b_{K_D^b})^{-1}(C'^*) = \delta.$$

Using this approach it may be possible to significantly reduce the workload of an attack. Even more interestingly, it may be possible to break more rounds of a cipher.

[1] We consider the related-key rectangle attack with four keys here; similar shortcuts apply in the version of the attack using two keys.

An example of this latter case is provided by the attack on SHACAL-2 described in Chapter 8.

4.5 Summary

In this chapter we have given a general description of the early abort technique for (related-key) impossible differential cryptanalysis and (related-key) rectangle attacks. This technique can, in certain circumstances, be used to improve the efficiency of an attack. More detailed descriptions of specific examples of the early abort technique are given in subsequent chapters.

We have only described the application of the early abort technique to two specific types of block cipher cryptanalysis; however, depending on the design of the round function, the technique can also be used to improve the efficiency of other cryptanalytic approaches, including differential cryptanalysis and its extensions. For example, it can be applied to the rounds preceding a differential-linear distinguisher in a differential-linear cryptanalysis procedure.

Chapter 5

Cryptanalysis of Reduced-Round AES

The Advanced Encryption Standard (AES) is a 128-bit block cipher with a user key of 128, 192 or 256 bits, which became a CRYPTREC-recommended e-government cipher in 2002, a NESSIE selected algorithm in 2003, and was adopted as an ISO international standard in 2005. In this chapter we present a number of novel attacks on reduced versions of AES; these attacks use the impossible differential and impossible boomerang techniques.

We first present the best currently published impossible differential cryptanalysis results on AES; these attacks make use of the early abort technique and a number of observations regarding the key schedule. We give an attack on 7-round AES-128 that requires $2^{112.2}$ *chosen plaintexts and has a time complexity of* $2^{115.6}$ *encryptions. We also describe two attacks on 7-round AES-192, one that requires* $2^{91.2}$ *chosen plaintexts and has a time complexity of* $2^{145.5}$ *encryptions, and another that requires* $2^{113.8}$ *chosen plaintexts and has a time complexity of* $2^{117.2}$ *encryptions. Finally we describe two attacks on 8-round AES-256, one that requires* 2^{89} *chosen plaintexts and has a time complexity of* $2^{247.7}$ *encryptions, and another that requires* $2^{111.6}$ *chosen plaintexts and has a time complexity of* $2^{233.1}$ *encryptions.*

Secondly, we present impossible boomerang attacks on 6-round AES-128, 7-round AES-192 and 7-round AES-256. The 6-round AES-128 attack requires $2^{112.2}$ *chosen plaintexts and has a time complexity of* $2^{112.3}$ *encryptions; the 7-round AES-192 attack requires* $2^{112.5}$ *chosen plaintexts and has a time complexity of* $2^{186.3}$ *encryptions; and the 7-round AES-256 attack requires* $2^{112.8}$ *chosen plaintexts and has a time complexity of* $2^{186.9}$ *encryptions.*

Finally, we present related-key impossible boomerang attacks on 8-round AES-192 and 9-round AES-256 using two keys. The 8-round AES-192 attack requires $2^{122.4}$ *chosen plaintexts and has a time complexity of* 2^{160} *encryptions; and the 9-round AES-256 attack requires* $2^{122.8}$ *chosen plaintexts and has a time complexity of* $2^{242.5}$ *encryptions. This latter attack is the first published attack on 9-round AES-256 using two keys.*

Contents

5.1 Introduction

In November 2001, NIST published the Advanced Encryption Standard (AES) [90] as the next-generation data encryption standard for use in the USA, designed to replace the Data Encryption Standard (DES) [91]. Subsequently, AES became a CRYPTREC-recommended e-government cipher in 2002, a NESSIE selected algorithm in 2003, and was adopted as an ISO international standard in 2005. AES is an SPN-based block cipher with a 128-bit block length and a user key length of 128, 192 or 256 bits. It was designed by Daemen and Rijndael [21], and was first published in 1998.

In this chapter, we first revisit the application of the impossible differential cryptanalysis technique to AES. Taking advantage of the early abort technique and certain other observations about the operation of the cipher, including some relating to the key schedule, we present the best currently published impossible differential cryptanalysis results on AES. We first give an attack on 7-round AES-128 that requires $2^{112.2}$ chosen plaintexts and has a time complexity of $2^{115.6}$ encryptions, which is also the best currently published cryptanalytic result on AES-128. Second we describe two attacks on 7-round AES-192, one that requires $2^{91.2}$ chosen plaintexts and has a time complexity of $2^{145.5}$ encryptions, and another that requires $2^{113.8}$ chosen plaintexts and has a time complexity of $2^{117.2}$ encryptions. Finally we describe two attacks on 8-round AES-256, one that requires 2^{89} chosen plaintexts and has a time complexity of $2^{247.7}$ encryptions, and another that requires $2^{111.6}$ chosen plaintexts and has a time complexity of $2^{233.1}$ encryptions.

We then present impossible boomerang attacks on 6-round AES-128, 7-round AES-192 and 7-round AES-256. The 6-round AES-128 attack requires $2^{112.2}$ chosen plaintexts and has a time complexity of $2^{112.3}$ encryptions; the 7-round AES-192 attack requires $2^{112.5}$ chosen plaintexts and has a time complexity of $2^{186.3}$ encryptions; and the 7-round AES-256 attack requires $2^{112.8}$ chosen plaintexts and has a time complexity of $2^{186.9}$ encryptions.

Finally, we present related-key impossible boomerang attacks on 8-round AES-192 and 9-round AES-256 using two keys. The 8-round AES-192 attack requires $2^{122.4}$ chosen plaintexts and has a time complexity of 2^{160} encryptions; and the 9-round AES-256 attack requires $2^{122.8}$ chosen plaintexts and has a time complexity of $2^{242.5}$ encryptions. This latter attack is the first published attack on 9-round AES-256 using two keys.

The remainder of this chapter is organised as follows. In Section 5.2 we describe AES. In Section 5.3 we briefly review previous cryptanalytic results relevant to AES. In Section 5.4 we present our impossible differential cryptanalytic results on AES. In

Section 5.5 we present our impossible boomerang cryptanalytic results on AES. In Section 5.6, we present our related-key impossible boomerang cryptanalytic results on AES. Section 5.7 summarises the main results given in this chapter.

5.2 The AES Block Cipher

In this section we briefly describe the AES block cipher [90].

5.2.1 Notation

In this chapter, the sixteen bytes of a 4×4 byte array are numbered from left to right and then from top to bottom, starting with 1 (i.e. 1, 2, $\cdots$, 16). We use the following notation.

- $\lll$: leftward rotation operation on a 32-bit word
- $\star$: an arbitrary 8-bit value, where two values represented by the $\star$ symbol may be different

5.2.2 Operations

The four elementary operations **BS**, **SR**, **MC** and **KA** are used to define the AES round function.

- **BS** (Byte Substitution) is a non-linear substitution operation on 4×4 byte arrays, constructed by applying the same 8×8-bit bijective S-box 16 times in parallel to a 4×4 byte array. See [90] for a definition of the S-box.
- **SR** (Shift Rows) is the linear function on 4×4 byte arrays which cyclically shifts the jth row of a 4×4 byte array to the left by j bytes, $(0 \leq j \leq 3)$.
- **MC** (Mix Columns) is a permutation of the set of all 4×4 byte arrays (it is a linear function over the finite field of 256 elements). It is equivalent

to pre-multiplying a 4×4 byte array by a fixed 4×4 byte array M, where addition of bytes is simply the XOR operation, and multiplication is equivalent to multiplication in the finite field of 256 elements (the field representation is defined in [90]). The matrix M is as follows.

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

- **KA** (Add Round Key) is the bitwise logical XOR operation on 4×4 byte arrays. It is used to combine a 4×4 byte array with a 16-byte subkey. If X and Y are 16-byte blocks, then $\mathbf{KA}(X, Y) = X \oplus Y$.

5.2.3 Generation of Subkeys

The AES cipher uses a total of $(N_r + 1)$ 128-bit subkeys K_i $(0 \le i \le N_r)$, all derived from the cipher key K of N_k 32-bit words long, where N_r is 10 for AES-128, 12 for AES-192, and 14 for AES-256 (i.e. the 128, 192 and 256-bit key versions of AES), and N_k is 4 for AES-128, 6 for AES-192, and 8 for AES-256. The key schedule is, where θ_{i/N_k} are public constants.

1. Represent the user key K as N_k 32-bit words $(W_1, W_2, \cdots, W_{N_k})$.
2. For $j = (N_k + 1)$ to $4(N_r + 1)$:
   - if $(j \bmod N_k = 1)$, then $W_j = W_{j-N_k} \oplus \mathbf{BS}(W_{j-1} \lll 8) \oplus \theta_{i/N_k}$;
   - else if $(N_k = 8)$ and $(j \bmod N_k = 5)$, then $W_j = W_{j-N_k} \oplus \mathbf{BS}(W_{j-1})$;
   - else $W_j = W_{j-N_k} \oplus W_{j-1}$.
3. $K_i = (W_{4i+1}, W_{4i+2}, W_{4i+3}, W_{4i+4})$, $(0 \le i \le N_r)$.

Each of the subkeys K_i consists of 16 bytes; we write $K_{i,l}$ for the lth byte of K_i, where $1 \le l \le 16$.

5.2.4 Encryption Procedure

AES takes as input a 128-bit plaintext block P, represented as a 4×4 byte array, and has a total of N_r rounds (where N_r is 10 for AES-128, 12 for AES-192, and 14 for AES-256). Its encryption procedure is as follows, where $A_0, A_i, B_i, C_i, D_i, A_{N_r}, B_{N_r}$ are 128-bit variables represented as 4×4 byte arrays.

1. $A_0 = \mathbf{KA}(P, K_0)$.
2. For $i = 1$ to $N_r - 1$:

   $B_i = \mathbf{BS}(A_{i-1})$,

   $C_i = \mathbf{SR}(B_i)$,

   $D_i = \mathbf{MC}(C_i)$,

   $A_i = \mathbf{KA}(D_i, K_i)$.
3. $A_{N_r} = \mathbf{BS}(D_{N_r-1}), B_{N_r} = \mathbf{SR}(A_{N_r})$.
4. Ciphertext= $\mathbf{KA}(B_{N_r}, K_{N_r})$.

An equivalent description of the algorithm can be derived by reversing the order of the third and fourth operations of step 2 of the above description, i.e. the operations involving **MC** and **KA**. These two steps then become:

$$D'_i = \mathbf{KA}(C_i, \widetilde{K}_i),$$

$$A_i = \mathbf{MC}(D'_i),$$

where $\widetilde{K}_i) = \mathbf{MC}^{-1}(K_i)$. (Note that $\mathbf{MC}^{-1}$ is well-defined since **MC** is a linear function equivalent to a full rank matrix). We use this alternative representation in certain of the attacks described later in this chapter.

The ith iteration of Step 2 in the above description is referred to below as Round i, $(1 \leq i \leq N_r - 1)$, and the transformation in Step 3 is referred to below as the final round, i.e. Round N_r.

5.3 Previous Cryptanalytic Results

In this section we briefly review previously published cryptanalytic attacks on AES.

- In 1998, using the square attack, the AES proposers Daemen and Rijmen [21] presented the first published attack on 6-round AES-128.
- In 2000, Gilbert and Minier [26] presented collision attacks on 7-round AES-128, 7-round AES-192 and 7-round AES-256.
- In 2000, Ferguson, Kelsey, Lucks, Schneier, Stay, Wagner and Whiting [25] presented partial sums square attacks on 7-round AES-128, 8-round AES-192 and 8-round AES-256, and presented a related-key square attack on 9-round AES-256 using 256 keys.
- In 2001, Cheon, Kim, Kim, Lee and Kang [16] presented an impossible differential attack on 6-round AES-128, building on the impossible differential attack on 5-round AES-128 of Biham and Keller [11].
- In 2003, Jakimoski and Desmedt [45] presented a related-key impossible differential attack on 8-round AES-192 using two keys.
- In 2004, Biryukov [15] presented a boomerang attack on 6-round AES-128.
- In 2004, Phan [96] presented impossible differential attacks on 7-round AES-192 and 7-round AES-256.
- In 2005, Hong, Kim, Lee and Preneel [40] presented a related-key rectangle attack on 8-round AES-192 using four keys.
- In 2005, Biham, Dunkelman and Keller [9] presented a related-key rectangle attack on 9-round AES-192 and 10-round AES-256 using 256 keys.
- In 2006, Zhang, Zhang, Wu and Feng [112] presented a related-key impossible differential attack on 8-round AES-192 using two keys, building on the related-key impossible differential attack of Biham et al. [10].
- In 2007, Kim, Hong and Preneel [52] presented related-key rectangle attacks on 8-round AES-192 using two keys, 9-round AES-192 using 64 keys, 10-round

AES-192 using 64 (or 256) keys, 9-round AES-256 using 4 keys and 10-round AES-256 using 64 (or 256) keys.

- In 2007, Bahrak and Aref [2] presented an impossible differential attack on 7-round AES-128.
- In 2007, Zhang, Wu and Feng [111] presented an impossible differential attack on 7-round AES-128, 7-round AES-192 and 8-round AES-256.
- In 2007, Zhang, Zhang, Wu and Feng [113] presented a related-key differential-linear attack [34] on 8-round AES-192 using two keys.
- In 2008, Demirci and Selcuk [22] presented a meet-in-the-middle attack on 7-round AES-192 and 8-round AES-256.

In summary, the square attack, the collision attack, the meet-in-the-middle attack, the impossible differential attack and the boomerang attack are the techniques that have previously been used to break 6 or more rounds of AES in a single key attack scenario. The best previously published cryptanalytic results on AES in a single key attack scenario are the square, collision and impossible differential attacks on 7-round AES-128 [2, 25, 26, 111], the square attack on 8-round AES-192 [25] and the square, collision and impossible differential attacks on 8-round AES-256 [22, 25, 111]. The best previously published cryptanalytic results on AES in a related-key attack scenario involving two keys are the related-key impossible differential, rectangle and differential-linear attacks on 8-round AES-192 [45, 52, 112, 113].

5.4 Impossible Differential Cryptanalysis of Reduced-Round AES

In this section, using the early abort technique as well as a number of observations on the key schedule, we present impossible differential attacks on 7-round AES-128, 7-round AES-192 and 8-round AES-256.

5.4.1 General Observations

As described in Section 2.2.7, impossible differential cryptanalysis is based on one or more impossible differentials, written $\Delta\alpha \nrightarrow \beta$. It usually involves treating a block cipher $\mathbf{E} : \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ as a cascade of three sub-ciphers $\mathbf{E} = \mathbf{E}^a \circ \mathbf{E}^0 \circ \mathbf{E}^b$, where $\mathbf{E}^0$ denotes the rounds for which $\Delta\alpha \nrightarrow \Delta\beta$ holds, $\mathbf{E}^a$ denotes a number of rounds before $\mathbf{E}^0$, and $\mathbf{E}^b$ denotes a number of rounds after $\mathbf{E}^0$. Given a guess for the subkeys used in $\mathbf{E}^a$ and $\mathbf{E}^b$, if a plaintext pair produces a difference of α just after $\mathbf{E}^a$ and the corresponding ciphertext pair produces a difference of β just before $\mathbf{E}^b$, then this guess for the subkey must be incorrect. Thus, given a sufficient number of matching plaintext/ciphertext pairs, we can find the correct subkey by discarding all the wrong guesses.

5.4.1.1 Observation I

When checking whether a plaintext pair produces a difference of α just after $\mathbf{E}^a$ (or the corresponding ciphertext pair produces a difference of β just before $\mathbf{E}^b$), the 'standard' approach is to guess all the unknown bits of the relevant round subkey necessary to partially encrypt (respectively decrypt) the pair through the substitution and diffusion layers. The attacker can then check whether the pair produces the expected difference just after $\mathbf{E}^a$ or just before $\mathbf{E}^b$. Consider the example shown in Figure 5.1. We assume that it is the the first round in $\mathbf{E}^a$, and the attacker needs to check whether a plaintext pair with a non-zero difference in only the four bytes numbered (1,6,11,16) can produce the output difference after the $\mathbf{MC}$ operation with only one non-zero byte in the first column.

Because of the diffusion properties of the $\mathbf{MC}$ operation, one possible approach, as followed in [16, 96], is to guess all the unknown required subkey bits (i.e. the four bytes (1,6,11,16) of the subkey K_0), then encrypt the pair through the $\mathbf{BS}\circ\mathbf{SR}\circ\mathbf{MC}$ operation to obtain the corresponding values just after the $\mathbf{MC}$ operation, and finally check whether they have the expected difference. This requires negligible memory, and has time complexity of one quarter of a round encryption. However, if this approach is used, then the impossible differential attacks on 7-round AES-128 and 8-round AES-256 presented in [2, 111] would have a time complexity much larger than that for an exhaustive key search; that is, they would be infeasible. Instead, a precomputation table is used, just as in [11], as we now describe.

First note that there are at most $2^{32} \times 255 \times 4 \approx 2^{42}$ pairs of 32-bit values of bytes (1,6,11,16) just after the **KA** operation which produce an output difference after the $\mathbf{BS} \circ \mathbf{SR} \circ \mathbf{MC}$ operation of the expected type (since there are very nearly 2^{40} ordered pairs that produce an output difference which have a single non-zero byte in a specified position). Let Ω_0 be the set of these 2^{42} pairs. Store all the 2^{42} pairs in Ω_0 in a table indexed by the difference between the two 32-bit values in a pair. For every such difference, there are, on average $\frac{2^{42}}{2^{32}} = 2^{10}$ pairs of 32-bit values, made up of the values of bytes (1,6,11,16) just after the **KA** operation. In addition to storing the 2^{42} pairs in Ω_0, the attacker needs to store the 2^{32} possible values of $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$ in a list. Finally, given a plaintext pair, the attacker can compute the 2^{10} values for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$ under which the plaintext pair produces the expected output difference, by XORing the plaintext pair with the approximately 2^{10} pairs of 32-bit values in the precomputation table that have the same difference as that between the two plaintexts from the plaintext pair, and then discard the 2^{10} values from the list of $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$.

To obtain all the values for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$ under which one plaintext pair produces an expected difference after applying $\mathbf{KA}\circ\mathbf{BS}\circ\mathbf{SR}\circ\mathbf{MC}$, the first approach described above requires negligible memory and has a time complexity of $2^{32} \times \frac{1}{4} = 2^{30}$ one-round encryptions. By contrast, the approach using a precomputation table requires $2^{42} \times 8 + 2^{32} \times 4 \approx 2^{45}$ bytes $\approx$ 262144 Gbits of memory, and has a time complexity of 2^{10} memory accesses (the time for precomputation is excluded, which is approximately $2^{32} \times \frac{1}{4} = 2^{30}$ one-round encryptions). That is, we have a trade-off between time (or computation workload) and memory, which makes Bahrak et al.'s and Zhang et al.'s attacks [2, 111] feasible in theory.

We now make the observation that the round structure of AES allows us to partially determine whether a candidate pair could produce the expected difference by guessing only a small fraction of the required round subkey bits at a time. We can then perform a series of partial checks by guessing other fractions of the unknown required subkey bits, instead of guessing all the unknown required subkey bits at once. More specifically, since we know the expected difference just after the **MC**

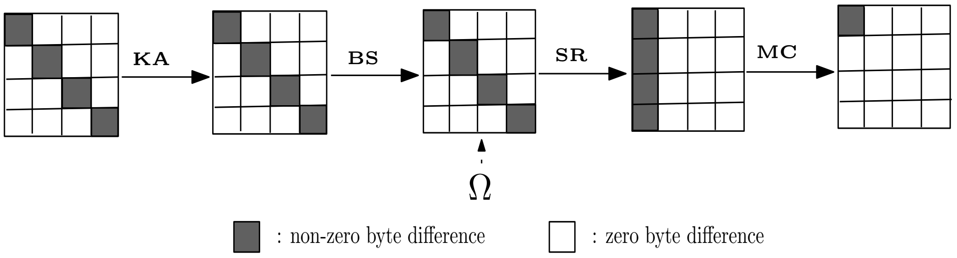


Figure 5.1: An example of the early abort technique

operation, we can compute the expected difference just before the **SR** operation, as the **MC** operation is a linear function equivalent to a full rank matrix, and hence readily invertible. There are 255 possible non-zero differences for a single byte, and thus there are a total of 4×255 possible differences with only one non-zero byte difference in the first column, since there are four different byte positions in this column. These differences are transformed by the $\mathbf{MC}^{-1} \circ \mathbf{SR}^{-1}$ operation to 4×255 possible all non-zero differences in the four bytes (1,6,11,16); we call the set of all such differences Ω. All the differences in Ω will be transformed by the $\mathbf{SR} \circ \mathbf{MC}$ operation to differences with a non-zero byte difference in only one byte of the first column.

We can now give the following result.

Property 5.1 *The differences in Ω have distinct values in the pair of byte positions $(1, 6)$.*

Proof. Suppose there exist two differences x and y from Ω that have the same value in bytes (1,6), that is to say, $x \oplus y$ is equal to zero in the first two bytes. Since x and y are transformed by the $\mathbf{MC}^{-1} \circ \mathbf{SR}^{-1}$ operation from two differences with only one non-zero byte in the first column, say $\widetilde{x}$ and $\widetilde{y}$, it follows that at least two out of the four bytes of $\widetilde{x} \oplus \widetilde{y}$ should be zero; however, this is impossible, as the **MC** operation has a branch number of 5 [21]. □

Thus, we can just guess bytes (1,6) of the subkey K_0, and partially encrypt the pair through the **BS** operation to check whether it produces a difference equal to the corresponding partial difference of any difference in Ω. If not, then the pair is not a valid candidate, and we can discard it immediately. Otherwise, by Property 5.1, we know that there is only one difference in Ω that has the same corresponding partial difference; we label this difference δ_1. We next guess another fraction of the required round subkey bits of K_0, i.e. either byte (11) or (16), and check whether the pair produces a difference equal to the corresponding partial difference in δ_1. A pair is a valid candidate only if it produces the expected partial differences just after the **BS** operation, under the guesses for the three parts, i.e. bytes (1,6), (11) and (16), of the subkey. It is expected that a proportion of about $1 - \frac{4\times 255}{2^{16}}$ of plaintext pairs will be discarded before the next guess for the subkey byte (11) or (16). Therefore, to obtain all the values for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$ under which one plaintext pair produces a particular difference after applying $\mathbf{KA} \circ \mathbf{BS} \circ \mathbf{SR} \circ \mathbf{MC}$, this approach requires $2^{10} \times 4 = 2^{12}$ bytes ≈ 32 kbits of memory, which is negligible for today's computers, and has a time complexity of approximately $2^{16} \times \frac{2}{16} + 2^{16} \times \frac{4\times 255}{2^{16}} \times 2^8 \times \frac{1}{16} + 2^{16} \times \frac{4\times 255}{2^{16}} \times 2^8 \times 2^{-8} \times 2^8 \times \frac{1}{16} \approx 2^{15.32}$ one-round encryptions. Hence, we can use this observation to reduce an attack's computational workload without using the precomputation table described earlier, and, even more significantly, we may be able to break more rounds of a cipher.

As shown in the attacks described in Sections 5.4.2–5.4.4, there exist other examples of the successful application of the early abort technique to impossible differential cryptanalysis of AES.

5.4.1.2 Observation II

From the definition of **MC** and the fact that **MC** has a branch number of 5, we can easily get the following result.

Property 5.2 *Suppose that a pair of inputs to* **MC** *(or* $\mathbf{MC}^{-1}$*) differ in only two fixed byte positions of the ith column, and in only three fixed byte positions of the output of the ith column* $(1 \leq i \leq 4)$. *Then, the following properties hold.*

1. *The number of possible pairs of input and output differences is 255.*
2. *If the difference in any of the five fixed byte positions is known, then the differences in the other four fixed byte positions can also be determined.*

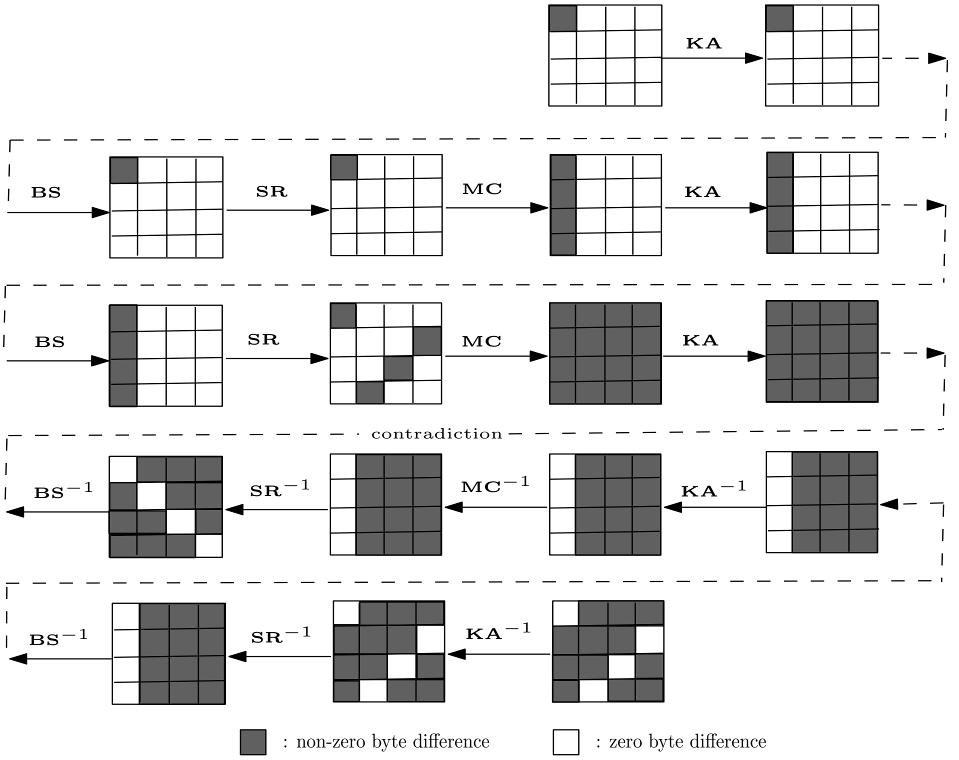


Figure 5.2: 4-round impossible differentials of AES of Biham et al.

5.4.1.3 Observation III

In 2000, Biham et al. [11] gave the following 4-round impossible differentials for AES: the input difference has zeros in all the bytes but one, and the output difference has zeros only in the four bytes of any of (1,8,11,14), (2,5,12,15), (3,6,9,16) and (4,7,10,13), (where, as described in Section 5.2, we are numbering the 16 bytes of a 128-bit block from 1 to 16). See Figure 5.2 for more details.

In 2007, Bahrak et al. [2] and Zhang et al. [111] gave a further class of 4-round impossible differentials for AES, namely: the input difference has zeros in all the bytes but one, and the output difference has zeros in all the bytes except three of a column.

We find that all the following 4-round differentials of AES are impossible: the input difference has zeros in all the bytes except one or more bytes of any of (1,6,11,16), (2,7,12,13), (3,8,9,14) and (4,5,10,15), and the output difference has zeros in the four bytes of any of (1,8,11,14), (2,5,12,15), (3,6,9,16) and (4,7,10,13), and has arbitrary values in the remaining 12 bytes.

These impossible differentials apply to any set of four consecutive rounds of AES.

5.4.2 Attacking 7-Round AES-128

In this subsection, we present an impossible differential attack on 7-round AES-128, using the early abort technique and an observation on the key schedule of AES-128. This is the best currently published cryptanalytic result on AES-128. Without loss of generality, we assume that the attacked 7 rounds are Rounds 1 to 7.

5.4.2.1 Preliminary Results

By the key schedule of AES-128, we have the following equations (5.1)–(5.3).

$$K_{7,1} = K_{6,1} \oplus \mathbf{BS}(K_{6,8}) \oplus \theta_1, \tag{5.1}$$

$$K_{7,13} = K_{6,13} \oplus \mathbf{BS}(K_{6,4}) \oplus \theta_1, \tag{5.2}$$

$$K_{7,8} = K_{6,8} \oplus K_{7,7}. \tag{5.3}$$

Hence, we can give the following property.

Property 5.3 *For AES-128, a value for* $(K_{7,1}, K_{7,7}, K_{7,8}, K_{7,13})$ *yields a 24-bit filtering condition on the possible values of* $(K_{6,1}, K_{6,4}, K_{6,8}, K_{6,13})$.

5.4.2.2 Attack Description

The above analysis enables us to give the following attack on 7-round AES-128. We use the 4-round impossible differentials of Bahrak et al. in Rounds 2 to 5, and reverse the order of the operations **MC** and **KA** for Rounds 5 and 6. Figure 5.3 illustrates the attack.

1. Choose $2^{80.2}$ structures S_i, $(i = 1, 2, \cdots, 2^{80.2})$, where a structure S_i is defined to be a set of 2^{32} plaintexts $P_{i,j}$ with bytes $(1, 6, 11, 16)$ taking all the possible values and the other 12 bytes being fixed, $(j = 1, 2, \cdots, 2^{32})$. In a chosen-plaintext attack scenario, obtain all the $2^{112.2}$ ciphertexts for the 2^{32} plaintexts in each of the $2^{80.2}$ structures; we denote by $C_{i,j}$ the ciphertext for plaintext $P_{i,j}$. Choose the plaintext pairs (P_{i,j_1}, P_{i,j_2}) such that (C_{i,j_1}, C_{i,j_2}) has a zero difference in bytes (2,3,5,6,9,12,15,16), where $1 \leq j_1 \neq j_2 \leq 2^{32}$.

2. Guess a value for the two subkey bytes $(K_{0,1}, K_{0,6})$, and perform Steps (a) and (b) below.

   (a) Partially encrypt every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) to get the corresponding values for bytes (1,6) just after the **BS** operation of Round 1, and check whether they have a difference equal to any of the corresponding two-byte partial differences in Ω, where Ω is defined in Observation I. Keep only the plaintext pairs that meet this condition. By Property 5.1, we know that there is only one difference in Ω for every such plaintext pair (P_{i,j_1}, P_{i,j_2}), and we denote this difference by δ^1_{i,j_1,j_2}.

   (b) Perform the following two sub-steps for $l = 11, 16$:

   - Guess a value for the subkey byte $K_{0,l}$.
   - Partially encrypt every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) to get the corresponding values for byte (l) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^1_{i,j_1,j_2}. Keep only the plaintext pairs that meet this condition.

3. Perform Steps (a)–(c) below for $m = 1, 5, 9, 13$:

   (a) There are 255 possible 32-bit differences in bytes (1,5,9,13) just after the **KA** operation of Round 6 that have a non-zero byte difference only in byte (m), which are transformed by the **MC** operation to 255 possible 32-bit differences in bytes (1,5,9,13) just after the **MC** operation of Round 6; we denote these differences by set Ω^m. Then, guess a value for the two subkey bytes $(K_{7,1}, K_{7,8})$, and perform the following two sub-steps.

   i. Partially decrypt every ciphertext pair (C_{i,j_1}, C_{i,j_2}) corresponding to a remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) to get the corresponding values

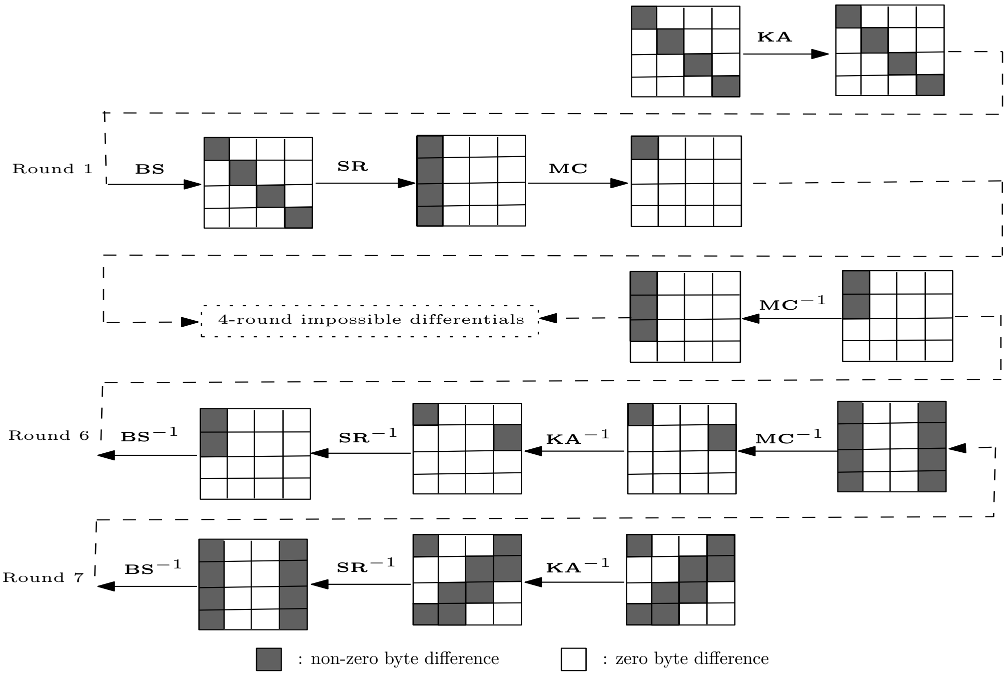


Figure 5.3: Impossible differential attack on 7-round AES-128

for bytes (1,5) just after the **MC** operation of Round 6, and check whether they have a difference equal to any of the corresponding two-byte partial differences in Ω^m. Keep only the ciphertext pairs that meet this condition. Similarly we know that there is only one difference in Ω^m for a pair (C_{i,j_1}, C_{i,j_2}) meeting the condition, and we denote this difference by δ^m_{i,j_1,j_2}.

ii. Perform the following tow sub-steps for $l = 11$, 14:

- Guess a value for the subkey byte $K_{7,l}$.
- Partially decrypt every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) to get the corresponding values for byte $(\lfloor\frac{5l-16}{4}\rfloor)$ just after the **MC** operation of Round 6, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^m_{i,j_1,j_2}. Keep only the ciphertext pairs that meet this condition.

(b) There are 255 possible 32-bit differences in bytes (4,8,12,16) just after the **KA** operation of Round 6 that have a non-zero byte difference only in byte $((m+6) \bmod 16+1)$, which are transformed by the **MC** operation to 255 possible 32-bit differences in bytes (4,8,12,16) just after the **MC** operation of Round 6; we denote these differences by set Ω^{m+7}. Then, guess a value for the two subkey bytes $(K_{7,4}, K_{7,7})$, and perform the following two sub-steps.

i. Partially decrypt every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) to get the corresponding values for bytes (4,8) just after the **MC** operation of Round 6, and check whether they have a difference equal to any of the corresponding two-byte partial differences in Ω^{m+7}. Keep only the ciphertext pairs that meet this condition. Similarly we know that there is only one difference in Ω^{m+7} for a pair (C_{i,j_1}, C_{i,j_2}) meeting the condition, and we denote this difference by δ^{m+7}_{i,j_1,j_2}.

ii. Perform the following two sub-steps for $l = 10$, 13:

- Guess a value for the subkey byte $K_{7,l}$.
- Partially decrypt every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) to get the corresponding values for byte $(\lfloor \frac{5l}{4} \rfloor)$ just after the **MC** operation of Round 6, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^{m+7}_{i,j_1,j_2}. Keep only the ciphertext pairs that meet this condition.

(c) Guess a value for the two subkey bytes $(\widetilde{K}_{6,m}, \widetilde{K}_{6,(m+6) \bmod 16+1})$. For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for bytes (1,4,5,8,9,12,13,16) just after the **MC** operation of Round 6 to get the corresponding values for bytes $(m+\lfloor \frac{m}{4} \rfloor, (m+\lfloor \frac{m}{4} \rfloor + 4) \bmod 16)$ just after the **MC** operation of Round 5, and check whether they produce a difference that has only one zero byte difference in bytes $(m + \lfloor \frac{m}{4} \rfloor, (m + \lfloor \frac{m}{4} \rfloor + 4) \bmod 16, (m + \lfloor \frac{m}{4} \rfloor + 8) \bmod 16, (m + \lfloor \frac{m}{4} \rfloor + 12) \bmod 16)$ just after the **KA** operation of Round 5. If there exists a ciphertext pair meeting this condition, discard the guessed value for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16}, K_{7,1}, K_{7,4}, K_{7,7}, K_{7,8}, K_{7,10}, K_{7,11}, K_{7,13}, K_{7,14}, \widetilde{K}_{6,m}, \widetilde{K}_{6,(m+6) \bmod 16+1})$, and try another guess.

4. For every guessed possible value for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16}, K_{7,1}, K_{7,4}, K_{7,7}, K_{7,8}, K_{7,10}, K_{7,11}, K_{7,13}, K_{7,14}, \widetilde{K}_{6,1}, \widetilde{K}_{6,4}, \widetilde{K}_{6,5}, \widetilde{K}_{6,8}, \widetilde{K}_{6,9}, \widetilde{K}_{6,12}, \widetilde{K}_{6,13}, \widetilde{K}_{6,16})$ after Step 3, check whether the value for $(K_{7,1}, K_{7,7}, K_{7,8}, K_{7,13}, \widetilde{K}_{6,1}, \widetilde{K}_{6,4}, \widetilde{K}_{6,5}, \widetilde{K}_{6,8}, \widetilde{K}_{6,9}, \widetilde{K}_{6,12}, \widetilde{K}_{6,13}, \widetilde{K}_{6,16})$ meets equations (5.1)–(5.3). If not, discard it; otherwise, determine the correct key by exhaustively searching the remaining 24 key bits.

5.4.2.3 Complexity Analysis

The attack requires $2^{112.2}$ chosen plaintexts, which take a time complexity of $2^{112.2}$ 7-round AES-128 encryptions.

In Step 1, a structure S_i yields $\binom{2^{32}}{2} \approx \frac{2^{32\times 2}}{2} = 2^{63}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}) that have a zero difference in all the bytes except bytes (1,6,11,16), $(i = 1, 2, \cdots, 2^{80.2}$, $1 \leq j_1 \neq j_2 \leq 2^{32})$. Thus the $2^{80.2}$ structures yield a total of $2^{80.2} \times 2^{63} = 2^{143.2}$ plaintext pairs that have a zero difference in all the bytes except bytes (1,6,11,16). There is a 64-bit filtering condition over the ciphertext pairs, hence it is expected that approximately $2^{143.2} \times 2^{-64} = 2^{79.2}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) are chosen in Step 1. Choosing these ciphertext pairs requires about $2^{112.2}$ memory accesses in a simple implementation using a hash table.

In Step 2(a), there are only 4×255 differences in Ω, thus it is expected that about $2^{79.2} \times \frac{4\times 255}{2^{16}} = 2^{73.2}$ plaintext pairs remain after Step 2(a) for every guess of $(K_{0,1}, K_{0,6})$. Step 2(a) has a time complexity of $2 \times 2^{79.2} \times 2^{16} \times \frac{2}{16} \times \frac{1}{7} \approx 2^{94.4}$ 7-round AES-128 encryptions.

In Step 2(b), as the difference δ^1_{i,j_1,j_2} for every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) is already fixed in Step 2(a), it is expected that for every subkey guess a proportion of about $1-2^{-8}$ of the remaining plaintext pairs will be discarded after every iteration. Step 2(b) has a time complexity of $2\times 2^{73.2}\times 2^{24}\times \frac{1}{16}\times \frac{1}{7}+2\times 2^{65.2}\times 2^{32}\times \frac{1}{16}\times \frac{1}{7} \approx 2^{96.4}$ 7-round AES-128 encryptions.

In Step 3(a)-i, for every iteration of m, there are 255 differences in Ω^m, thus the expected number of remaining pairs for every subkey guess is about $2^{57.2} \times \frac{255}{2^{16}} = 2^{49.2}$. In Step 3(a)-ii, as the difference δ^m_{i,j_1,j_2} for every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) is already fixed in Step 3(a)-i, it is expected that a proportion of about $1 - 2^{-8}$ of the remaining ciphertext pairs will be discarded after every iteration of l. Step 3(a) has a total time complexity of $4 \times (2 \times 2^{57.2} \times 2^{48} \times \frac{2}{16} \times \frac{1}{7} + 2 \times 2^{49.2} \times 2^{56} \times \frac{1}{16} \times \frac{1}{7} + 2 \times 2^{41.2} \times 2^{64} \times \frac{1}{16} \times \frac{1}{7}) \approx 2^{104.7}$ 7-round AES-128 encryptions.

In Step 3(b)-i, for every iteration of m, there are 255 differences in Ω^{m+7}, thus the expected number of remaining pairs for every subkey guess is about $2^{33.2} \times \frac{255}{2^{16}} =$

$2^{25.2}$. In Step 3(b)-ii, as the difference δ^{m+7}_{i,j_1,j_2} for every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) is already fixed in Step 3(b)-i, it is expected that for every subkey guess a proportion of about $1 - 2^{-8}$ of the remaining pairs will be discarded after every iteration. Step 3(b) has a time complexity of $4 \times (2 \times 2^{33.2} \times 2^{80} \times \frac{2}{16} \times \frac{1}{7} + 2 \times 2^{25.2} \times 2^{88} \times \frac{1}{16} \times \frac{1}{7} + 2 \times 2^{17.2} \times 2^{96} \times \frac{1}{16} \times \frac{1}{7}) \approx 2^{112.7}$ 7-round AES-128 encryptions.

In Step 3(c), for every iteration of m, the probability that a remaining ciphertext pair (C_{i,j_1}, C_{i,j_1}) meets the condition is $4 \times 2^{-8} = 2^{-6}$. For every guessed 96-bit value for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16}, K_{7,1}, K_{7,4}, K_{7,7}, K_{7,8}, K_{7,10}, K_{7,11}, K_{7,13}, K_{7,14})$, it is expected that about $2^{16} \times (1 - 2^{-6})^{2^{9.2}} \approx 2^{2.77}$ values for the two bytes $(\widetilde{K}_{6,m}, \widetilde{K}_{6,(m+6 \bmod 16)+1})$ remain after Step 3(c). After considering the four iterations of m, we get that, for every guessed value for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16}, K_{7,1}, K_{7,4}, K_{7,7}, K_{7,8}, K_{7,10}, K_{7,11}, K_{7,13}, K_{7,14})$, there remain approximately $2^{2.77\times 4} = 2^{11.08}$ possible values for $(\widetilde{K}_{6,1}, \widetilde{K}_{6,4}, \widetilde{K}_{6,5}, \widetilde{K}_{6,8}, \widetilde{K}_{6,9}, \widetilde{K}_{6,12}, \widetilde{K}_{6,13}, \widetilde{K}_{6,16})$; however, by Property 5.3, we get that there are only $2^{96} \times 2^{11.08} \times 2^{-24} = 2^{83.08}$ possible values for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16}, K_{7,1}, K_{7,4}, K_{7,7}, K_{7,8}, K_{7,10}, K_{7,11}, K_{7,13}, K_{7,14},\widetilde{K}_{6,1},\widetilde{K}_{6,4},\widetilde{K}_{6,5}, \widetilde{K}_{6,8}, \widetilde{K}_{6,9}, \widetilde{K}_{6,12}, \widetilde{K}_{6,13}, \widetilde{K}_{6,16})$. By the key schedule of AES-128, we learn that, given a value for $(K_{7,1}, K_{7,4}, K_{7,7}, K_{7,8}, K_{7,10}, K_{7,11}, K_{7,13}, K_{7,14}, \widetilde{K}_{6,1}, \widetilde{K}_{6,4}, \widetilde{K}_{6,5}, \widetilde{K}_{6,8}, \widetilde{K}_{6,9}, \widetilde{K}_{6,12}, \widetilde{K}_{6,13}, \widetilde{K}_{6,16})$, only three additional subkey bytes are required to recover the user key. Hence, Step 3(c) has a time complexity of about $4 \times \{2 \times 2^{112} \times [1 + (1 - 2^{-6}) + \cdots + (1 - 2^{-6})^{2^{9.2}}] \times \frac{2}{16} \times \frac{1}{7}\} \approx 2^{115.2}$ 7-round AES-128 encryptions.

The exhaustive search in Step 4 has a time complexity of $2^{83.08} \times 2^{24} = 2^{107.08}$ 7-round AES-128 encryptions.

Therefore, the attack has a total time complexity of approximately $2^{115.6}$ 7-round AES-128 encryptions.

5.4.3 Attacking 7-Round AES-192

In this subsection, we present an impossible differential attack on 7-round AES-192, using the early abort technique and an observation on the key schedule of AES-192. Without loss of generality, we assume that the attacked 7 rounds are Rounds 1 to 7.

5.4.3.1 Preliminary Results

By the key schedule of AES-192, we have the following equations (5.4)–(5.6).

$$K_{7,11} = K_{6,9} \oplus \mathbf{BS}(K_{7,14}) \oplus \theta_1, \tag{5.4}$$

$$K_{7,15} = K_{6,13} \oplus \mathbf{BS}(K_{7,2}) \oplus \theta_1, \tag{5.5}$$

$$K_{7,12} = K_{6,10} \oplus \mathbf{BS}(K_{7,11}). \tag{5.6}$$

We can now give the following result.

Property 5.4 *For AES-192, the value for* $(K_{6,9}, K_{6,10}, K_{6,13})$ *can be known from a value of* $(K_{7,2}, K_{7,11}, K_{7,12}, K_{7,14}, K_{7,15})$.

5.4.3.2 Attack Description

As a result, we can give the following attack procedure breaking 7-round AES-192 with a time complexity significantly lower than those for the attacks of Phan [96] and Zhang et al. [111]. We use the 4-round impossible differentials of Biham et al. in Rounds 2 to 5, and reverse the order of the operations **MC** and **KA** for Rounds 5 and 6. Figure 5.4 illustrates the attack.

1. Choose $2^{59.2}$ structures S_i, $(i = 1, 2, \cdots, 2^{59.2})$, where a structure S_i is defined to be a set of 2^{32} plaintexts $P_{i,j}$ with bytes $(1, 6, 11, 16)$ taking all the possible values and the other 12 bytes fixed, $(j = 1, 2, \cdots, 2^{32})$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{32} plaintexts in each of the $2^{59.2}$ structures; we denote by $C_{i,j}$ the ciphertext for plaintext $P_{i,j}$. Choose the plaintext pairs (P_{i,j_1}, P_{i,j_2}) such that (C_{i,j_1}, C_{i,j_2}) has a zero difference in bytes (3,4,6,7,9,10,13,16), where $1 \leq j_1 \neq j_2 \leq 2^{32}$.

2. Guess a value for the 10 subkey bytes $(K_{7,1}, K_{7,2}, K_{7,5}, K_{7,8}, K_{7,11}, K_{7,12}, K_{7,14}, K_{7,15}, K_{6,1}, K_{6,5})$. By equations (5.4) and (5.5), we deduce the value for the two subkey bytes $(K_{6,9}, K_{6,13})$, and then perform Steps (a)–(d) below.

   (a) Partially decrypt every ciphertext pair (C_{i,j_1}, C_{i,j_2}) to get the corresponding values for bytes (1,6,11,16) just after the **MC** operation of Round 5,

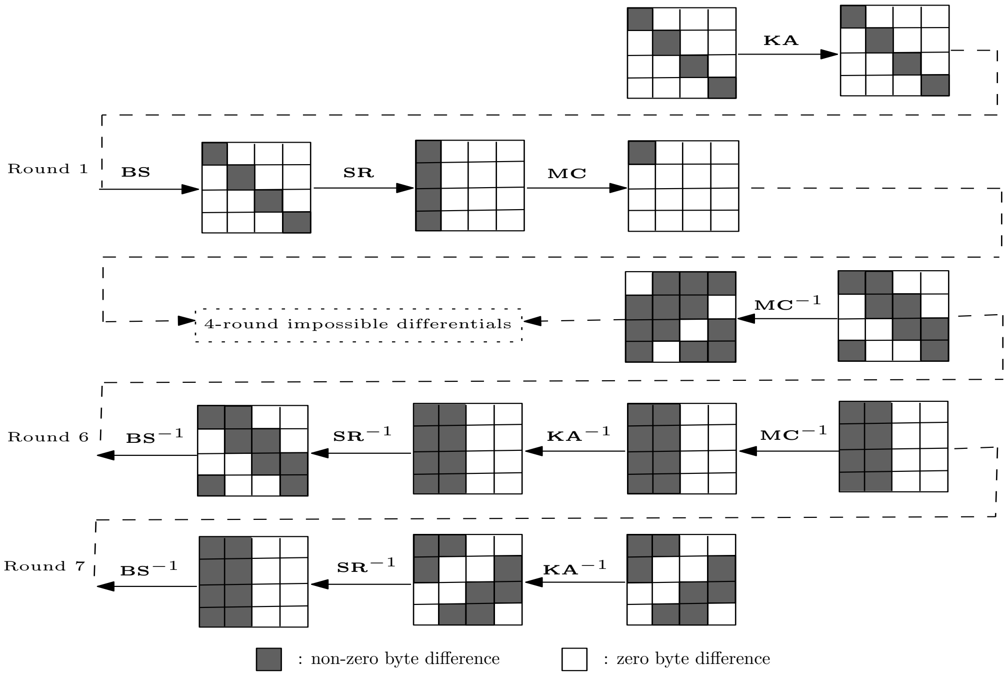


Figure 5.4: Impossible differential attack on 7-round AES-192

and compute the difference; we denote it by δ_{i,j_1,j_2}. We use all the 4-round impossible differentials that have a zero difference in only the four bytes of one of the four sets: bytes (1,8,11,14), bytes (2,5,12,15), bytes (3,6,9,16) and bytes (4,7,10,13) just after the **KA** operation of Round 5. Thus, for every ciphertext pair (C_{i,j_1}, C_{i,j_2}), by Property 5.2 we get from δ_{i,j_1,j_2} four possible 32-bit differences in bytes (2,7,12,13) just after the **MC** operation of Round 5; let Ω^5_{i,j_1,j_2} be the set of these four 32-bit differences.

(b) Guess a value for the subkey byte $\widetilde{K}_{6,2}$. For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for byte (2) just after the **MC** operation of Round 6 to get the corresponding values for byte (2) just after the **MC** operation of Round 5, and check whether they have a difference equal to any of the corresponding one-byte partial differences in Ω^5_{i,j_1,j_2}. Keep only the ciphertext pairs that meet this condition. Let δ^5_{i,j_1,j_2} be the difference in Ω^5_{i,j_1,j_2} such that a ciphertext pair (C_{i,j_1}, C_{i,j_2}) meets the condition.

(c) Perform the following two sub-steps for $l = 6, 10$:

- Guess a value for the subkey byte $\widetilde{K}_{6,l}$.
- For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for byte (l) just after the **MC** operation of Round 6 to get the corresponding values for byte ($\lfloor \frac{5l}{4} \rfloor$) just after the **MC** operation of Round 5, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^5_{i,j_1,j_2}. Keep only the ciphertext pairs that meet this condition.

(d) Guess a value for the subkey byte $\widetilde{K}_{6,14}$, and then check whether the above guessed value for $(\widetilde{K}_{6,2}, \widetilde{K}_{6,6}, \widetilde{K}_{6,10}, \widetilde{K}_{6,14})$ meets equation (5.6). If not, discard it, and guess another; otherwise, for every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for byte (14) just after the **MC** operation of Round 6 to get the corresponding values for byte (13) just after the **MC** operation of Round 5, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^5_{i,j_1,j_2}. Keep only the ciphertext pairs that meet this condition.

3. Guess a value for the subkey bytes $(K_{0,1}, K_{0,6})$, and perform Steps (a)–(c) below.

(a) Partially encrypt every plaintext pair (P_{i,j_1}, P_{i,j_2}) corresponding to a remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) to get the corresponding values for bytes (1,6) just after the **BS** operation of Round 1, and check whether they have a difference equal to any of the corresponding two-byte partial differences in Ω, where Ω is defined in Observation I. Keep only the plaintext pairs that meet this condition. By Property 5.1, we know that there is only one difference in Ω for a pair (P_{i,j_1}, P_{i,j_2}) meeting this condition, and we denote this difference by δ^1_{i,j_1,j_2}.

(b) Guess a value for the subkey byte $K_{0,11}$. Partially encrypt every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) to get the corresponding values for byte (11) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^1_{i,j_1,j_2}. Keep only the plaintext pairs that meet this condition.

(c) Guess a value for the subkey byte $K_{0,16}$. Partially encrypt every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) to get the corresponding values for byte (16) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^1_{i,j_1,j_2}. If there exists a plaintext pair meeting this condition, discard the guessed value for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16}, K_{7,1}, K_{7,2}, K_{7,5}, K_{7,8}, K_{7,11}, K_{7,12}, K_{7,14}, K_{7,15}, K_{6,1}, K_{6,5}, \widetilde{K}_{6,2}, \widetilde{K}_{6,6}, \widetilde{K}_{6,10}, \widetilde{K}_{6,14})$, and try another guess.

4. For every guessed value for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16}, K_{7,1}, K_{7,2}, K_{7,5}, K_{7,8}, K_{7,11}, K_{7,12}, K_{7,14}, K_{7,15}, K_{6,1}, K_{6,5}, \widetilde{K}_{6,2}, \widetilde{K}_{6,6}, \widetilde{K}_{6,10}, \widetilde{K}_{6,14})$ after Step 3, determine the correct key by exhaustively searching the remaining 96 key bits for the value of $(K_{7,1}, K_{7,2}, K_{7,5}, K_{7,14}, K_{6,1}, K_{6,5}, K_{6,9}, K_{6,13}, \widetilde{K}_{6,2}, \widetilde{K}_{6,6}, \widetilde{K}_{6,10}, \widetilde{K}_{6,14})$.

5.4.3.3 Complexity Analysis

The attack requires $2^{91.2}$ chosen plaintexts, which take a time complexity of $2^{91.2}$ 7-round AES-192 encryptions.

In Step 1, a structure S_i yields $\binom{2^{32}}{2} \approx \frac{2^{32\times 2}}{2} = 2^{63}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}) that have zero byte differences in all the bytes except bytes (1,6,11,16), ($i = 1, 2, \cdots, 2^{59.2}$, $1 \leq j_1 \neq j_2 \leq 2^{32}$). Thus the $2^{59.2}$ structures yield a total of $2^{59.2} \times 2^{63} = 2^{122.2}$ plaintext pairs that have a zero byte difference in all the bytes except bytes (1,6,11,16). There is a 64-bit filtering condition over the ciphertext pairs, hence it is expected that about $2^{122.2} \times 2^{-64} = 2^{58.2}$ plaintext pairs are chosen in Step 1. Choose these plaintext pairs requires about $2^{91.2}$ memory accesses in a simple implementation using a hash table.

Step 2(a) has a time complexity of $2 \times 2^{58.2} \times 2^{80} \times \frac{1}{2} \times \frac{2}{7} \approx 2^{136.4}$ 7-round AES-192 encryptions.

Step 2(b) has a time complexity of $2 \times 2^{58.2} \times 2^{88} \times \frac{1}{16} \times \frac{1}{7} \approx 2^{140.4}$ 7-round AES-192 encryptions. In Step 2(b), there are 4 differences in Ω^5_{i,j_1,j_2} given a ciphertext pair (C_{i,j_1}, C_{i,j_2}), thus the expected number of remaining ciphertext pairs after Step 2(b) for every subkey guess is about $2^{58.2} \times \frac{4}{2^8} = 2^{52.2}$.

In Step 2(c), it is expected that for every subkey guess a proportion of about $1 - 2^{-8}$ of the remaining pairs will be discarded after every iteration of l. Step 2(c) has a time complexity of $2 \times 2^{52.2} \times 2^{96} \times \frac{1}{16} \times \frac{1}{7} + 2 \times 2^{44.2} \times 2^{104} \times \frac{1}{16} \times \frac{1}{7} \approx 2^{143.4}$ 7-round AES-192 encryptions.

In Step 2(d), there is a filtering condition of 2^{-8} on the possible subkey bytes $(K_{7,1}, K_{7,2}, K_{7,5}, K_{7,8}, K_{7,11}, K_{7,12}, K_{7,14}, K_{7,15}, K_{6,1}, K_{6,5}, \widetilde{K}_{6,2}, \widetilde{K}_{6,6}, \widetilde{K}_{6,10}, \widetilde{K}_{6,14})$, so Step 2(d) has a time complexity of $2 \times 2^{36.2} \times 2^{112} \times 2^{-8} \times \frac{1}{16} \times \frac{1}{7} \approx 2^{134.4}$ 7-round AES-192 encryptions. In Step 2(d), it is expected that for every subkey guess a proportion of about $1 - 2^{-8}$ of the remaining pairs will be discarded.

Step 3(a) has a time complexity of $2 \times 2^{28.2} \times 2^{120} \times \frac{2}{16} \times \frac{1}{7} \approx 2^{143.4}$ 7-round AES-192 encryptions. In Step 3(a), there are 4×255 differences in Ω, thus the expected number of remaining pairs for every subkey guess is $2^{28.2} \times \frac{4 \times 255}{2^{16}} = 2^{22.2}$.

Step 3(b) has a time complexity of $2 \times 2^{22.2} \times 2^{128} \times \frac{1}{16} \times \frac{1}{7} \approx 2^{144.4}$ 7-round AES-192 encryptions. There is a 8-bit filtering condition in Step 3(b), thus the expected number of remaining pairs after Step 3(b) for every subkey guess is $2^{22.2} \times 2^{-8} = 2^{14.2}$.

In Step 3(c), with a probability of 2^{-8} we can get a plaintext pair meeting the condition, thus it is expected that there remain $2^{104} \times 2^{32} \times (1-2^{-8})^{2^{14.2}} \approx 2^{31}$ guessed values for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16}, K_{7,1}, K_{7,2}, K_{7,5}, K_{7,8}, K_{7,11}, K_{7,12}, K_{7,14}, K_{7,15}, K_{6,1}, K_{6,5}, \widetilde{K}_{6,2}, \widetilde{K}_{6,6}, \widetilde{K}_{6,10}, \widetilde{K}_{6,14})$. Step 3(c) has a time complexity of $2 \times 2^{136} \times [1 + (1 - 2^{-8}) + \cdots + (1 - 2^{-8})^{2^{14.2}}] \times \frac{1}{16} \times \frac{1}{7} \approx 2^{138.2}$ 7-round AES-192 encryptions.

The exhaustive search in Step 4 has a time complexity of $2^{31} \times 2^{96} = 2^{127}$ 7-round AES-192 encryptions.

Therefore, the attack has a total time complexity of approximately $2^{145.5}$ 7-round AES-192 encryptions.

Note: Another impossible differential attack on 7-round AES-192 can be obtained from the 7-round AES-128 attack presented in Section 5.4.2. After a similar analysis, we get that the attack requires $2^{113.8}$ chosen plaintexts, and has a time complexity of $2^{117.2}$ 7-round AES-192 encryptions, dramatically faster than any previously published attack on 7-round AES-192.

5.4.4 Attacking 8-Round AES-256

In this subsection, we extend the above presented 7-round AES-128/192 attack to break 8-round AES-256, using a number of specific observations for AES-256. Without loss of generality, we assume that the attacked 8 rounds are Rounds 1 to 8.

5.4.4.1 Preliminary Results

By the key schedule of AES-256, we have the following property for AES-256.

Property 5.5 *For AES-256, the value for* $(K_{6,2}, K_{6,3}, K_{6,4}, K_{6,6}, K_{6,7}, K_{6,8}, K_{6,10}, K_{6,11}, K_{6,12}, K_{6,14}, K_{6,15}, K_{6,16})$ *can be known from a value of* K_8*.*

Property 5.6 implies that, to improve an attack's efficiency, we should use some 4-round impossible differentials such that there is no need to guess any key byte of Round 6 after K_8 is known or guessed.

5.4.4.2 Extending the 7-Round AES-128 Attack to Break 8-Round AES-256

We extend the 7-round AES-128 attack by adding one more round at the end, and reverse the order of the operations **MC** and **KA** for Rounds 5, 6 and 7. As implied by Property 5.6 we use 4-round impossible differentials different from those used in the 7-round AES-128 attack. We briefly describe the attack procedure as follows.

Attack Description

1. Choose $2^{79.6}$ structures S_i, $(i = 1, 2, \cdots, 2^{79.6})$, where a structure S_i is defined to be a set of 2^{32} plaintexts $P_{i,j}$ with bytes $(1, 6, 11, 16)$ taking all the possible values and the other 12 bytes being fixed, $(j = 1, 2, \cdots, 2^{32})$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{32} plaintexts in each of the $2^{79.6}$ structures; we denote by $C_{i,j}$ the ciphertext for plaintext $P_{i,j}$.

2. Perform Steps (a)–(c) below for $(m, n) \in \{(2, 3), (2, 4), (3, 4)\}$:

   (a) Perform the following two sub-steps for $l = 1$ to 4:

   - Guess a value for the subkey bytes $(K_{8,(l-2) \bmod 4+5}, K_{8,(l-3) \bmod 4+9}, K_{8,(l-4) \bmod 4+13}, K_{8,l})$.
   - Partially decrypt bytes $(l, (l-2) \bmod 4+5, (l-3) \bmod 4+9, (l-4) \bmod 4+13)$ of every (remaining) ciphertext pair (C_{i,j_1}, C_{i,j_2}) to get the corresponding values for bytes $(l, l+4, l+8, l+12)$ just after the $\widetilde{\mathbf{KA}}$ operation of Round 7, where $1 \leq j_1 \neq j_2 \leq 2^{32}$, and check whether they have a zero byte difference only in bytes $((4-3l) \bmod 16, (4-3l-4m-4n-1) \bmod 16+1)$. Keep only the ciphertext pairs that meet this condition.

     Finally, for every remaining ciphertext pair, we know the corresponding values just after the **KA** operation of Round 7.

   (b) Perform Steps (i)–(iii) below for $l = 1$ to 4:

   i. There are 255 possible 32-bit differences in bytes $(m, m+4, m+8, m+12)$ just after the **KA** operation of Round 6 that have a non-zero byte difference only in byte $((m+4l-5) \bmod 16+1)$, which are transformed by the **MC** operation to 255 possible 32-bit differences in bytes $(m, m+4, m+8, m+12)$ just after the **MC** operation of Round 6; we denote these differences by set Ω^{m+4l}. Then, guess a value for the two subkey bytes $(\widetilde{K}_{7,m}, \widetilde{K}_{7,(m-2) \bmod 4+5})$, and perform Steps (A) and (B) below.

      A. For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for bytes $(m, (m-2) \bmod 4+5)$ just after the **KA** operation of Round 7 to get the corresponding values for bytes $(m, m+4)$ just after the **MC** operation of Round 6, and check whether they have a difference equal to any of the corresponding two-byte partial differences in Ω^{m+4l}. Keep only the ciphertext pairs that meet this condition. Similarly we know that there is only one difference in Ω^{m+4l} for a pair (C_{i,j_1}, C_{i,j_2}) meeting the condition, and we denote this difference by $\delta^{m+4l}_{i,j_1,j_2}$.

      B. Perform the following tow sub-steps for $s = (m-3) \bmod 4+9$, $(m-4) \bmod 4+13$:

      - Guess a value for the subkey byte $\widetilde{K}_{7,s}$.

- For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for byte (s) just after the **KA** operation of Round 7 to get the corresponding values for byte $(m + 4\lfloor\frac{s-1}{4}\rfloor)$ just after the **MC** operation of Round 6, and check whether they have a difference equal to the corresponding one-byte partial difference in $\delta^{m+4l}_{i,j_1,j_2}$. Keep only the ciphertext pairs that meet this condition.

ii. There are 255 possible 32-bit differences in bytes $(n, n+4, n+8, n+12)$ just after the **KA** operation of Round 6 that have a non-zero byte difference only in byte $((4m + 4l - 3n - 5) \bmod 16 + 1)$, which are transformed by the **MC** operation to 255 possible 32-bit differences in bytes $(n, n+4, n+8, n+12)$ just after the **MC** operation of Round 6; we denote these differences by set $\Omega^{4m+4l-3n}$. Then, guess a value for the two subkey bytes $(\widetilde{K}_{7,n}, \widetilde{K}_{7,(n-2) \bmod 4+5})$, and perform Steps (A) and (B) below.

A. For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for bytes $(n, (n-2) \bmod 4 + 5)$ just after the **MC** operation of Round 7 to get the corresponding values for bytes $(n, n+4)$ just after the **MC** operation of Round 6, and check whether they have a difference equal to any of the corresponding two-byte partial differences in $\Omega^{4m+4l-3n}$. Keep only the ciphertext pairs that meet this condition. Similarly we know that there is only one difference in $\Omega^{4m+4l-3n}$ for a pair (C_{i,j_1}, C_{i,j_2}) meeting the condition, and we denote this difference by $\delta^{4m+4l-3n}_{i,j_1,j_2}$.

B. Perform the following two sub-steps for $t = (n-3) \bmod 4 + 9$, $(n-4) \bmod 4 + 13$:

- Guess a value for the subkey byte $K_{7,t}$.
- For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for byte (t) just after the **MC** operation of Round 7 to get the corresponding values for byte $(n+4\lfloor\frac{t-1}{4}\rfloor)$ just after the **MC** operation of Round 6, and check whether they have a difference equal to the corresponding one-byte partial difference in $\delta^{4m+4l-3n}_{i,j_1,j_2}$. Keep only the ciphertext pairs that meet this condition.

iii. Compute the value for the two subkey bytes $(\widetilde{K}_{6,(m+4l-5) \bmod 16+1}, \widetilde{K}_{6,(4m+4l-3n-5) \bmod 16+1})$ from the $(K_{8,1}, K_{8,2}, \cdots, K_{8,16})$ guessed in Step 2(a). For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for bytes $((m + 4l - 5) \bmod 16 + 1, (4m + 4l - 3n - 5) \bmod 16 + 1)$ just after the **KA** operation of Round 6 to check whether they produce a difference that has only one zero byte difference in bytes $((m + l - 2) \bmod 4 + 1, (m + l - 2) \bmod 4+5, (m+l-2) \bmod 4+9, (m+l-2) \bmod 4+13)$ just after the **KA** operation of Round 5. Keep only the ciphertext pairs that meet this condition.

(c) Guess a value for the two subkey bytes $(K_{0,1}, K_{0,6})$, and perform Steps (i)–(iii) below.

i. Partially encrypt every plaintext pair (P_{i,j_1}, P_{i,j_2}) corresponding to a remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) to get the corresponding values for bytes (1,6) just after the **BS** operation of Round 1, and check whether they have a difference equal to any of the corresponding two-byte partial differences in Ω, where Ω is defined in Observation I. Keep only the plaintext pairs that meet this condition. By Property 5.1, we know that there is only one difference in Ω for a plaintext pair (P_{i,j_1}, P_{i,j_2}) meeting the condition, and we denote this difference by δ^1_{i,j_1,j_2}.

ii. Guess a value for the subkey byte $K_{0,11}$. Partially encrypt every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) to get the corresponding values for byte (11) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^1_{i,j_1,j_2}. Keep only the plaintext pairs that meet this condition.

iii. Guess a value for the subkey byte $K_{0,16}$. Partially encrypt every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) to get the corresponding values for byte (16) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^1_{i,j_1,j_2}. If there exists a plaintext pair meeting this condition, discard the guessed value for $(\widetilde{K}_{7,m}, \widetilde{K}_{7,(m-2) \bmod 4+5}, \widetilde{K}_{7,(m-3) \bmod 4+9}, \widetilde{K}_{7,(m-4) \bmod 4+13}, \widetilde{K}_{7,n}, \widetilde{K}_{7,(n-2) \bmod 4+5}, K_{8,1}, K_{8,2}, \cdots, K_{8,16}, \widetilde{K}_{7,(n-3) \bmod 4+9}, \widetilde{K}_{7,(n-4) \bmod 4+13}, K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$, and try another.

3. For every guessed possible value for $(K_{8,1}, K_{8,2}, \cdots, K_{8,16}, \widetilde{K}_{7,2}, \widetilde{K}_{7,3}, \widetilde{K}_{7,4}, \widetilde{K}_{7,5}, \widetilde{K}_{7,6}, \widetilde{K}_{7,7}, \widetilde{K}_{7,9}, \widetilde{K}_{7,10}, \widetilde{K}_{7,12}, \widetilde{K}_{7,13}, \widetilde{K}_{7,15}, \widetilde{K}_{7,16})$, determine the correct key by exhaustively searching the remaining 32 key bits.

Complexity Analysis

The attack requires $2^{111.6}$ chosen plaintexts, which take a time complexity of $2^{111.6}$ 8-round AES-256 encryptions.

In Step 1, the $2^{79.6}$ structures yield a total of $2^{79.6} \times 2^{63} = 2^{142.6}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}), $(i = 1, 2, \cdots, 2^{79.6}, 1 \leq j_1 \neq j_2 \leq 2^{32})$.

In Step 2(a) there is a 16-bit filtering condition in every iteration of l, thus it is expected that about $2^{142.6-16\times4} = 2^{78.6}$ ciphertext pairs pass Step 2(a) for every guessed value of $(K_{8,1}, K_{8,2}, \cdots, K_{8,16})$. Step 2(a) has a time complexity of $3 \times \sum_{i=0}^{3}(2 \times 2^{142.6} \times 2^{32\times(i+1)} \times 2^{-16\times i} \times \frac{4}{16} \times \frac{1}{8}) \approx 2^{188.2}$ 8-round AES-256 encryptions.

For every iteration of (m, n) and every iteration of l in Step 2(b)-i-A, there are 255 differences in Ω^{m+4l}, thus it is expected that $2^{78.6} \times \frac{255}{2^{16}} \approx 2^{70.6}$ ciphertext pairs pass Step 2(b)-i-A for every guess of $(\widetilde{K}_{7,m}, \widetilde{K}_{7,(m-2) \bmod 4+5}, K_{8,1}, K_{8,2}, \cdots, K_{8,16})$. For every iteration of (m, n) and every iteration of l in Step 2(b)-i-B, the difference $\delta_{i,j_1,j_2}^{m+4l}$ for every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) is already fixed in Step 2(b)-i-A, thus it is expected that for every subkey guess a proportion of $1-2^{-8}$ of remaining ciphertext pairs will be discarded after every iteration of s. It follows that about $2^{70.6-8\times2} = 2^{54.6}$ ciphertext pairs pass Step 2(b)-i for every subkey guess. Step 2(b)-i has a total time complexity of $12 \times (2 \times 2^{78.6} \times 2^{144} \times \frac{2}{16} \times \frac{1}{8} + 2 \times 2^{70.6} \times 2^{152} \times \frac{1}{16} \times \frac{1}{8} + 2 \times 2^{62.6} \times 2^{160} \times \frac{1}{16} \times \frac{1}{8}) \approx 2^{222.2}$ 8-round AES-256 encryptions.

For every iteration of (m, n) and every iteration of l in Step 2(b)-ii-A, there are 255 differences in $\Omega^{4m+4l-3n}$, thus it is expected that $2^{54.6} \times \frac{255}{2^{16}} \approx 2^{46.6}$ ciphertext pairs pass Step 2(b)-ii-A for every guess of $(K_{8,1}, K_{8,2}, \cdots, K_{8,16}, \widetilde{K}_{7,m}, \widetilde{K}_{7,(m-2) \bmod 4+5}, \widetilde{K}_{7,(m-3) \bmod 4+9}, \widetilde{K}_{7,(m-4) \bmod 4+13}, \widetilde{K}_{7,n}, \widetilde{K}_{7,(n-2) \bmod 4+5})$. For every iteration of (m, n) and every iteration of l in Step 2(b)-ii-B, the difference $\delta_{i,j_1,j_2}^{4m+4l-3n}$ for every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) is already fixed in Step 2(b)-ii-A, thus it is expected that for every subkey guess a proportion of $1 - 2^{-8}$ of remaining ciphertext pairs will be discarded after every iteration of t. It follows that about $2^{46.6-8\times 2} = 2^{30.6}$ ciphertext pairs pass Step 2(b)-i for every subkey guess. Step 2(b)-ii has a total time complexity of $12 \times (2 \times 2^{54.6} \times 2^{176} \times \frac{2}{16} \times \frac{1}{8} + 2 \times 2^{46.6} \times 2^{184} \times \frac{1}{16} \times \frac{1}{8} + 2 \times 2^{38.6} \times 2^{192} \times \frac{1}{16} \times \frac{1}{8}) \approx 2^{230.2}$ 8-round AES-256 encryptions.

For every iteration of (m, n) and every iteration of l in Step 2(b)-iii, we can get a ciphertext pair meeting the condition with a probability of $\binom{4}{1} \times 2^{-8} = 2^{-6}$. After considering the four iterations of l in Step 2(b), we expect that about $4 \times 2^{24.6} = 2^{26.6}$ ciphertext pairs pass Step 2(b)-iii for every guessed value for $(\widetilde{K}_{7,m}, \widetilde{K}_{7,(m-2) \bmod 4+5}, \widetilde{K}_{7,(m-3) \bmod 4+9}, \widetilde{K}_{7,(m-4) \bmod 4+13}, \widetilde{K}_{7,n}, \widetilde{K}_{7,(n-2) \bmod 4+5}, \widetilde{K}_{7,(n-3) \bmod 4+9}, K_{8,1}, K_{8,2}, \cdots, K_{8,16}, \widetilde{K}_{7,(n-4) \bmod 4+13})$. Step 2(b)-iii has a time complexity of $12 \times 2 \times 2^{30.6} \times 2^{192} \times \frac{2}{16} \times \frac{1}{8} \approx 2^{221.2}$ 8-round AES-256 encryptions.

In Step 2(c)-i, there are 4×255 differences in Ω, thus the expected number of remaining pairs for every subkey guess is $2^{26.6} \times \frac{4\times 255}{2^{16}} = 2^{20.6}$. There is a 8-bit filtering condition in Step 2(c)-ii, thus the expected number of remaining pairs after Step 2(c)-ii for every subkey guess is $2^{20.6} \times 2^{-8} = 2^{12.6}$. In Step 2(c)-iii, we can get a plaintext pair meeting the condition with a probability of 2^{-8}. Thus, in Step 2(c)-iii, for every guessed value for $(\widetilde{K}_{7,m}, \widetilde{K}_{7,(m-2) \bmod 4+5}, \widetilde{K}_{7,(m-3) \bmod 4+9}, \widetilde{K}_{7,(m-4) \bmod 4+13}, \widetilde{K}_{7,n}, \widetilde{K}_{7,(n-2) \bmod 4+5}, \widetilde{K}_{7,(n-3) \bmod 4+9}, \widetilde{K}_{7,(n-4) \bmod 4+13}, K_{8,1}, K_{8,2}, \cdots, K_{8,16})$, there remain about $2^{32} \times (1-2^{-8})^{2^{12.6}} \approx 2^{-2.92}$ values for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$; it follows that, given a value for $(K_{8,1}, K_{8,2}, \cdots, K_{8,16})$, every value for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$ is suggested on average by about $\frac{2^{64}\times 2^{-2.92}}{2^{32}} = 2^{29.08}$ values for $(\widetilde{K}_{7,m}, \widetilde{K}_{7,(m-2) \bmod 4+5}, \widetilde{K}_{7,(m-3) \bmod 4+9}, \widetilde{K}_{7,(m-4) \bmod 4+13}, \widetilde{K}_{7,n}, \widetilde{K}_{7,(n-2) \bmod 4+5}, \widetilde{K}_{7,(n-3) \bmod 4+9}, \widetilde{K}_{7,(n-4) \bmod 4+13})$. Considering that there are three iterations of (m, n), we get that every value for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$ is suggested by about $3 \times 2^{29.08} \times \frac{2^{29.08}}{2^{32}} = 2^{27.74}$ values for $(\widetilde{K}_{7,2}, \widetilde{K}_{7,3}, \widetilde{K}_{7,4}, \widetilde{K}_{7,5}, \widetilde{K}_{7,6}, \widetilde{K}_{7,7}, \widetilde{K}_{7,9}, \widetilde{K}_{7,10}, \widetilde{K}_{7,12}, \widetilde{K}_{7,13}, \widetilde{K}_{7,15}, \widetilde{K}_{7,16})$. As a consequence, given a value for $(K_{8,1}, K_{8,2}, \cdots, K_{8,16})$, we get $2^{32} \times 2^{27.74} = 2^{59.74}$ possible values for $(\widetilde{K}_{7,2}, \widetilde{K}_{7,3}, \widetilde{K}_{7,4}, \widetilde{K}_{7,5}, \widetilde{K}_{7,6}, \widetilde{K}_{7,7}, \widetilde{K}_{7,9}, \widetilde{K}_{7,10}, \widetilde{K}_{7,12}, \widetilde{K}_{7,13}, \widetilde{K}_{7,15}, \widetilde{K}_{7,16})$ after summarising all the 2^{32} possible values for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$, which correspond to $2^{59.74}$ possible values of $(K_{7,2}, K_{7,3}, K_{7,4}, K_{7,5}, K_{7,6}, K_{7,7}, K_{7,9}, K_{7,10}, K_{7,12}, K_{7,13}, K_{7,15}, K_{7,16})$. Step 2(c) has a total time complexity of $3 \times \{2 \times 2^{26.6} \times 2^{208} \times \frac{2}{16} \times \frac{1}{8} + 2 \times 2^{20.6} \times 2^{216} \times \frac{1}{16} \times \frac{1}{8} + 2 \times 2^{224} \times [1 + (1-2^{-8}) + \cdots + (1-2^{-8})^{2^{12.6}}] \times \frac{1}{16} \times \frac{1}{8}\} \approx 2^{232.8}$ 8-round AES-256 encryptions.

The exhaustive search in Step 3 has a time complexity of about $2^{128} \times 2^{59.74} \times 2^{32} = 2^{219.74}$ 8-round AES-256 encryptions.

Therefore, the attack has a total time complexity of approximately $2^{230.2} + 2^{232.8} \approx 2^{233.1}$ 8-round AES-256 encryptions.

5.4.4.3 Extending the 7-Round AES-192 Attack to Break 8-Round AES-256

We extend the 7-round AES-192 attack presented in Section 5.4.3 to break 8-round AES-256 by adding an additional round at the end. We reverse the order of the operations **MC** and **KA** for Rounds 5 and 7. The attack procedure is as follows.

Attack Description

1. Choose 2^{57} structures S_i, $(i = 1, 2, \cdots, 2^{57})$, where a structure S_i is defined to be a set of 2^{32} plaintexts $P_{i,j}$ with bytes $(1, 6, 11, 16)$ taking all the possible values and the other 12 bytes being fixed, $(j = 1, 2, \cdots, 2^{32})$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{32} plaintexts in each of the 2^{57} structures; we denote by $C_{i,j}$ the ciphertext for plaintext $P_{i,j}$. The 2^{57} plaintext structures yield a total of $2^{57} \times 2^{63} = 2^{120}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}), where $1 \leq j_1 \neq j_2 \leq 2^{32}$.

2. Perform Steps (a)–(c) below for $(m, n) \in \{(2,3), (2,4), (3,4)\}$.

   (a) Conduct a step similar to Step 2(a) of the 8-round AES-256 attack presented in Section 5.4.4.2.

   (b) Compute the value for the eight subkey bytes $(K_{6,m}, K_{6,m+4}, K_{6,m+8}, K_{6,n}, K_{6,m+12}, K_{6,n+4}, K_{6,n+8}, K_{6,n+12})$ from the $(K_{8,1}, K_{8,2}, \cdots, K_{8,16})$ guessed in Step 2(a). Guess a value for the eight subkey bytes $(\widetilde{K}_{7,(m-2) \bmod 4+5}, \widetilde{K}_{7,(m-3) \bmod 4+9}, \widetilde{K}_{7,(m-4) \bmod 4+13}, \widetilde{K}_{7,(n-2) \bmod 4+5}, \widetilde{K}_{7,(n-3) \bmod 4+9}, \widetilde{K}_{7,m}, \widetilde{K}_{7,n}, \widetilde{K}_{7,(n-4) \bmod 4+13})$. Partially decrypt every remaining pair of ciphertexts to check whether they produce a difference just after the **KA** operation of Round 5 that has a zero byte difference in only the four bytes of one of the four set: bytes (1,8,11,14), bytes (2,5,12,15), bytes (3,6,9,16) and bytes (4,7,10,13). Keep only the ciphertext pairs that meet this condition.

(c) Conduct a step similar to Step 2(c) of the 8-round AES-256 attack in Section 5.4.4.2.

3. Conduct a step similar to Step 3 of the 8-round AES-256 attack in Section 5.4.4.2.

Complexity Analysis

The attack requires 2^{89} chosen plaintexts, which take a time complexity of 2^{89} 8-round AES-256 encryptions.

In Step 2(a) there is a 16-bit filtering condition in every iteration of l, thus it is expected that about $2^{120-16\times4} = 2^{56}$ ciphertext pairs pass Step 2(a) for every guessed value for $(K_{8,1}, K_{8,2}, \cdots, K_{8,16})$. Step 2(a) has a time complexity of $3 \times \sum_{i=0}^{3}(2 \times 2^{120} \times 2^{32\times(i+1)} \times 2^{-16\times i} \times \frac{4}{16} \times \frac{1}{8}) \approx 2^{165.6}$ 8-round AES-256 encryptions.

In Step 2(b), for every iteration of (m, n), we can get a ciphertext pair meeting the condition with a probability of $\binom{4}{1} \times 2^{-32} = 2^{-30}$, thus it is expected that about $2^{56} \times 2^{-30} = 2^{26}$ ciphertext pairs pass Step 2(b) for every guessed subkey value. Step 2(b) has a time complexity of $3 \times 2 \times 2^{56} \times 2^{192} \times \frac{8}{16} \times \frac{2}{8} = 2^{247.6}$ 8-round AES-256 encryptions.

In Step 2(c), we similarly know that, for a guessed value for $(\widetilde{K}_{7,m}, \widetilde{K}_{7,(m-2) \bmod 4+5}, \widetilde{K}_{7,(m-3) \bmod 4+9}, \widetilde{K}_{7,(m-4) \bmod 4+13}, \widetilde{K}_{7,n}, \widetilde{K}_{7,(n-2) \bmod 4+5}, \widetilde{K}_{7,(n-3) \bmod 4+9}, K_{8,1}, K_{8,2}, \cdots, K_{8,16}, \widetilde{K}_{7,(n-4) \bmod 4+13})$, it is expected that there remain about $2^{32} \times (1 - 2^{-8})^{2^{12}} \approx 2^{8.96}$ values for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$; it follows that, given a value for $(K_{8,1}, K_{8,2}, \cdots, K_{8,16})$, every value for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$ is suggested on average by about $\frac{2^{64}\times2^{8.96}}{2^{32}} = 2^{40.96}$ values for $(\widetilde{K}_{7,m}, \widetilde{K}_{7,(m-2) \bmod 4+5}, \widetilde{K}_{7,(m-3) \bmod 4+9}, \widetilde{K}_{7,(m-4) \bmod 4+13}, \widetilde{K}_{7,n}, \widetilde{K}_{7,(n-2) \bmod 4+5}, \widetilde{K}_{7,(n-3) \bmod 4+9}, \widetilde{K}_{7,(n-4) \bmod 4+13})$. Considering that there are three different iterations of (m, n), we get that every value for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$ is suggested by about $3 \times 2^{40.96} \times \frac{2^{40.96}}{2^{32}} = 2^{51.5}$ values for $(\widetilde{K}_{7,2}, \widetilde{K}_{7,3}, \widetilde{K}_{7,4}, \widetilde{K}_{7,5}, \widetilde{K}_{7,6}, \widetilde{K}_{7,7}, \widetilde{K}_{7,9}, \widetilde{K}_{7,10}, \widetilde{K}_{7,12}, \widetilde{K}_{7,13}, \widetilde{K}_{7,15}, \widetilde{K}_{7,16})$. As a consequence, given a value for $(K_{8,1}, K_{8,2}, \cdots, K_{8,16})$, we get $2^{32} \times 2^{51.5} = 2^{83.5}$ possible values for $(\widetilde{K}_{7,2}, \widetilde{K}_{7,3}, \widetilde{K}_{7,4}, \widetilde{K}_{7,5}, \widetilde{K}_{7,6}, \widetilde{K}_{7,7}, \widetilde{K}_{7,9}, \widetilde{K}_{7,10}, \widetilde{K}_{7,12}, \widetilde{K}_{7,13}, \widetilde{K}_{7,15}, \widetilde{K}_{7,16})$, after summarising all the 2^{32} possible values for $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$, which correspond to $2^{83.5}$ possible values for $(K_{7,2}, K_{7,3}, K_{7,4}, K_{7,5}, K_{7,6}, K_{7,7}, K_{7,9}, K_{7,10}, K_{7,12}, K_{7,13}, K_{7,15}, K_{7,16})$. Step 2(c) has a total time complexity of $3 \times \{2 \times 2^{26} \times 2^{208} \times \frac{2}{16} \times \frac{1}{8} + 2 \times 2^{20} \times 2^{216} \times \frac{1}{16} \times \frac{1}{8} + 2 \times 2^{224} \times [1 + (1-2^{-8}) + \cdots + (1-2^{-8})^{2^{12}}] \times \frac{1}{16} \times \frac{1}{8}\} \approx 2^{232.2}$ 8-round AES-256 encryptions.

The exhaustive search in Step 3 has a time complexity of about $2^{128} \times 2^{83.5} \times 2^{32} = 2^{243.5}$ 8-round AES-256 encryptions.

Therefore, the attack has a total time complexity of approximately $2^{243.5} + 2^{247.6} \approx 2^{247.7}$ 8-round AES-256 encryptions.

5.5 Impossible Boomerang Attack on Reduced-Round AES

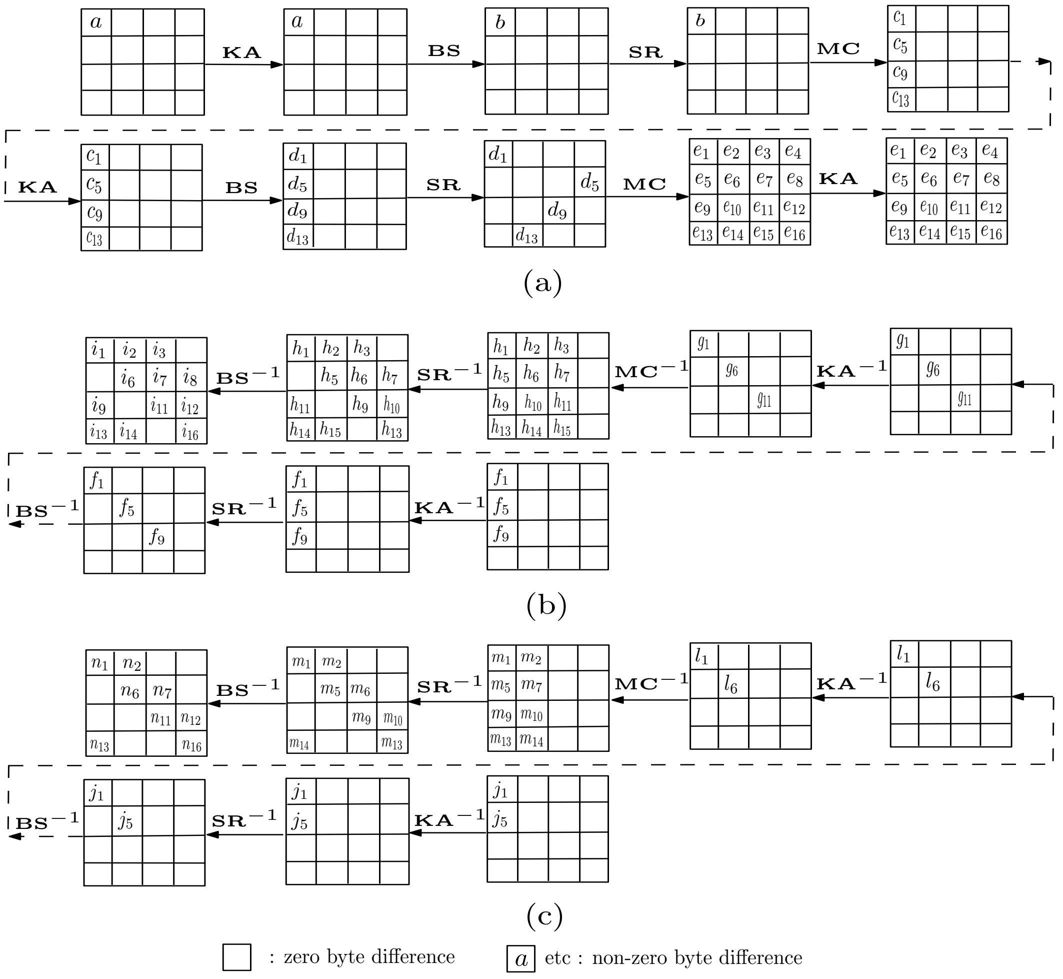
In this section we first describe certain 4-round impossible boomerang distinguishers (using two tuples) of AES. We then use them as the basis of impossible boomerang attacks on 6-round AES-128, 7-round AES-192 and 7-round AES-256.

5.5.1 4-Round Impossible Boomerang Distinguishers

We now describe certain impossible boomerang distinguishers for Rounds 2 to 5 of AES. Let $\mathbf{E}^0$ denote Rounds 2 and 3 including the **KA** operation of Round 1, and $\mathbf{E}^1$ denote Rounds 4 and 5 excluding the **MC** operation for Round 5. Figure 5.5 shows the set of four differentials making up the 4-round impossible boomerang distinguishers for $\mathbf{E}^0 \circ \mathbf{E}^1$. In this figure, a (small) square corresponds to a byte, a blank indicates a zero 8-bit difference, and a square labeled a value $a, b, \cdots$ indicates an (arbitrary[1]) non-zero 8-bit difference. The symbols given in the figure for individual byte differences are used to simplify our description below. The four differentials making up the impossible boomerang distinguisher are as follows.

The first differential $\Delta\alpha \rightarrow \Delta\beta$ for $\mathbf{E}^0$ is $((a,0,0,0),(0,0,0,0),(0,0,0,0),(0,0,0,0))$

[1] By "arbitrary" we mean that these differentials hold with probability 1.



(a)

(b)

(c)

Figure 5.5: The differentials in the 4-round impossible boomerang distinguisher

$\rightarrow ((e_1, e_2, e_3, e_4), (e_5, e_6, e_7, e_8), (e_9, e_{10}, e_{11}, e_{12}), (e_{13}, e_{14}, e_{15}, e_{16}))$, as shown in Figure 5.5(a).

The second differential $\Delta\alpha' \rightarrow \Delta\beta'$ for $\mathbf{E}^0$ has the same format as $\Delta\alpha \rightarrow \Delta\beta$; we denote it by $((a', 0, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0)) \rightarrow ((e'_1, e'_2, e'_3, e'_4), (e'_5, e'_6, e'_7, e'_8), (e'_9, e'_{10}, e'_{11}, e'_{12}), (e'_{13}, e'_{14}, e'_{15}, e'_{16}))$.

The first differential $\Delta\delta \rightarrow \Delta\gamma$ for $\mathbf{E}^1$ is $((f_1, 0, 0, 0), (f_5, 0, 0, 0), (f_9, 0, 0, 0), (0, 0, 0, 0)) \rightarrow ((i_1, i_2, i_3, 0), (0, i_6, i_7, i_8), (i_9, 0, i_{11}, i_{12}), (i_{13}, i_{14}, 0, i_{16}))$, as shown in Figure 5.5(b).

The second differential $\Delta\delta' \rightarrow \Delta\gamma'$ for $\mathbf{E}^1$ is $((j_1, 0, 0, 0), (j_5, 0, 0, 0), (0, 0, 0, 0), (0, 0, 0, 0)) \rightarrow ((n_1, n_2, 0, 0), (0, n_6, n_7, 0), (0, 0, n_{11}, n_{12}), (n_{13}, 0, 0, n_{16}))$, as shown in Figure 5.5(c).

We can now give the following result.

Property 5.6 *The four differentials described above constitute an impossible boomerang distinguisher for* $\mathbf{E}^0 \circ \mathbf{E}^1$*:* $((a,0,0,0),(0,0,0,0),(0,0,0,0),(0,0,0,0)),(a',0,0,0),(0,0,0,0),(0,0,0,0),(0,0,0,0))) \nrightarrow (((f_1,0,0,0),(f_5,0,0,0),(f_9,0,0,0),(0,0,0,0)),((j_1,0,0,0),(j_5,0,0,0),(0,0,0,0),(0,0,0,0)))$*, where* $a, a', f_1, f_5, f_9, j_1$ *and* j_5 *are arbitrary but non-zero 8-bit values.*

Proof. For the differential $\Delta\alpha \to \Delta\beta$, we have (by definition of **MC**):

$$e_5 = d_1, \tag{5.7}$$

$$e_9 = d_1. \tag{5.8}$$

Similarly, for the differential $\Delta\alpha' \to \Delta\beta'$, we have:

$$e'_5 = d'_1, \tag{5.9}$$

$$e'_9 = d'_1. \tag{5.10}$$

From [21] we know that **MC** has a branch number of 5; hence $h_{11} \neq 0$. Consequently, $i_9 \neq 0$.

Note that the 5th and 9th bytes of $\Delta\gamma$ are 0 and i_9, respectively; and the 5th and 9th bytes of $\Delta\gamma'$ are both 0. Thus, from equations (5.7) and (5.9), the 5th byte of $\beta \oplus \beta' \oplus \gamma \oplus \gamma'$ is $e_5 \oplus e'_5 = d_1 \oplus d'_1$, and by equations (5.8) and (5.10) the 9th byte of $\beta \oplus \beta' \oplus \gamma \oplus \gamma'$ is $e_9 \oplus e'_9 \oplus i_9 = d_1 \oplus d'_1 \oplus i_9$.

Since $i_9 \neq 0$, thus $d_1 \oplus d'_1$ and $d_1 \oplus d'_1 \oplus i_9$ cannot both be zero, and hence $\beta \oplus \beta' \oplus \gamma \oplus \gamma' \neq 0$ holds for the four differentials. The result follows. □

Before proceeding observe that there are many other similar 4-round impossible boomerang distinguishers for AES. These impossible boomerang distinguishers apply to any set of four consecutive rounds of AES.

5.5.2 Attacking 6-Round AES-128

We now describe an impossible boomerang attack on the first 6 rounds of AES-128 based on the above 4-round impossible boomerang distinguisher. We reverse the order of the operations **MC** and **KA** for Round 5.

5.5.2.1 Attack Description

1. Choose $2^{80.2}$ plaintext structures S_i, $(i = 1, 2, \cdots, 2^{80.2})$, where a structure S_i is defined to be a set of 2^{32} plaintexts $P_{i,j}$ with bytes $(1, 6, 11, 16)$ taking all the possible values and the other 12 bytes are fixed, $(j = 1, 2, \cdots, 2^{32})$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{32} plaintexts in each of the $2^{80.2}$ structures; let $C_{i,j}$ be the ciphertext for plaintext $P_{i,j}$. Choose the plaintext quartets $((P_{i_1,j_1}, P_{i_1,j_2}), (P_{i_2,j_3}, P_{i_2,j_4}))$ such that the corresponding ciphertext quartets $((C_{i_1,j_1}, C_{i_1,j_2}), (C_{i_2,j_3}, C_{i_2,j_4}))$ satisfy $C_{i_1,j_1} \oplus C_{i_2,j_3} = ((\star, 0, 0, 0), (0, 0, 0, \star), (0, 0, 0, 0), (0, 0, 0, 0))$ and $C_{i_1,j_2} \oplus C_{i_2,j_4} = ((\star, 0, 0, 0), (0, 0, 0, \star), (0, 0, \star, 0), (0, 0, 0, 0))$, where $1 \leq i_1, i_2 \leq 2^{80.2}, 1 \leq j_1 \neq j_2, j_3 \neq j_4 \leq 2^{32}$.

2. Guess a value for the two subkey bytes $(K_{6,1}, K_{6,8})$, and perform Steps (a) and (b) below for every remaining quartet $((C_{i_1,j_1}, C_{i_1,j_2}), (C_{i_2,j_3}, C_{i_2,j_4}))$.

   (a) Partially decrypt bytes (1,8) of C_{i_1,j_1} and C_{i_2,j_3} to get the corresponding values for bytes (1,5) just after the **MC** operation of Round 5, and check whether they produce a difference that has a zero in only one of bytes (1,5,9,13) just after the **KA** operation of Round 5. Keep only the ciphertext quartets that meet this condition.

   (b) Guess a value for the subkey byte $K_{6,11}$. Partially decrypt bytes (1,8,11) of C_{i_1,j_2} and C_{i_2,j_4} to get the corresponding values for bytes (1,5,9) just after the **MC** operation of Round 5, and check whether they produce a difference that has a zero in only two of bytes (1,5,9,13) just after the **KA** operation of Round 5 which include the one byte position with a zero difference in Step 2(a). Keep only the ciphertext quartets that meet this condition.

3. Guess a value for the four subkey bytes $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$. For every plaintext quartet $((P_{i_1,j_1}, P_{i_1,j_2}), (P_{i_2,j_3}, P_{i_2,j_4}))$ corresponding to a remaining ciphertext quartet $((C_{i_1,j_1}, C_{i_1,j_2}), (C_{i_2,j_3}, C_{i_2,j_4}))$, partially encrypt P_{i_1,j_1} and P_{i_1,j_2} to get the corresponding values for bytes (1,5,9,13) just after the **MC** operation of Round 1, and check whether they have only one non-zero byte difference; partially encrypt P_{i_1,j_2} and P_{i_2,j_4} to get the corresponding values for bytes (1,5,9,13) just after the **MC** operation of Round 1, and check whether they have only one non-zero byte difference. If there exists a plaintext quartet meeting both the conditions, discard the guessed value for $(K_{6,1}, K_{6,8}, K_{6,11}, K_{0,1}, K_{0,6}, K_{0,11}, K_{0,11})$, and try another.

4. For every guessed value for $(K_{6,1}, K_{6,8}, K_{6,11}, K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$ after Step 3, determine the correct user key by exhaustively searching the remaining 96 bits for every value of $(K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$.

5.5.2.2 Complexity Analysis

The attack requires $2^{112.2}$ chosen plaintexts, which take a time complexity of $2^{112.2}$ 6-round AES-128 encryptions.

In Step 1, a structure S_i yields about $\binom{2^{32}}{2} \approx 2^{63}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}), where $1 \leq j_1 \neq j_2 \leq 2^{32}$; thus the $2^{80.2}$ structures yield a total of $2^{80.2} \times 2^{63} = 2^{143.2}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}) that have a zero difference in all the bytes except bytes (1,6,11,16), which propose $\binom{2^{143.2}}{2} \approx 2^{285.4}$ candidate plaintext quartets $((P_{i_1,j_1}, P_{i_1,j_2}), (P_{i_2,j_3}, P_{i_2,j_4}))$, $(1 \leq i_1, i_2 \leq 2^{80.2}, 1 \leq j_3 \neq j_4 \leq 2^{32})$. Expected number of the remaining ciphertext quartets is about $2^{285.4} \times 2^{-(13+14)\times 8} = 2^{69.4}$. Choosing the useful ciphertext quartets requires $2 \times 2^{112.2} = 2^{113.2}$ memory accesses in a simple implementation.

In Step 2(a), the expected number of remaining quartets for every subkey guess is $2^{69.4} \times \binom{4}{1} \times 2^{-8} = 2^{63.4}$. Step 2(a) has a time complexity of $2 \times 2^{69.4} \times 2^{16} \times \frac{1}{6} \times \frac{2}{16} \approx 2^{80.82}$ 6-round AES-128 encryptions.

In Step 2(b), the expected number of remaining quartets for every subkey guess is $2^{63.4} \times \binom{3}{1} \times 2^{-16} \approx 2^{48.98}$. Step 2(b) has a time complexity of $2 \times 2^{63.4} \times 2^{24} \times \frac{1}{6} \times \frac{3}{16} \approx$

$2^{83.4}$ 6-round AES-128 encryptions.

In Step 3, it is expected that we can get a plaintext quartet meeting both the conditions with probability $(\binom{4}{1} \times 2^{-24})^2 = 2^{-44}$; thus after analysing $2^{48.98}$ remaining plaintext quartets we get that there remain only $2^{56} \times (1-2^{-44})^{2^{48.98}} \approx 2^{10.56}$ guessed values for $(K_{6,1}, K_{6,8}, K_{6,11}, K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$. Step 3 has a time complexity of $4 \times 2^{56} \times [1 + (1-2^{-44}) + \cdots + (1-2^{-44})^{2^{48.98}}] \times \frac{1}{6} \times \frac{4}{16} \approx 2^{97.42}$ 6-round AES-128 encryptions.

The exhaustive search in Step 4 has a time complexity of about $2^{10.56} \times 2^{96} = 2^{106.56}$ 6-round AES-128 encryptions.

Therefore, the attack has a total time complexity of approximately $2^{112.2} + 2^{106.56} \approx 2^{112.3}$ 6-round AES-128 encryptions.

5.5.3 Attacking 7-Round AES-192 and 7-Round AES-256

With an additional round appended at the end, the above 6-round AES-128 attack can be extended to break 7-round AES-192/256, as follows. We reverse the order of the operations **MC** and **KA** for Rounds 5 and 6.

1. Choose 2^x plaintext structures S_i, $(i = 1, 2, \cdots, 2^x)$, where a structure S_i is defined to be a set of 2^{32} plaintexts $P_{i,j}$ with bytes $(1, 6, 11, 16)$ taking all the possible values and the other 12 bytes being fixed, $(j = 1, 2, \cdots, 2^{32})$; and the value of x will be given below. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{32} plaintexts in each of the 2^x structures; we denote by $C_{i,j}$ the ciphertext for plaintext $P_{i,j}$. This step has a time complexity of 2^{x+32} 7-round AES-192/256 encryptions.
2. Guess a value for the subkey bytes $(K_{7,1}, K_{7,4}, K_{7,7}, K_{7,8}, K_{7,10}, K_{7,11}, K_{7,13}, K_{7,14})$, and partially decrypt bytes $(1, 4, 7, 8, 10, 11, 13, 14)$ of all the ciphertexts to get the corresponding values for bytes (1,4,5,8,9,12,13,16) just after the **KA** operation of Round 6. A structure S_i yields about $\binom{2^{32}}{2} \approx 2^{63}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}), where $1 \leq j_1 \neq j_2 \leq 2^{32}$; thus the 2^x structures yield a total of $2^x \times 2^{63} = 2^{x+63}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}) that have a zero difference in all

the bytes except bytes (1,6,11,16), which propose $\binom{2^{x+63}}{2} \approx 2^{2x+125}$ candidate plaintext quartets $((P_{i_1,j_1}, P_{i_1,j_2}), (P_{i_2,j_3}, P_{i_2,j_4}))$, where $(1 \leq i_1, i_2 \leq 2^x, 1 \leq j_3 \neq j_4 \leq 2^{32})$. Choose the ciphertext quartets $((C_{i_1,j_1}, C_{i_1,j_2}), (C_{i_2,j_3}, C_{i_2,j_4}))$ that satisfy the following three conditions:

(i) $C_{i_1,j_1} \oplus C_{i_2,j_3} = ((\star, 0, 0, \star), (0, 0, \star, \star), (0, \star, \star, 0), (\star, \star, 0, 0))$;

(ii) $C_{i_1,j_2} \oplus C_{i_2,j_4} = ((\star, 0, \star, \star), (0, \star, \star, \star), (\star, \star, \star, 0), (\star, \star, 0, \star))$.

(iii) Either of the pairs $(C_{i_1,j_1}, C_{i_2,j_3})$ and $(C_{i_1,j_2}, C_{i_2,j_4})$ has a non-zero difference only in bytes (1,8) of bytes (1,4,5,8,9,12,13,16) just after the **KA** operation of Round 6;

This step has a time complexity of $2 \times 2^{x+32} \times 2^{64} = 2^{x+97}$ memory accesses and $2^{x+32} \times 2^{64} \times \frac{8}{16} = 2^{x+95}$ one-round AES-192/256 encryptions. It is expected that there remain $2^{2x+125} \times 2^{-48\times 2} \times 2^{-64-32} = 2^{2x-67}$ ciphertext quartets for every subkey guess.

3. Guess a value for the four subkey bytes $(K_{7,3}, K_{7,6}, K_{7,9}, K_{7,16})$. For every remaining ciphertext quartet $((C_{i_1,j_1}, C_{i_1,j_2}), (C_{i_2,j_3}, C_{i_2,j_4}))$, partially decrypt bytes (3,6,9,16) of C_{i_1,j_2} and C_{i_2,j_4} to get the corresponding values for bytes (3,7,11,15) just after the **KA** operation of Round 6, and check whether they have a non-zero difference only in byte (11). Keep only the ciphertext quartets that meet this condition.

   This step has a time complexity of $2 \times 2^{2x-67} \times 2^{96} \times \frac{4}{16} = 2^{2x+28}$ one-round AES-192/256 encryptions. It is expected that there remain $2^{2x-67} \times 2^{-24} = 2^{2x-91}$ ciphertext quartets for every subkey guess.

4. Conduct a step similar to Step 2 of the 6-round AES-128 attack.

   This step has a time complexity of $2 \times 2^{2x-91} \times 2^{112} \times \frac{2}{16} + 2 \times 2^{2x-97} \times 2^{120} \times \frac{3}{16} \approx 2^{2x+21.8}$ one-round AES-192/256 encryptions. It is expected that there remain $2^{2x-91} \times \binom{4}{1} \times 2^{-8} \times \binom{3}{1} \times 2^{-16} \approx 2^{2x-111.42}$ ciphertext quartets for every subkey guess.

5. Guess a value for the two subkey bytes $(K_{0,1}, K_{0,6})$. Perform Steps (a)–(c) below for every plaintext quartet $((P_{i_1,j_1}, P_{i_1,j_2}), (P_{i_2,j_3}, P_{i_2,j_4}))$ corresponding to a remaining quartet $((C_{i_1,j_1}, C_{i_1,j_2}), (C_{i_2,j_3}, C_{i_2,j_4}))$.

   (a) Partially encrypt P_{i_1,j_1} and P_{i_1,j_2} to get the corresponding values for bytes (1,6) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding two-byte difference of one of those in set Ω defined in Section 5.4.1; if yes, we denote by $\Delta s_{i_1,j_1,j_2}$ the difference from Ω for $(P_{i_1,j_1}, P_{i_1,j_2})$. Partially encrypt P_{i_2,j_3} and P_{i_2,j_4} to get the corresponding values for bytes (1,6) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding two-byte difference of one of those in set Ω defined in Section 5.4.1; if yes, we denote by $\Delta t_{i_2,j_3,j_4}$ the difference from Ω for $(P_{i_2,j_3}, P_{i_2,j_4})$. Keep the plaintext quartets that meet both the conditions.

This step has a time complexity of $2 \times 2^{2x-111.42} \times 2^{136} \times \frac{2}{16} = 2^{2x+22.6}$ one-round AES-192/256 encryptions. It is expected that there remain $2^{2x-111.42} \times 2^{-6\times 2} = 2^{2x-123.42}$ plaintext quartets for every subkey guess.

(b) Guess a value for the subkey byte $K_{0,11}$. Partially encrypt P_{i_1,j_1} and P_{i_1,j_2} to get the corresponding values for byte (11) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding one-byte difference of $\Delta s_{i_1,j_1,j_2}$; and partially encrypt P_{i_2,j_3} and P_{i_2,j_4} to get the corresponding values for byte (11) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding one-byte difference of $\Delta t_{i_2,j_3,j_4}$. Keep the plaintext quartets that meet both the conditions.

This step has a time complexity of $2 \times 2^{2x-123.42} \times 2^{144} \times \frac{1}{16} = 2^{2x+17.6}$ one-round AES-192/256 encryptions. It is expected that there remain $2^{2x-123.42} \times 2^{-8\times 2} = 2^{2x-139.42}$ plaintext quartets for every subkey guess.

(c) Guess a value for the subkey byte $K_{0,16}$. Partially encrypt P_{i_1,j_1} and P_{i_1,j_2} to get the corresponding values for byte (16) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding one-byte difference of $\Delta s_{i_1,j_1,j_2}$; and partially encrypt P_{i_2,j_3} and P_{i_2,j_4} to get the corresponding values for byte (16) just after the **BS** operation of Round 1, and check whether they have a difference equal to the corresponding one-byte difference of $\Delta t_{i_2,j_3,j_4}$. If there exists a plaintext quartet meeting both the conditions, discard the guessed value for $(K_{7,1}, K_{7,3}, K_{7,4}, K_{7,6}, K_{7,7}, K_{7,8}, K_{7,9}, K_{7,10}, K_{7,11}, K_{7,13}, K_{7,14}, K_{7,16}, K_{6,1}, K_{6,8}, K_{6,11}, K_{0,1}, K_{0,6}, K_{0,11}, K_{0,16})$, and try another.

The probability that there exists a plaintext quartet meeting both the conditions is $2^{-8\times 2} = 2^{-16}$; thus the probability that a subkey guess remains after the remaining $2^{2x-139.42}$ quartets are tested is $(1-2^{-16})^{2^{2x-139.42}}$

$\approx \mathsf{e}^{-2^{2x-155.42}}$, here $\mathsf{e}(= 2.71828\ldots)$ is the base of the natural logarithm. This step has a time complexity of $4 \times 2^{152} \times [1 + (1-2^{-16}) + \cdots + (1-2^{-16})^{2^{2x-139.42}}] \times \frac{1}{16} \approx 2^{166}$ one-round AES-192/256 encryptions.

6. For every remaining value for $(\widetilde{K}_{6,8}, \widetilde{K}_{6,11}, K_{7,1}, K_{7,3}, K_{7,4}, K_{7,6}, K_{7,7}, K_{7,8}, K_{7,9}, K_{7,10}, K_{7,11}, K_{7,13}, K_{7,14}, K_{7,16})$ under AES-192 or every remaining value for $(\widetilde{K}_{6,1}, \widetilde{K}_{6,8}, \widetilde{K}_{6,11}, K_{7,1}, K_{7,3}, K_{7,4}, K_{7,6}, K_{7,7}, K_{7,8}, K_{7,9}, K_{7,10},K_{7,11},K_{7,13},K_{7,14}, K_{7,16})$ under AES-256, determine the correct user key by exhaustively searching the remaining bits.

For AES-192, the attack requires $x = 2^{80.5}$ plaintext structures and has a time complexity of $(2^{189} + 2^{182.8} + 2^{183.6} + 2^{152} \times \mathsf{e}^{-2^{2\times 80.5-155.42}} \times 2^{80}) \times \frac{1}{7} \approx 2^{186.3}$ 7-round AES-192 encryptions.

For AES-256, the attack requires $x = 2^{80.8}$ plaintext structures and has a time complexity of $(2^{189.6} + 2^{183.4} + 2^{184.2} + 2^{152} \times \mathsf{e}^{-2^{2\times 80.8-155.42}} \times 2^{136}) \times \frac{1}{7} \approx 2^{186.9}$ 7-round AES-256 encryptions.

5.6 Related-Key Impossible Boomerang Attack on Reduced-Round AES

In this section we describe 6-round related-key impossible boomerang distinguishers of AES-192/256, and use them to conduct a related-key impossible boomerang attack on 8-round AES-192 and 9-round AES-256 using two keys.

Let $\mathbf{E}^0$ denote Rounds 2 to 5 (of AES-192/256) including the **KA** operation of Round 1, $\mathbf{E}^1$ denote Rounds 6 to 7 excluding the **MC** operation of Round 7. We use a related-key impossible boomerang distinguisher such that $K_A = K_C$ and $K_B = K_D$.

5.6.1 Attacking 8-Round AES-192 Using Two Related Keys

The related-key differentials $\Delta\alpha \rightarrow \Delta\beta$ and $\Delta\alpha' \rightarrow \Delta\beta'$ for $\mathbf{E}^0$ are both $((0,0,a,a),(0,0,0,0),(0,0,0,0),(0,0,0,0)) \rightarrow ((\star,\star,\star,\star),(\star,\star,\star,\star),(\star,\star,\star,\star),(\star,\star,\star,\star))$, where the use key difference is $K_A \oplus K_B (= K_C \oplus K_D) = ((a,0,a,0,0,0),(0,0,0,0,0,0),(0,0,0,0,0,0),(0,0,0,0,0,0))$, with a being a specific non-zero 8-bit value. The same differentials as those depicted in Figure 5.5(b) and (c) are used for $\mathbf{E}^1$.

Table 5.1 gives the subkey differences for the first eight rounds of AES-192 given the user key difference $((a,0,a,0,0,0),(0,0,0,0,0,0),(0,0,0,0,0,0),(0,0,0,0,0,0))$, where b and c are indeterminate 8-bit values.

Similarly, we can learn that there exist the following 6-round related-key impossible boomerang distinguishers for $\mathbf{E}^0 \circ \mathbf{E}^1$: $(((0,0,a,a),(0,0,0,0),(0,0,0,0),(0,0,0,0)),((0,0,a,a),(0,0,0,0),(0,0,0,0),(0,0,0,0))) \nrightarrow (((\star,0,0,0),(\star,0,0,0),(\star,0,0,0),(0,0,0,0)),((\star,0,0,0),(\star,0,0,0),(0,0,0,0),(0,0,0,0)))$.

Table 5.1: Subkey differences for the 8-round AES-192 attack

(i)	ΔK_{5i}	ΔK_{5i+1}	ΔK_{5i+2}	ΔK_{5i+3}	ΔK_{5i+4}
0	$\begin{pmatrix} a & 0 & a & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & a & a \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} a & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & a & a \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$
1	$\begin{pmatrix} a & a & a & a \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} a & 0 & a & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \end{pmatrix}$	$\begin{pmatrix} a & 0 & a & a \\ 0 & 0 & 0 & 0 \\ 0 & 0 & c & c \\ b & b & b & 0 \end{pmatrix}$	$\begin{pmatrix} 0 & 0 & a & a \\ 0 & 0 & 0 & 0 \\ c & c & c & c \\ b & 0 & b & 0 \end{pmatrix}$	/

As a result, we can conduct a related-key impossible boomerang attack on AES-192 reduced to the first 8 rounds (i.e. Rounds 1 to 8), similarly to the 6-round AES-128 attack given in Section 5.5.2.

From Table 5.1 we know that the differences for the subkey bytes $K_{8,1}$ and $K_{8,8}$ are both zero. After an analysis similar to the 6-round AES-128 attack described in Section 5.5.2, we get that, the attack requires $2^{90.4}$ plaintext structures, and has a time complexity of $2 \times 2^{90.4+32} + 2 \times 2^{2\times 90.4-91} \times 2^{16} \times \frac{1}{8} \times \frac{2}{16} + 2 \times 2^{2\times 90.4-97} \times 2^{24} \times \frac{1}{8} \times \frac{3}{16} + 4 \times 2^{56} \times [1 + (1-2^{-64}) + \cdots + (1-2^{-64})^{2\times 90.4-111.42}] \times \frac{1}{8} \times \frac{4}{16} + 2^{160} \approx 2^{160}$ 8-round AES-192 encryptions, recovering the entire 192-bit user key.

5.6.2 Attacking 9-Round AES-256 Using Two Related Keys

The related-key differentials $\Delta\alpha \to \Delta\beta$ and $\Delta\alpha' \to \Delta\beta'$ for $\mathbf{E}^0$ are both $((0,a,a,0), (0,0,0,0),(0,0,0,0),(0,0,0,0)) \to ((\star,\star,\star,\star),(\star,\star,\star,\star),(\star,\star,\star,\star),(\star,\star,\star,\star))$, where the use key difference is $K_A \oplus K_B (= K_C \oplus K_D) = ((0,0,0,0,0,a,a,0),(0,0,0,0,0,0,0,0),(0,0,0,0,0,0,0,0),(0,0,0,0,0,0,0,0))$, with a being a specific non-zero 8-bit value. The same differentials as those in Figure 5.5(b) and (c) are used for $\mathbf{E}^1$.

Table 5.2 gives the subkey differences for the first nine rounds of AES-256 given the user key difference $((0,0,0,0,0,a,a,0),(0,0,0,0,0,0,0,0),(0,0,0,0,0,0,0,0),(0,0,0,0,0,0,0,0))$, where b,c,d,e,f,g are indeterminate 8-bit values.

Table 5.2: Subkey differences for the 9-round AES-256 attack

(i)	ΔK_{5i}	ΔK_{5i+1}	ΔK_{5i+2}	ΔK_{5i+3}	ΔK_{5i+4}
0	$\begin{pmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix}$	$\begin{pmatrix} 0&a&a&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix}$	$\begin{pmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix}$	$\begin{pmatrix} 0&a&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix}$	$\begin{pmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix}$
1	$\begin{pmatrix} 0&a&a&a \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix}$	$\begin{pmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ b&b&b&b \end{pmatrix}$	$\begin{pmatrix} 0&a&0&a \\ 0&0&0&0 \\ 0&0&0&0 \\ c&c&c&c \end{pmatrix}$	$\begin{pmatrix} 0&0&0&0 \\ 0&0&0&0 \\ d&d&d&d \\ b\oplus e&e&b\oplus e&e \end{pmatrix}$	$\begin{pmatrix} 0&a&a&0 \\ 0&0&0&0 \\ f&f&f&f \\ g\oplus c&g&g\oplus c&g \end{pmatrix}$

We can similarly learn that there exist the following 6-round related-key impossible boomerang distinguishers for $\mathbf{E}^0 \circ \mathbf{E}^1$: $(((0,a,a,0),(0,0,0,0),(0,0,0,0),(0,0,0,0)), ((0,a,a,0),(0,0,0,0),(0,0,0,0),(0,0,0,0))) \nrightarrow (((\star,0,0,0),(\star,0,0,0),(\star,0,0,0),(0,0,0,0)),((\star,0,0,0),(\star,0,0,0),(0,0,0,0),(0,0,0,0)))$.

Subsequently, we can conduct a related-key impossible boomerang attack on AES-256 reduced to the first 9 rounds (i.e. Rounds 1 to 9), similarly to the 7-round AES-192/256 attack in Section 5.5.3. We reverse the order of the operations **MC** and **KA** for Rounds 7 and 8. From the key difference $K_A \oplus K_B$ we have:

(i) The differences for $K_{9,1}$, $K_{9,4}$, $K_{9,7}$ and $K_{9,8}$ are all zero;

(ii) The differences for $K_{9,10}$ and $K_{9,11}$ are identical and indeterminate non-zero values;

(iii) The differences for $K_{9,13}$ and $K_{9,14}$ are different and indeterminate non-zero values, with neither of them equal to the difference for $K_{9,10}$ (or $K_{9,11}$);

(iv) The differences for $\widetilde{K}_{8,1}$ and $\widetilde{K}_{8,8}$ are indeterminate.

Thus, when conducting a step similar to Step 2 of the 7-round AES-192/256 attack, we first guess a value for the eight subkey bytes $(K_{9,1}, K_{9,4}, K_{9,7}, K_{9,8}, K_{9,10}, K_{9,11}, K_{9,13}, K_{9,14})$ of K_A, and partially decrypt all the ciphertexts to get the corresponding values for bytes (1,4,5,8,9,12,13,16) just after the **KA** operation of Round 8; then for every guessed value for $(K_{9,1}, K_{9,4}, K_{9,7}, K_{9,8},\ K_{9,10}, K_{9,11}, K_{9,13}, K_{9,14})$ of K_A, we guess a value for the differences for $K_{9,10}$, $K_{9,13}$ and $K_{9,14}$, compute the eight subkey bytes $(K_{9,1}, K_{9,4}, K_{9,7}, K_{9,8}, K_{9,10}, K_{9,11}, K_{9,13}, K_{9,14})$ of K_B, and partially decrypt all the ciphertexts to get the corresponding values for bytes (1,4,5,8,9,12,13,16) just after the **KA** operation of Round 8. Finally, choose the ciphertext quartets $((C_{i_1,j_1}, C_{i_1,j_2}), (C_{i_2,j_3}, C_{i_2,j_4}))$ that meet the following three conditions:

(i) $C_{i_1,j_1} \oplus C_{i_2,j_3} = ((\star,0,0,\star),(0,0,\star,\star),(0,\star,\star,0),(\star,\star,0,0))$;

(ii) $C_{i_1,j_2} \oplus C_{i_2,j_4} = ((\star,0,\star,\star),(0,\star,\star,\ \star),(\star,\star,\star,0),(\star,\star,0,\star))$;

(iii) For either of the pairs $(C_{i_1,j_1}, C_{i_2,j_3})$ and $(C_{i_1,j_2}, C_{i_2,j_4})$, the difference between the corresponding values for bytes (1,4,5,8,9,12,13,16) just after the **KA** operation of Round 8 has a non-zero byte difference only in bytes (1,8).

Subsequently, after an analysis similar to the 7-round AES-192/256 attack described in Section 5.5.3, we get that, the attack requires $2^{90.8}$ plaintext structures, and has a time complexity of $2\times2^{90.8+32}+2^{90.8+32}\times2^{64}\times\frac{1}{9}\times\frac{8}{16}+2^{90.8+32}\times2^{64+24}\times\frac{1}{9}\times\frac{8}{16}+2\times2^{2\times90.8-67}\times2^{88+32}\times\frac{1}{9}\times\frac{4}{16}+2\times2^{2\times90.8-91}\times2^{120+16}\times\frac{1}{9}\times\frac{2}{16}+2\times2^{2\times90.8-97}\times2^{136+24}\times\frac{1}{9}\times\frac{3}{16}+4\times2^{2\times90.8-111.42}\times2^{160+16}\times\frac{1}{9}\times\frac{2}{16}+4\times2^{2\times90.8-143.42}\times2^{176+8}\times\frac{1}{9}\times\frac{1}{16}+4\times2^{192}\times[1+(1-2^{-16})+\cdots+(1-2^{-16})^{2\times90.8-159.42}]\times\frac{1}{9}\times\frac{1}{16}+2^{192}\times\mathsf{e}^{-2^{2\times90.8-175.42}}\times2^{136}\approx2^{242.5}$ 9-round AES-256 encryptions, recovering the entire 256-bit user key.

This is the first published attack on 9-round AES-256 using two keys.

5.7 Summary

In this chapter we have presented impossible differential cryptanalyses of 7-round AES-128, 7-round AES-192 and 8-round AES-256, extending the results given in [2, 11, 96]. We then present impossible boomerang attacks on 6-round AES-128, 7-round AES-192 and 7-round AES-256, and finally we present related-key impossible boomerang attacks on 8-round AES-192 and 9-round AES-256 in a related-key attack scenario using two keys. Table 5.3 summarises the published cryptanalytic results on AES, where CP, ACPC and RK-CP refer to the required numbers of chosen plaintexts, adaptive chosen plaintexts and ciphertexts and related-key chosen plaintexts, respectively; and Encryptions refers to the required number of encryption operations of the relevant reduced-round version of AES-128/192/256.

Note that the early abort technique can be used to improve certain cryptanalytic results on AES using related keys, such as the related-key truncated and impossible differential attacks on reduced AES-192 described by Jakimoski and Desmedt in [45].

Table 5.3: Cryptanalytic results on AES

Key Size	Attack Type	Rounds	Keys	Data	Time	Source
128	Square	6	1	2^{32}CP	2^{72}Encryptions	[21]
		7	1	$2^{128} - 2^{119}$CP	2^{120}Encryptions	[25]
	Collision	7	1	2^{32}CP	2^{128}Encryptions	[26]
	Boomerang	6	1	2^{71}ACPC	2^{71}Encryptions	[15]
	Impossible boomerang	6	1	$2^{112.2}$CP	$2^{112.3}$Encryptions	Section 5.5
	Impossible differential	5	1	$2^{29.5}$CP	2^{31}Encryptions	[11]
		6	1	$2^{91.5}$CP	2^{122}Encryptions	[16]
		7	1	$2^{117.5}$CP	2^{121}Encryptions	[2]
		7	1	$2^{115.5}$CP	2^{119}Encryptions	[111]
		7	1	$2^{112.2}$CP	$2^{115.6}$Encryptions	Section 5.4
192	Square	7	1	2^{32}CP	2^{184}Encryptions	[81]
		8	1	$2^{128} - 2^{119}$CP	2^{188}Encryptions	[25]
	Collision	7	1	2^{32}CP	2^{140}Encryptions	[26]
		7	1	2^{40}CP	2^{80}Encryptions	[22]
	Impossible boomerang	7	1	$2^{112.5}$CP	$2^{186.3}$Encryptions	Section 5.5
	Impossible differential	7	1	2^{92}CP	2^{186}Encryptions	[96]
		7	1	2^{92}CP	2^{162}Encryptions	[111]
		7	1	$2^{91.2}$CP	$2^{145.5}$Encryptions	Section 5.4
		7	1	$2^{113.8}$CP	$2^{117.2}$Encryptions	Section 5.4
	RK impossible differential	8	2	2^{88}RK-CP	2^{183}Encryptions	[45]
		8	2	2^{112}RK-CP	2^{136}Encryptions	[112]
	RK impossible boomerang	8	2	$2^{122.4}$RK-CP	2^{160}Encryptions	Section 5.6
	RK rectangle	8	4	$2^{86.5}$RK-CP	$2^{86.5}$Encryptions	[40]
		8	2	2^{94}RK-CP	2^{120}Encryptions	[52]
		9	64	2^{85}RK-CP	2^{182}Encryptions	[52]
		10	256	2^{125}RK-CP	2^{182}Encryptions	[52]
		10	64	2^{124}RK-CP	2^{183}Encryptions	[52]
256	Square	7	1	2^{32}CP	2^{200}Encryptions	[81]
		8	1	$2^{128} - 2^{119}$CP	2^{204}Encryptions	[25]
	Collision	7	1	2^{32}CP	2^{140}Encryptions	[26]
		8	1	2^{40}CP	2^{208}Encryptions	[22]
	Impossible boomerang	7	1	$2^{112.8}$CP	$2^{186.9}$Encryptions	Section 5.5
	Impossible differential	7	1	$2^{92.5}$CP	$2^{250.5}$Encryptions	[96]
		8	1	$2^{116.5}$CP	$2^{247.5}$Encryptions	[111]
		8	1	2^{89}CP	$2^{247.7}$Encryptions	Section 5.4
		8	1	$2^{111.6}$CP	$2^{233.1}$Encryptions	Section 5.4
	RK square	9	256	2^{85}RK-CP	$2^{226.4}$Encryptions	[25]
	RK impossible boomerang	9	2	$2^{122.8}$RK-CP	$2^{242.5}$Encryptions	Section 5.6
	RK rectangle	9	4	2^{99}RK-CP	2^{120}Encryptions	[52]
		10	256	$2^{114.9}$RK-CP	$2^{171.8}$Encryptions	[9]
		10	64	$2^{113.9}$RK-CP	$2^{172.8}$Encryptions	[52]

Chapter 6

Impossible Differential Cryptanalysis of Reduced Camellia

Camellia is a 128-bit block cipher with a user key of 128, 192 or 256 bits, which became a CRYPTREC-recommended e-government cipher in 2002, a NESSIE selected algorithm in 2003, and was adopted as an ISO international standard in 2005. In this chapter we present impossible differential attacks on 11-round Camellia-128 without the FL functions, 12-round Camellia-192 without the FL functions, and 13-round Camellia-256 without the FL functions, all of which use the early abort technique. The 11-round Camellia-128 attack requires 2^{118} *chosen plaintexts and has a time complexity of* 2^{118} *encryptions and* 2^{126} *memory accesses; the 12-round Camellia-192 attack requires* 2^{119} *chosen plaintexts and has a time complexity of* $2^{147.3}$ *encryptions; and the 13-round Camellia-256 attack requires* 2^{120} *chosen plaintexts and has a time complexity of* $2^{211.7}$ *encryptions. These are better than any previously published cryptanalytic results on Camellia without the FL functions.*

Contents

6.1 Introduction

The block cipher Camellia was designed by Aoki, Ichikawa, Kanda, Matsui, Moriai, Nakajima and Tokita [1], and published in 2000. Camellia has a Feistel structure, a 128-bit block length, and a user key length of 128, 192 or 256 bits. It became a CRYPTREC-recommended e-government cipher in 2002, a NESSIE selected algorithm in 2003, and was adopted as an ISO international standard in 2005.

In this chapter we present impossible differential attacks on 11-round Camellia-128 without the FL functions, 12-round Camellia-192 without the FL functions, and 13-round Camellia-256 without the FL functions, all of which use the early abort technique. The attack on 11-round Camellia-128 requires 2^{118} chosen plaintexts and has a time complexity of about 2^{118} encryptions and 2^{126} memory accesses; the attack on 12-round Camellia-192 requires 2^{119} chosen plaintexts and has a time complexity of about $2^{147.3}$ encryptions; and the attack on 13-round Camellia-256 requires 2^{120} chosen plaintexts and has a time complexity of about $2^{211.7}$ encryptions. These are better than any previously published cryptanalytic results on Camellia without the FL functions.

The remainder of this chapter is organised as follows. In Section 6.2 we describe Camellia. In Section 6.3 we briefly review previous cryptanalytic results on Camellia. In Section 6.4, we describe the 8-round impossible differentials for Camellia of Wu et al. [108]. In Sections 6.5, 6.6 and 6.7 we present our cryptanalytic results. Section 6.8 summarises the results given in this chapter.

6.2 The Camellia Block Cipher

In this section we briefly describe the Camellia block cipher [1].

6.2.1 Notation

In this chapter, a 128-bit value is represented as a sequence of sixteen bytes, numbered from 1 to 16 from left to right. We use the following notation.

- $\star$: an arbitrary 8-bit value, where two values represented by the $\star$ symbol may be different

6.2.2 Functions

Camellia uses the following five functions.

- $\mathbf{S} : \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ is a non-linear substitution constructed by applying eight 8×8-bit S-boxes $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3, \mathbf{S}_4, \mathbf{S}_5, \mathbf{S}_6, \mathbf{S}_7$ and $\mathbf{S}_8$ in parallel to the input, where $\mathbf{S}_1$ and $\mathbf{S}_8$ are identical, $\mathbf{S}_2$ and $\mathbf{S}_5$ are identical, $\mathbf{S}_3$ and $\mathbf{S}_6$ are identical, and $\mathbf{S}_4$ and $\mathbf{S}_7$ are identical. See [1] for specifications of the S-boxes.
- $\mathbf{P} : GF(2^8)^8 \rightarrow GF(2^8)^8$ is a linear permutation equivalent to multiplication by the following matrix:

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

- $\mathbf{F} : \{0,1\}^{64} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ is a Feistel function. If X and Y are 64-bit blocks, $\mathbf{F}(X,Y) = \mathbf{P}(\mathbf{S}(X \oplus Y))$.

- $\mathbf{FL} : \{0,1\}^{64} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ and $\mathbf{FL}^{-1} : \{0,1\}^{64} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ are key-dependent linear functions. As we consider the version of Camellia without the $\mathbf{FL}$ or $\mathbf{FL}^{-1}$ functions, we omit the description of these two functions; see [1] for specifications.

6.2.3 Generation of Subkeys

The Camellia cipher uses a total of four 64-bit whitening subkeys KW_j, $2\lfloor\frac{N_r-6}{6}\rfloor$ 64-bit subkeys KI_l for the $\mathbf{FL}$ and $\mathbf{FL}^{-1}$ functions, and N_r 64-bit round subkeys K_i, $(1 \leq j \leq 4, 1 \leq l \leq 2\lfloor\frac{N_r-6}{6}\rfloor, 1 \leq i \leq N_r)$, all derived from a N_k-bit key K, where N_r is 18 for Camellia-128, and 24 for Camellia-192/256 (i.e. the 128, 192 and 256-bit key versions of Camellia), N_k is 128 for Camellia-128, 192 for Camellia-192, and 256 for Camellia-256. How this derivation is performed is not of significance to our attacks, and so we do not describe it here (for details see [1]).

Each of the round subkeys K_i consists of 8 bytes; we write $K_{i,l}$ for the lth byte of K_i, where $1 \leq l \leq 8$.

6.2.4 Encryption Procedure

Camellia takes as input a 128-bit plaintext block P, and has a total of N_r rounds, where N_r is 18 for Camellia-128, and 24 for Camellia-192/256. The encryption procedure is, where L^0, R^0, L^i, R^i, L'^i and R'^i are 64-bit variables.

```
1. $L^0||R^0 = P \oplus (KW_1||KW_2)$
2. For $i = 1$ to $N_r$:
     if $i = 6$ or 12 (or 18 for Camellia-192/256),
         $L'^i = \mathbf{F}(L^{i-1}, K_i) \oplus R^{i-1}$, $R'^i = L^{i-1}$;
         $L^i = \mathbf{FL}(L'^i, KI_{\frac{i}{3}-1})$, $R^i = \mathbf{FL}^{-1}(R'^i, KI_{\frac{i}{3}})$;
     else
         $L^i = \mathbf{F}(L^{i-1}, K_i) \oplus R^{i-1}$, $R^i = L^{i-1}$;
```

3. Ciphertext = $(R^{N_r} \oplus KW_3)||(L^{N_r} \oplus KW_4)$.

The ith iteration of Step 2 in the above description is referred to below as Round i, $(1 \leq i \leq N_r)$.

6.3 Previous Cryptanalytic Results

In this section we briefly review previously published cryptanalytic attacks on Camellia.

- In 2001, He and Qing [35] presented a square attack on 6-round Camellia-128 without the **FL** functions.
- In 2001, Sugita, Kobara, and Imai [102] described an impossible differential attack on 7-round Camellia-128 without the **FL** functions.
- In 2001, Lee, Hong, Lee, Lim, and Yoon [71] presented a truncated differential attack on 8-round Camellia-128 without the **FL** functions.
- In 2002, Shirai [99] presented a boomerang attack on 9-round Camellia-192/256 with the **FL** functions, a rectangle attack on 10-round Camellia-256 with the **FL** functions, a differential attack on 11-round Camellia-256 without the **FL** functions, and a linear attack on 12-round Camellia-256 without the **FL** functions.
- In 2002, Yeom, Park, and Kim [109] presented a square attack on 9-round Camellia-256 with the **FL** functions.
- In 2002, Hatano, Sekine, and Kaneko [33] presented higher-order differential attacks on 11-round Camellia-256 both with and without the **FL** functions.
- In 2003, Yeom, Park, and Kim [110] presented an integral attack on 9-round Camellia-256 with the **FL** functions.
- In 2004, Wu, Feng, and Chen [107] presented collision attacks on 9-round Camellia-192/256 without the **FL** functions and 10-round Camellia-256 without the **FL** functions.

- In 2005, Duo, Li, and Feng [24] presented square attacks on 10-round Camellia-192/256 without the **FL** functions and 11-round Camellia-256 without the **FL** functions.
- In 2007, Wu, Zhang, and Feng [108] presented an impossible differential attack on 12-round Camellia-192/256 without the **FL** functions.

In summary, the best previously published cryptanalytic results on Camellia without the **FL** functions are the truncated differential attack on 8-round Camellia-128 [71], the impossible differential attack on 12-round Camellia-192 [108], and the linear and impossible differential cryptanalysis of 12-round Camellia-256 [99, 108].

6.4 8-Round Impossible Differentials of Camellia

In 2007, Wu et al. [108] gave the following 8-round impossible differentials for Camellia: $(0, 0, 0, 0, 0, 0, 0, 0, a, 0, 0, 0, 0, 0, 0, 0) \nrightarrow (h, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$, where a and h are any non-zero bytes.

These impossible differentials apply to any set of eight consecutive rounds of Camellia.

6.5 Attacking 13-Round Camellia-256 without the FL Functions

In this section, we present an impossible differential cryptanalysis of 13-round Camellia-256. Without loss of generality, we assume that the attacked 13 rounds are Rounds 1 to 13, and use the 8-round impossible differentials of Wu et al. applied to Rounds 4 to 11.

6.5.1 Preliminary Results

It is easy to verify by a computer program that, for every S-box of Camellia, there exist 127 possible output differences for any non-zero input difference, of which 1 output difference occurs with probability 2^{-6}, and each of the other 126 output differences occurs with probability 2^{-7}. Thus an output difference $(h,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$ of the 8-round impossible differentials will propagate to about 2^7 possible output differences $(g,g,g,0,g,0,0,g,h,0,0,0,0,0,0,0)$ after Round 12, where g is non-zero. Then, every $(g,g,g,0,g,0,0,g,h,0,0,0,0,0,0,0)$ will propagate to about $(2^7)^5$ possible output differences after Round 13. Hence, there are at most $(2^8-1)\times 2^7\times(2^7)^5\approx 2^{50}$ possible output differences after Round 13; let Ω_{13} be the set of all possible output differences after Round 13.

We use the early abort technique in the first two rounds and the last round of the 13-round attack. We first give the following result.

Property 6.1 *The following properties hold.*

1. *If $(P_i=(L_i^0,R_i^0), P_j=(L_j^0,R_j^0))$ is a plaintext pair, then $\mathbf{P}^{-1}(R_i^0\oplus R_j^0\oplus(u,u,u,0,u,0,0,u))$ has a unique value in the first two bytes for every non-zero value of u (one byte long).*
2. *For a pair of ciphertexts (C_i,C_j), if their corresponding values just after Round 13 have a difference $(\Delta L^{13}=L_i^{13}\oplus L_j^{13}, \Delta R^{13}=R_i^{13}\oplus R_j^{13})$ belonging to Ω_{13}, then the difference between their corresponding values just after the S-box substitution layer of Round 13 must have the form $(\star,\star,\star,0,\star,0,0,\star)$, and there is a unique value of h such that $\mathbf{P}^{-1}(L_i^{13}\oplus L_j^{13}\oplus(h,0,0,0,0,0,0,0))$ has the form $(\star,\star,\star,0,\star,0,0,\star)$.*

Proof. (1) Suppose that there are two values u_1 and u_2 such that $\mathbf{P}^{-1}(R_i^0\oplus R_j^0\oplus(u_1,u_1,u_1,0,u_1,0,0,u_1))\oplus\mathbf{P}^{-1}(R_i^0\oplus R_j^0\oplus(u_2,u_2,u_2,0,u_2,0,0,u_2))=(0,0,\star,\star,\star,\star,\star,\star)$. Thus $\mathbf{P}^{-1}(u_1\oplus u_2,u_1\oplus u_2,u_1\oplus u_2,0,u_1\oplus u_2,0,0,u_1\oplus u_2)=(0,0,\star,\star,\star,\star,\star,\star)$. By definition of $\mathbf{P}^{-1}$ it follows that the first byte of $\mathbf{P}^{-1}(x,x,x,0,x,0,0,x)$ for any x is equal to x, and hence $u_1\oplus u_2=0$, i.e. $u_1=u_2$, and (1) follows.

(2) The left half of a difference from Ω_{13} has the form $(g, g, g, 0, g, 0, 0, g)$, where g is a non-zero byte value; thus, for a pair of ciphertexts (C_i, C_j) such that their corresponding values just after Round 13 have a difference belonging to Ω_{13}, the difference between their corresponding values just after the S-box substitution layer of Round 13 must have the form $(\star, \star, \star, 0, \star, 0, 0, \star)$. We now prove the latter part of Property 6.1-2. Because any difference from Ω_{13} is obtained given the input difference $(g, g, g, 0, g, 0, 0, g, h, 0, 0, 0, 0, 0, 0, 0)$ to Round 13, therefore, for a pair of ciphertexts (C_i, C_j) such that their corresponding values just after Round 13 have a difference belonging to Ω_{13}, there must be a value of h such that $\mathbf{P}^{-1}(L_i^{13} \oplus L_j^{13} \oplus (h, 0, 0, 0, 0, 0, 0, 0))$ has the form $(\star, \star, \star, 0, \star, 0, 0, \star)$. Assume there are two different values h_1 and h_2 that satisfy the condition, then it follows that $\mathbf{P}^{-1}((h_1, 0, 0, 0, 0, 0, 0, 0) \oplus (h_2, 0, 0, 0, 0, 0, 0, 0))$ also has the form $(\star, \star, \star, 0, \star, 0, 0, \star)$; note that the fourth byte is 0; however, by definition of $\mathbf{P}^{-1}$ it follows that the fourth byte of $\mathbf{P}^{-1}((h_1, 0, 0, 0, 0, 0, 0, 0) \oplus (h_2, 0, 0, 0, 0, 0, 0, 0))$ should be $h_1 \oplus h_2 \neq 0$, giving a contradiction. □

6.5.2 Attack Description

We now present a procedure for attacking 13-round Cammellia-256; it involves the following series of steps. The attack is shown diagrammatically in Figure 6.1.

1. Choose 2^8 structures S_i, $(1 \leq i \leq 2^8)$, where a structure S_i is defined to be a set of 2^{112} plaintexts $P_{i,j} = (L_{i,j}^0, R_{i,j}^0)$ with $L_{i,j}^0 = \mathbf{P}(x_1^{i,j}, x_2^{i,j}, x_3^{i,j}, \alpha_4, x_5^{i,j}, \gamma_6^i, \gamma_7^i, x_8^{i,j}) \oplus (x^{i,j}, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8)$ and $R_{i,j}^0 = (y_1^{i,j}, y_2^{i,j}, y_3^{i,j}, y_4^{i,j}, y_5^{i,j}, y_6^{i,j}, y_7^{i,j}, y_8^{i,j})$, where the bytes $\alpha_4, \beta_2, \beta_3, \cdots, \beta_8$ are arbitrary but fixed values (for the 2^8 structures), the bytes $x^{i,j}, x_1^{i,j}, x_2^{i,j}, x_3^{i,j}, x_5^{i,j}, x_8^{i,j}, y_1^{i,j}, y_2^{i,j}, \cdots, y_8^{i,j}$ take all the possible values in $\{0, 1\}^8$, and the bytes γ_6^i, γ_7^i are fixed, $(j = 1, 2, \cdots, 2^{112})$. In a chosen-plaintext attack scenario, obtain all the 2^{120} ciphertexts for all the 2^{112} plaintexts in each of the 2^8 structures; we denote the ciphertext for plaintext $P_{i,j}$ by $C_{i,j} = (L_{i,j}^{13}, R_{i,j}^{13})$. Choose the pairs (C_{i,j_1}, C_{i,j_2}) with a difference belonging to Ω_{13}, where $1 \leq j_1 \neq j_2 \leq 2^{112}$.

2. For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), by Property 6.1-2 there is only one value of h such that $\mathbf{P}^{-1}(L_{i,j_1}^{13} \oplus L_{i,j_2}^{13} \oplus (h, 0, 0, 0, 0, 0, 0, 0))$ has the form $(\star, \star, \star, 0, \star, 0, 0, \star)$; we denote by δ_{i,j_1,j_2}^{13} the value $\mathbf{P}^{-1}(L_{i,j_1}^{13} \oplus L_{i,j_2}^{13} \oplus$

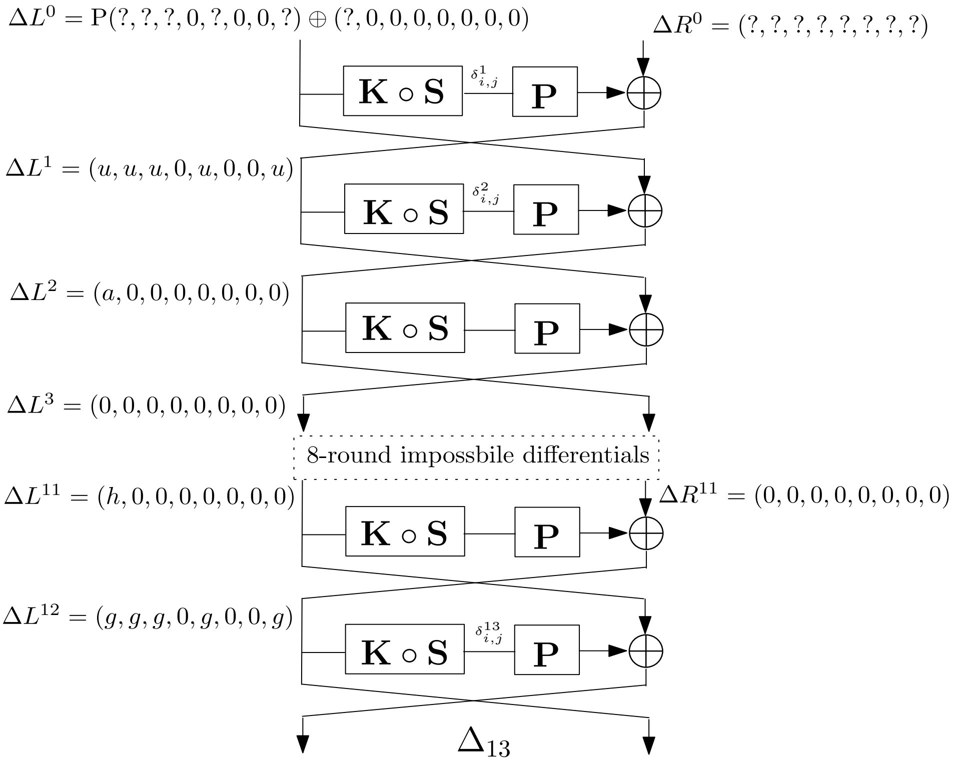

$\Delta L^0 = \mathrm{P}(?,?,?,0,?,0,0,?) \oplus (?,0,0,0,0,0,0,0)$
$\Delta R^0 = (?,?,?,?,?,?,?,?)$
$\mathbf{K} \circ \mathbf{S}$
$\delta^1_{i,j}$
$\mathbf{P}$
$\Delta L^1 = (u,u,u,0,u,0,0,u)$
$\delta^2_{i,j}$
$\Delta L^2 = (a,0,0,0,0,0,0,0)$
$\Delta L^3 = (0,0,0,0,0,0,0,0)$
8-round impossbile differentials
$\Delta L^{11} = (h,0,0,0,0,0,0,0)$
$\Delta R^{11} = (0,0,0,0,0,0,0,0)$
$\Delta L^{12} = (g,g,g,0,g,0,0,g)$
$\delta^{13}_{i,j}$
Δ_{13}

Figure 6.1: Impossible differential attack on 13-round Camellia-256

$(h,0,0,0,0,0,0,0)$) with the form $(\star,\star,\star,0,\star,0,0,\star)$. Then, perform Steps (a) and (b) below.

(a) Perform the following two sub-steps for $l = 1,2,3,5,8$.

- Guess a value for the subkey byte $K_{13,l}$;
- For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt R^{13}_{i,j_1} and R^{13}_{i,j_2} to get the corresponding values for byte (l) just after the **S** function of Round 13, and check whether they have a difference equal to the corresponding one-byte difference in δ^{13}_{i,j_1,j_2}. Keep only the pairs that meet this condition.

(b) Guess a value for the subkey bytes $(K_{13,4}, K_{13,6}, K_{13,7})$, such that for every remaining ciphertext pair we can get the corresponding values for byte (1) just before Round 12.

3. Guess a value for the subkey byte $K_{12,1}$. For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt R^{12}_{i,j_1} and R^{12}_{i,j_2} to get the corresponding values for byte (1) just after the **S** function of Round 12, and check whether they

have a difference equal to byte (1) of $L^{12}_{i,j_1} \oplus L^{12}_{i,j_2}$. Keep only the pairs that meet this condition.

4. For every plaintext pair (P_{i,j_1}, P_{i,j_2}) corresponding to a remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), compute $\mathbf{P}^{-1}(R^0_{i,j_1} \oplus R^0_{i,j_2} \oplus (u,u,u,0,u,0,0,u))$ for all the 255 possible non-zero values of u; label the resulting set of 255 values Δ^1_{i,j_1,j_2}. Then, perform Steps (a) and (b) below.

   (a) Guess a value for the two subkey bytes $(K_{1,1}, K_{1,2})$. For every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}), partially encrypt L^0_{i,j_1} and L^0_{i,j_2} to get the corresponding values for bytes (1,2) just after the **S** function of Round 1, and check whether they have a difference equal to any of the corresponding two-byte partial differences in Δ^1_{i,j_1,j_2}. Keep only the pairs that meet this condition. By Property 6.1-1 there is only one difference in Δ^1_{i,j_1,j_2} for a pair meeting the condition, and we denote this difference from Δ^1_{i,j_1,j_2} by δ^1_{i,j_1,j_2}.

   (b) Perform the following two sub-steps for $l = 3$ to 8:

   - Guess a value for the subkey byte $K_{1,l}$;
   - For every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}), partially encrypt L^0_{i,j_1} and L^0_{i,j_2} to get the corresponding values for byte (l) just after the **S** function of Round 1, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^1_{i,j_1,j_2}. Keep only the pairs that meet this condition.

5. For every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}), from Property 6.1-2 we similarly know that there is only one value of a such that $\mathbf{P}^{-1}(L^0_{i,j_1} \oplus L^0_{i,j_1} \oplus (a,0,0,0,0,0,0,0))$ has the form $(\star,\star,\star,0,\star,0,0,\star)$; we denote by δ^2_{i,j_1,j_2} the value $\mathbf{P}^{-1}(L^0_{i,j_1} \oplus L^0_{i,j_1} \oplus (a,0,0,0,0,0,0,0))$ with the form $(\star,\star,\star,0,\star,0,0,\star)$. Then, perform Steps (a) and (b) below.

   (a) Perform the following two sub-steps for $l = 1,2,3,5,8$.

   - Guess a value for the subkey byte $K_{2,l}$;
   - For every remaining pair (P_{i,j_1}, P_{i,j_2}), partially encrypt L^1_{i,j_1} and L^1_{i,j_2} to get the corresponding values for byte (l) just after the **S** function of Round 2, and check whether they have a difference equal to the corresponding one-byte partial difference in δ^2_{i,j_1,j_2}. Keep only the pairs that meet this condition.

(b) Guess a value for the subkey bytes $(K_{2,4}, K_{2,6}, K_{2,7})$, such that for every remaining plaintext pair we can get the corresponding values for byte (1) just after Round 2.

6. Guess a value for the subkey byte $K_{3,1}$. For every plaintext pair (P_{i,j_1}, P_{i,j_2}), partially encrypt L^2_{i,j_1} and L^2_{i,j_2} to get the corresponding values for byte (1) just after the **S** function of Round 3, and check whether they have a difference equal to byte (1) of $L^1_{i,j_1} \oplus L^1_{i,j_2}$. If there exists a ciphertext pair that meets this condition, then discard this subkey guess, and try another; otherwise, for every subkey guessed value for (K_1, K_2), exhaustively search for the remaining 128 key bits.

6.5.3 Complexity Analysis

The attack requires 2^{120} chosen plaintexts, which take a time complexity of 2^{120} 13-round Camellia-256 encryptions.

In Step 1, after an analysis we learn that, for different values of $(x^{i,j}, x_1^{i,j}, x_2^{i,j}, x_3^{i,j}, x_5^{i,j}, x_8^{i,j}, y_1^{i,j}, \cdots, y_8^{i,j})$ in a structure S_i, the resultant 128-bit blocks are different. Thus a structure S_i yields $\binom{2^{112}}{2} \approx \frac{2^{112\times 2}}{2} = 2^{223}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}), $(j = 1, 2, \cdots, 2^{112})$, and hence the 2^8 structures yield a total of 2^{231} ciphertext pairs. Choosing the pairs (C_{i,j_1}, C_{i,j_2}) with a difference belonging to Ω_{13} requires about $2^{120} \times 2^{50} = 2^{176}$ memory accesses in a simple implementation. There are 2^{50} possible differences in Ω_{13}, thus approximately $2^{231} \times \frac{2^{50}}{2^{128}} = 2^{153}$ are chosen in Step 1.

In Step 2(a), a proportion of about $1 - 2^{-7}$ of the remaining ciphertext pairs will be discarded after every iteration. Step 2(b) does not put any filtering condition on the remaining ciphertext pairs. Step 2 has a total time complexity of about $\sum_{i=0}^{4}(2 \times 2^{153-7\times i} \times 2^{8\times(i+1)} \times \frac{1}{13} \times \frac{1}{8}) + 2 \times 2^{118} \times 2^{64} \times \frac{1}{13} \times \frac{3}{8} \approx 2^{177.9}$ 13-round Camellia-256 decryptions.

In Step 3, a proportion of about $1 - 2^{-7}$ of the remaining ciphertext pairs will be discarded. Step 3 has a time complexity of about $2 \times 2^{118} \times 2^{72} \times \frac{1}{13} \times \frac{1}{8} \approx 2^{184.3}$ 13-round Camellia-256 decryptions.

In Step 4(a), there are 255 possible values in Δ^1_{i,j_1,j_2} for every pair (P_{i,j_1}, P_{i,j_2}), thus it is expected that about $2^{111} \times \frac{255}{2^{16}} \approx 2^{103}$ pairs (P_{i,j_1}, P_{i,j_2}) remain after Step 4(a) for every guess of $(K_{13,1}, K_{13,2}, K_{13,3}, K_{13,5}, K_{13,8}, K_{12,1}, K_{1,1}, K_{1,2})$. In Step 4(b), the difference δ^1_{i,j_1,j_2} is already fixed in Step 4(a), so it is expected that a proportion of about $1 - 2^{-8}$ of the remaining pairs (P_{i,j_1}, P_{i,j_2}) will be discarded after every iteration. Step 4 has a total time complexity of about $2 \times 2^{111} \times 2^{88} \times \frac{1}{13} \times \frac{2}{8} + \sum_{i=0}^{5}(2 \times 2^{103-8\times i} \times 2^{88+8\times(i+1)} \times \frac{1}{13} \times \frac{1}{8}) \approx 2^{196.3}$ 13-round Camellia-256 encryptions.

In Step 5(a), similarly it is expected that a proportion of about $1 - 2^{-8}$ of the remaining plaintext pairs (P_{i,j_1}, P_{i,j_2}) will be discarded after every iteration. Step 5(b) does not put any filtering condition on the remaining plaintext pairs. Step 5 has a total time complexity of about $\sum_{i=0}^{4}(2 \times 2^{55-8\times i} \times 2^{136+8\times(i+1)} \times \frac{1}{13} \times \frac{1}{8}) + 2 \times 2^{15} \times 2^{200} \times \frac{1}{13} \times \frac{3}{8} \approx 2^{210.9}$ 13-round Camellia-256 encryptions.

In Step 6, with a probability of 2^{-8} we can get a pair (C_{i,j_1}, C_{i,j_2}) that meets the condition, thus the expected number of remaining subkey guesses is about $2^{208} \times (1 - 2^{-8})^{2^{15}} \approx 2^{23.68}$, meaning that $2^{151.68}$ trial encryptions are required to find the correct 256 key bits. Step 6 has a time complexity of about $2 \times 2^{208} \times [1 + (1 - 2^{-8}) + \cdots + (1 - 2^{-8})^{2^{15}}] \times \frac{1}{13} \times \frac{1}{8} + 2^{151.68} \approx 2^{210.3}$ 13-round Camellia-256 encryptions.

Therefore, the attack has a total time complexity of approximately $2^{211.7}$ 13-round Camellia-256 encryptions.

6.6 Attacking 12-Round Camellia-192 without the FL Functions

As mentioned earlier, Wu et al. [108] presented an impossible differential cryptanalysis on 12-round Camellia-192 without the **FL** functions. The attack requires 2^{120} chosen plaintexts, and has a time complexity of 2^{181} Camellia-192 encryptions. However, it can be improved; the improved attack is basically the version of the above 13-round Camellia-256 attack when the last round is removed. The main difference is that in the last step we exhaustively search for the remaining 64 key bits for every guessed value for (K_1, K_2). After a similar analysis, we get that the improved attack on 12-round Camellia-192 requires 2^{119} chosen plaintexts, and has a time complexity of approximately $2^{147.3}$ 12-round Camellia-192 encryptions;

6.7 Attacking 11-Round Camellia-128 without the FL Functions

Without loss of generality, we assume that the attacked 11 rounds are Rounds 1 to 11. We use the 8-round impossible differentials of Wu et al. in Rounds 3 to 10, and use the early abort technique in the first round.

6.7.1 Attack Description

The attack procedure is as follows.

1. Choose 2^{30} structures S_i, $(i = 1, 2, \cdots, 2^{30})$, where a structure is defined to be a set of 2^{88} plaintexts $P_i = (L^0_{i,j}, R^0_{i,j})$ with $R^0_{i,j} = \mathbf{P}(x_1^{i,j}, x_2^{i,j}, x_3^{i,j}, \alpha_4, x_5^{i,j}, \sigma_6^i, \sigma_7^i, x_8^{i,j}) \oplus (x^{i,j}, \beta_2^i, \beta_3^i, \beta_4^i, \beta_5^i, \beta_6^i, \beta_7^i, \beta_8^i)$ and $L^0_{i,j} = (y_1^{i,j}, y_2^{i,j}, y_3^{i,j}, \gamma_4^i, y_5^{i,j}, \gamma_6^i, \gamma_7^i, y_8^{i,j})$, where the bytes $\alpha_4, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8$ are arbitrary but fixed values (for the 2^{30} structures), the bytes $x^{i,j}, x_1^{i,j}, x_2^{i,j}, x_3^{i,j}, x_5^{i,j}, x_8^{i,j}, y_1^{i,j}, y_2^{i,j}, y_3^{i,j}, y_5^{i,j}, y_8^{i,j}$ take all the possible values in $\{0,1\}^8$, and the bytes $\sigma_6^i, \sigma_7^i, \gamma_4^i, \gamma_6^i, \gamma_7^i$ are fixed, $(j = 1, 2, \cdots, 2^{88})$. In a chosen-plaintext attack scenario, obtain all the 2^{118} ciphertexts for the 2^{88} plaintexts in each of the 2^{30} structures; we denote the ciphertext for plaintext $P_{i,j}$ by $C_{i,j} = (L^{11}_{i,j}, R^{11}_{i,j})$. Choose the ciphertext pairs (C_{i,j_1}, C_{i,j_2}) such that $L^0_{i,j_1} \oplus L^0_{i,j_2} = (u, u, u, 0, u, 0, 0, u)$ and $(L^{11}_{i,j_1} \oplus L^{11}_{i,j_2}, R^{11}_{i,j_1} \oplus R^{11}_{i,j_2})$ belonging to the 2^{15} possible output differences after Round 11.

2. Conduct a step similar to Step 3 of the 13-round Camellia-256 attack presented in Section 6.5.

3. Conduct a step similar to Step 5 of the 13-round Camellia-256 attack.

4. Conduct a step similar to Step 6 of the 13-round Camellia-256 attack; here, for every remaining guess for $(K_1, K_{2,1})$, exhaustively search for the remaining 56 key bits.

6.7.2 Complexity Analysis

In Step 1, a structure yields about $\frac{2^{88\times 2}}{2} \times \frac{255}{2^{40}} \approx 2^{143}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) with $L^0_{i,j_1} \oplus L^0_{i,j_2} = (u,u,u,0,u,0,0,u)$, so the 2^{30} structures yield a total of 2^{173} ciphertext pairs with $\Delta L^0 = (u,u,u,0,u,0,0,u)$, which generate $2^{173} \times \frac{2^{15}}{2^{128}} = 2^{60}$ useful pairs. To get the useful ciphertext pairs, we first store the ciphertexts in a structure S_i into a hash table indexed by bytes (4,6,7) of $L^{11}_{i,j}$, bytes $(2,3,\cdots,8)$ of $R^{11}_{i,j}$, the XOR of the 1st and 2-nd bytes of $L^{11}_{i,j}$, the XOR of the 1st and 3-rd bytes of $L^{11}_{i,j}$, the XOR of the 1st and 5th bytes of $L^{11}_{i,j}$ and the XOR of the 1st and 8th bytes of $L^{11}_{i,j}$; and then we choose the qualified pairs. Thus, it requires about $2^{118} \times 2^8 = 2^{126}$ memory accesses in a simple implementation. The expected number of remaining ciphertext pairs is about 2^{60}.

Step 2 has a time complexity of about $2 \times 2^{60} \times 2^8 \times \frac{1}{11} \times \frac{1}{8} \approx 2^{62.6}$ 11-round Camellia-128 encryptions.

Step 3 has a time complexity of about $\sum_{i=0}^{4}(2 \times 2^{53-8\times i} \times 2^{8+8\times(i+1)} \times \frac{1}{11} \times \frac{1}{8}) + 2 \times 2^{13} \times 2^{72} \times \frac{1}{11} \times \frac{3}{8} \approx 2^{81.2}$ 11-round Camellia-128 encryptions.

In Step 4, it is expected that there remain about $2^{80} \times (1-2^{-8})^{2^{13}} \approx 2^{33.92}$ guesses for $(K_1, K_{2,1}, K_{11,1})$; thus $2^{89.92}$ trial encryptions are required to find the 128 key bits. This step has a time complexity of about $2 \times 2^{80} \times [1 + (1-2^{-8}) + \cdots + (1-2^{-8})^{2^{13}}] \times \frac{1}{11} \times \frac{1}{8} + 2^{89.92} \approx 2^{90}$ 11-round Camellia-128 encryptions.

Therefore, the attack has a total time complexity of approximately 2^{118} 11-round Camellia-128 encryptions and 2^{126} memory accesses.

6.8 Summary

In this chapter we have presented impossible differential attacks on 11-round Camellia-128 without the **FL** functions, 12-round Camellia-192 without the **FL** functions, and 13-round Camellia-256 without the **FL** functions. Table 6.1 summarises the published cryptanalytic results on Camellia, where CP, KP and ACPC refer to the required numbers of chosen plaintexts, known plaintexts and adaptive chosen plaintexts and ciphertexts, respectively; MA and Encryptions refer to the required numbers of memory accesses and encryption operations of the relevant reduced version of Camellia-128/192/256, respectively; “none” means “no **FL** function”; and “all” means “all the **FL** functions”.

Table 6.1: Cryptanalytic results on Camellia

Key Size	Attack Type	Rounds	$\mathbf{FL}/\mathbf{FL}^{-1}$	Data	Time	Source
128	Square	6	none	$2^{11.7}$CP	2^{112}Encryptions	[35]
	Truncated differential	8	none	$2^{83.6}$CP	$2^{55.6}$Encryptions	[71]
	Impossible differential	7	none	not specified	not specified	[102]
		11	none	2^{118}CP	2^{126}MA&2^{118}Encryptions	Section 6.7
192	Boomerang	9	all	2^{124}ACPC	2^{170}Encryptions	[99]
	Collision	9	none	2^{13}CP	$2^{175.6}$Encryptions	[107]
	Square	10	none	not specified	2^{186}Encryptions	[24]
	Impossible differential	12	none	2^{120}CP	2^{181}Encryptions	[108]
		12	none	2^{119}CP	$2^{147.3}$Encryptions	Section 6.6
256	Boomerang	9	all	2^{124}ACPC	2^{170}Encryptions	[99]
	Square	9	all	2^{60}CP	2^{202}Encryptions	[109]
		10	none	not specified	2^{186}Encryptions	[24]
	Integral	9	all	$2^{60.5}$CP	$2^{202.2}$Encryptions	[110]
	Rectangle	10	all	2^{127}CP	2^{241}Encryptions	[99]
	Collision	10	none	2^{14}CP	$2^{239.9}$Encryptions	[107]
	Differential	11	none	2^{104}CP	2^{232}Encryptions	[99]
	High-order differential	11	none	2^{21}CP	2^{255}Encryptions	[33]
		11	all	2^{93}CP	2^{256}Encryptions	[33]
	Square	11	none	not specified	2^{250}Encryptions	[24]
	Linear	12	none	2^{119}KP	2^{247}Encryptions	[99]
	Impossible differential	12	none	2^{120}CP	2^{181}Encryptions	[108]
		13	none	2^{120}CP	$2^{211.7}$Encryptions	Section 6.5

Chapter 7

Related-Key Cryptanalysis of the Full Cobra-F64a and Cobra-F64b

Cobra-F64a and Cobra-F64b, designed for firmware-oriented applications, are 64-bit Data-dependent Permutation based block ciphers with 128 key bits, which involve 16 and 20 rounds, respectively. In this chapter, we present a related-key rectangle attack on the full Cobra-F64a and a related-key differential attack on the full Cobra-F64b. The attack on Cobra-F64a requires $2^{64.81}$ related-key chosen plaintexts, and has a time complexity of approximately $2^{123.81}$ encryptions; the attack on Cobra-F64b requires 2^{61} related-key chosen plaintexts, and has a time complexity of approximately $2^{110.67}$ encryptions.

Contents

7.1 Introduction

Cobra-F64a and Cobra-F64b was designed by Goots, Moldovyan, Moldovyan and Summerville [28], and published in 2003. They have a Feistel structure, a 64-bit block length, and a 128-bit user key, which involve 16 and 20 rounds, respectively.

Recently, a number of block ciphers, including SPECTR-H64 [29], the CIKS family — CIKS-1 [85], CIKS-128 [28] and CIKS-128H [100], and the Cobra family — Cobra-128, Cobra-F64a and Cobra-F64b [30], Cobra-H64 and Cobra-H128 [101], have been proposed for use in applications that require a small amount of data to be encrypted with frequently changed user keys. One example of such an application is provided by IPsec (Internet Protocol security) [44]. However, many of them have been shown to be vulnerable to related-key cryptanalytic attacks [62, 63, 68, 69], although Cobra-F64a and Cobra-F64b [30] have, until now, been exceptions.

In this chapter, we describe a 15-round related-key rectangle distinguisher with probability $2^{-123.62}$ for Cobra-F64a, and use it to mount a related-key rectangle attack on the full 16-round Cobra-F64a. The attack requires $2^{64.81}$ related-key chosen plaintexts and has a time complexity of approximately $2^{123.81}$ encryptions. We also describe a 19.5-round related-key differential with probability 2^{-57} for Cobra-F64b, and use it as the basis of a related-key differential attack on the full 20-round Cobra-F64b. The second attack requires 2^{61} related-key chosen plaintexts and has a time complexity of approximately $2^{110.67}$ encryptions.

The remainder of this chapter is organised as follows. In Section 7.2 we describe Cobra-F64a and Cobra-F64b. In Section 7.3 we briefly review previous cryptanalytic results on Cobra-F64a and Cobra-F64b. In Section 7.4 we give a number of properties of Cobra-F64a and Cobra-F64b. In Sections 7.5 and 7.6 we present our cryptanalytic results on Cobra-F64a and Cobra-F64b, respectively. Section 7.7 summarises the results given in this chapter.

7.2 Cobra-F64a and Cobra-F64b

In this section we briefly describe the Cobra-F64a and Cobra-F64b block ciphers [30].

7.2.1 Notation

In this chapter, the bits of an n-bit value are numbered from 1 to n from left to right, where the least significant bit is referred as the nth bit, and the most significant bit is referred as the 1st bit. We use the following notation.

- $\boxplus$: addition modulo 2^{32}
- $\boxminus$: subtraction modulo 2^{32}
- $\langle x \rangle_2$: x is in binary (base 2) notation

7.2.2 Functions and DDP-Boxes

Cobra-F64a and Cobra-F64b use the function $\mathbf{T}$ and a number of so-called DDP-boxes $\mathbf{P}_{n,m}$ (for specific values of n and m) to construct the round function $\mathbf{F}$. These functions are defined as follows.

- $\mathbf{T} : \{0,1\}^{32} \rightarrow \{0,1\}^{96}$ is a linear function. If $L = (l_1, \cdots, l_{32})$ is 32-bit block, then $\mathbf{T}(L)$ is defined to equal $(L_1 || L_1 \ggg 6 || L_1 \ggg 12 || L_2 || L_2 \ggg 6 || L_2 \ggg 12)$, where $L_1 = (l_1, \cdots, l_{16})$ and $L_2 = (l_{17}, \cdots, l_{32})$.
- For certain specific values of n and m (see below), the non-linear function $\mathbf{P}_{n,m} : \{0,1\}^n \times \{0,1\}^m \rightarrow \{0,1\}^n$ with the property that, for any fixed m-bit value V, $\mathbf{P}_{n,m}(\cdot, V) : \{0,1\}^n \rightarrow \{0,1\}^n$ is a bijective mapping. Such a function is called a Data-Dependent Permutation box (DDP-box), and V is called the controlling vector. We write $\mathbf{P}^{-1}_{n,m}(\cdot, V)$ as $(\mathbf{P}^{-1}_{n,m}(\cdot, V))^{-1}$ for any fixed V, or simply write $\mathbf{P}^{-1}_{n,m}$. Cobra-F64a and Cobra-F64b use the following DDP-boxes.

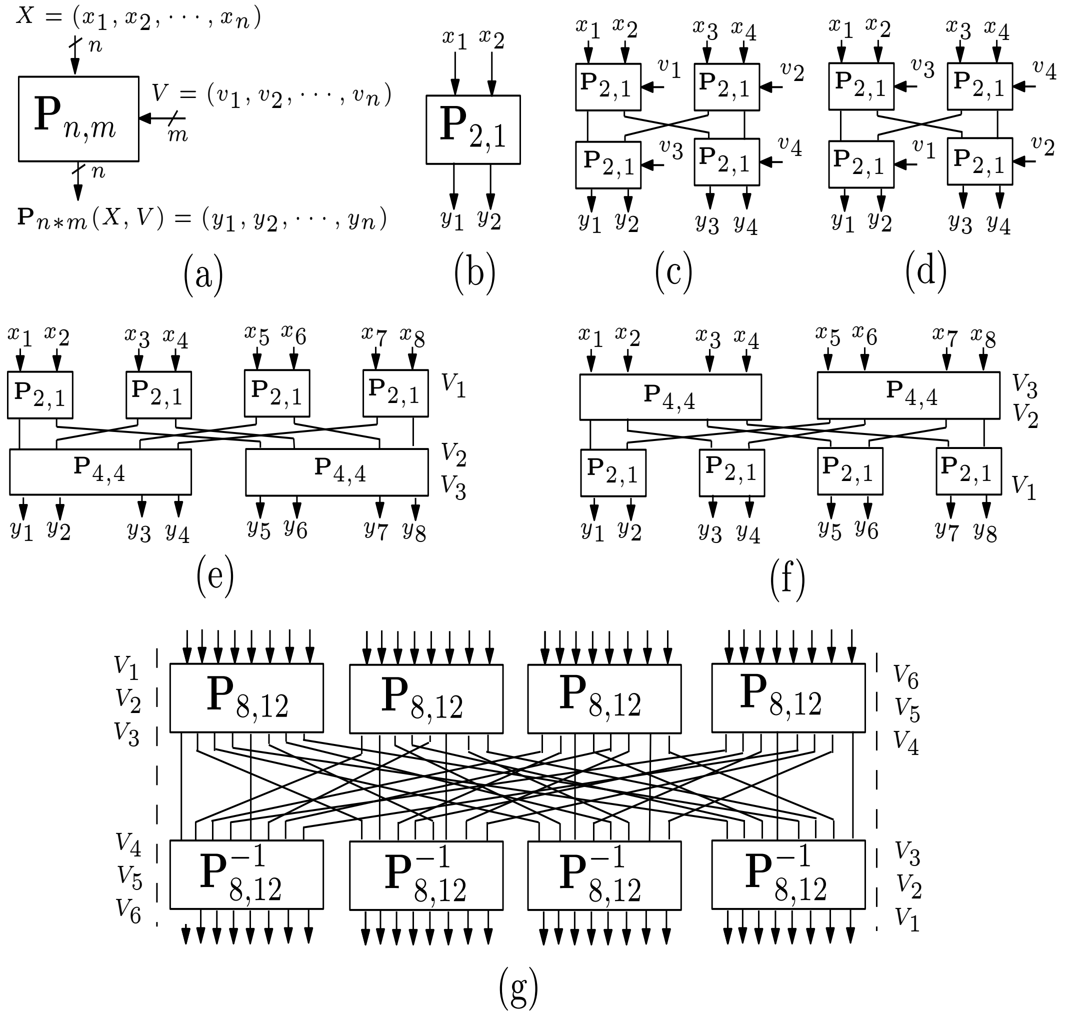


Figure 7.1: (a) $\mathbf{P}_{n,m}$; (b) $\mathbf{P}_{2,1}$; (c) $\mathbf{P}_{4,4}$; (d) $\mathbf{P}^{-1}_{4,4}$; (e) $\mathbf{P}_{8,12}$; (f) $\mathbf{P}^{-1}_{8,12}$; (g) $\mathbf{P}_{32,96}$ and $\mathbf{P}^{-1}_{32,96}$

- $\mathbf{P}_{2,1}$: If $x = (x_1, x_2) \in \{0,1\}^2$ and $v \in \{0,1\}$, $\mathbf{P}_{2,1}(x, v) = (x_{1+v}, x_{2-v})$. That is, $\mathbf{P}_{2,1}(\cdot, v)$ swaps the two input bits if $v = 1$; otherwise, it is the identity function.
- $\mathbf{P}_{4,4}$, $\mathbf{P}_{8,12}$, $\mathbf{P}_{32,96}$, and their inverses $\mathbf{P}^{-1}_{4,4}$, $\mathbf{P}^{-1}_{8,12}$ and $\mathbf{P}^{-1}_{32,96}$ are all defined using the 2 × 1 DDP-box $\mathbf{P}_{2,1}$. Figure 7.1 depicts these DDP-boxes. Detailed specifications of these functions are given in Goots et al. [30].
- The function $\mathbf{P}^{(\omega)}_{96,1}$: is defined in a series of $\mathbf{P}_{2,1}$ that use the same 1-bit 'control' input ω, as shown in Figure 7.2(a). $\omega = 0$ is used for the encryption function of Cobra-F64a or Cobra-F64b, and $\omega = 1$ is used for

decryption.

- The function $\mathbf{P}^{(\omega)}_{32,32}$ is defined as the functional composition of $\mathbf{T}$, followed by $\mathbf{P}^{(\omega)}_{96,1}$ and then $\mathbf{P}_{32,96}$, as shown in Figure 7.2(b).

- $\mathbf{F} : \{0,1\}^{64} \times \{0,1\}^{64} \rightarrow \{0,1\}^{64}$ is an non-linear Feistel structure. Figure 7.3 depicts $\mathbf{F}$ for Cobra-F64a and Cobra-F64b. Detailed specifications of these two functions are given in [30].

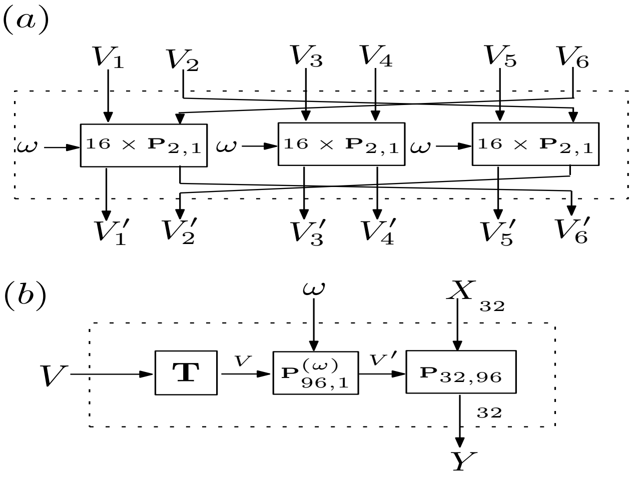


Figure 7.2: (a) $\mathbf{P}^{(\omega)}_{96,1}$; (b) $\mathbf{P}^{(\omega)}_{32,32}$

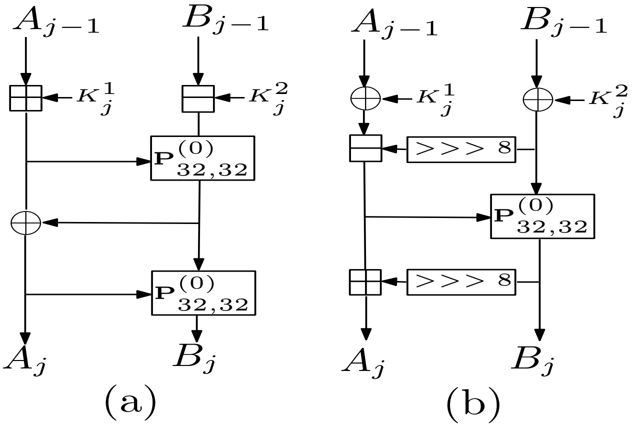


Figure 7.3: (a) $\mathbf{F}$ of Cobra-F64a; (b) $\mathbf{F}$ of Cobra-F64b

7.2.3 Generation of Subkeys

Cobra-F64a uses a total of 34 32-bit subkeys K_i^j $(1 \leq i \leq 17), j \in \{1,2\}$, all derived from a 128-bit user key K. Similarly, Cobra-F64b uses a total of 42 32-bit subkeys K_i^j $(1 \leq i \leq 21), j \in \{1,2\}$, all derived from a 128-bit user key K. Let K be represented as a sequence of as four 32-bit words $K = (W_1, W_2, W_3, W_4)$, then the subkeys of Cobra-F64a and Cobra-F64b are generated as shown in Table 7.1.

Table 7.1: The key schedules of Cobra-F64a and Cobra-F64b

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
K_i^1	W_1	W_2	W_3	W_4	W_2	W_1	W_4	W_3	W_1	W_2	W_4	W_3	W_1	W_4	W_2	W_3	W_2	W_4	W_3	W_1	W_2
K_i^2	W_4	W_3	W_1	W_2	W_3	W_2	W_1	W_4	W_2	W_3	W_1	W_2	W_3	W_1	W_3	W_4	W_3	W_1	W_4	W_2	W_3

7.2.4 Encryption Procedure

Cobra-F64a and Cobra-F64b both take as input a 64-bit plaintext block P and have a total of N rounds, where N is 16 for Cobra-F64a, and 20 for Cobra-F64b. The encryption procedures of Cobra-F64a and Cobra-F64b are as follows, where A_0, B_0, A_i, B_i are 32-bit variables.

```
1. $P = (A_0, B_0)$.
2. For $i = 1$ to $N$:
     if $i \leq N - 1$,
         $(A_i, B_i) = \mathbf{F}(A_{i-1}, B_{i-1}, K_i^1, K_i^2)$,
         $(A_i, B_i) = (B_i, A_i)$.
     else
         $(A_i, B_i) = \mathbf{F}(A_{i-1}, B_{i-1}, K_i^1, K_i^2)$.
3.   • For Cobra-F64a: Ciphertext $= (A_N \boxminus K_{N+1}^1, B_N \boxplus K_{N+1}^2)$.
     • For Cobra-F64b: Ciphertext $= (A_N \oplus K_{N+1}^1, B_N \oplus K_{N+1}^2)$.
```

The ith iteration of Step 2 in the above description is referred to below as Round i, $(1 \leq i \leq N)$, and the transformation in Step 3 is referred to below as the final transformation.

7.3 Previous Cryptanalytic Results

In 2005, Lee, Kim, Hong, Sung and Lee [68] presented a related-key differential attack on the first 11 rounds of Cobra-F64a, and a related-key differential attack on the first 18 rounds of Cobra-F64b. These are the only previously published cryptanalytic results on Cobra-F64a and Cobra-F64b.

7.4 Properties of Cobra-F64a and Cobra-F64b

In 2004, Ko et al. [62, 63] gave the following three properties of the Cobra DDP-boxes.

Property 7.1 *Let Δx be the difference between two inputs x and x' of $\mathbf{P}_{2,1}$, Δv be the difference between two controlling vectors v and v' of $\mathbf{P}_{2,1}$, and Δy be the difference between the two outputs $\mathbf{P}_{2,1}(x,v)$ and $\mathbf{P}_{2,1}(x',v')$. Then:*

(a) $\mathbf{P}_{2,1}(x,0) = \mathbf{P}_{2,1}(x,1)$ *holds if and only if the two bits of the input x are equal, i.e. it holds with probability $\frac{1}{2}$.*

(b) $\Pr(\Delta y = \langle 10\rangle_2 | \Delta x \in \{\langle 10\rangle_2, \langle 01\rangle_2\}, \Delta v = 0) = \Pr(\Delta y = \langle 01\rangle_2 | \Delta x \in \{\langle 10\rangle_2, \langle 01\rangle_2\}, \Delta v = 0) = 2^{-1}$.

(c) $\Pr(\Delta y = \langle 10\rangle_2 | \Delta x \in \{\langle 10\rangle_2, \langle 01\rangle_2\}, \Delta v = 1) = \Pr(\Delta y = \langle 01\rangle_2 | \Delta x \in \{\langle 10\rangle_2, \langle 01\rangle_2\}, \Delta v = 1) = 2^{-1}$.

(d) $\Pr(\Delta y = \langle 11\rangle_2 | \Delta x = \langle 00\rangle_2, \Delta v = 1) = \Pr(\Delta y = \langle 00\rangle_2 | \Delta x = \langle 00\rangle_2, \Delta v = 1) = 2^{-1}$.

Property 7.2 *Suppose $X, X' \in \{0,1\}^8$ and $V \in \{0,1\}^{12}$. If $X \oplus X' = e_i$ for some i ($1 \leq i \leq 8$), then $\mathbf{P}_{8,12}(X,V) \oplus \mathbf{P}_{8,12}(X',V) = e_j$, for some j, ($1 \leq j \leq 8$). If i and j are fixed, then the path for the differential $\Delta e_i \rightarrow \Delta e_j$ is fixed.*

Property 7.3 *Suppose $X \in \{0,1\}^n$ and $V \in \{0,1\}^m$. Then the following properties hold for all the various values of n and m.*

(a) $\Pr(\mathbf{P}_{n,m}(X,V) = \mathbf{P}_{n,m}(X, V \oplus e_i)) = 2^{-1}$, *for every i ($1 \leq i \leq m$).*

(b) If $X' \in \{0,1\}^n$ *then* $W(X \oplus X') = W(\mathbf{P}_{n,m}(X,V) \oplus \mathbf{P}_{n,m}(X',V))$*, where* W *is the Hamming Weight function.*

In 2005, Lee et al. [68] gave two further properties of the DDP-boxes $\mathbf{P}_{32,96}$ and $\mathbf{P}^{(\omega)}_{32,32}$ used in Cobra-F64a and Cobra-F64b; we now give these two properties, correcting certain errors in the versions given in [68].

Property 7.4 *Let* Δx *be the difference between two inputs* x *and* x' *of* $\mathbf{P}_{32,96}$*,* Δv *be the difference between two controlling vectors* v *and* v' *of* $\mathbf{P}_{32,96}$*, and* Δy *be the difference between the two outputs* $\mathbf{P}_{32,96}(x,v)$ *and* $\mathbf{P}_{32,96}(x',v')$*. Then:*

(a) $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = 0) = 2^{-5}$.

(b) $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = e_1) = 2^{-5}$.

Proof. (a) As shown in Figure 7.1, there are six layers of DDP-boxes $\mathbf{P}_{2,1}$ in a $\mathbf{P}_{32,96}$. Given the difference e_1 between two inputs and a zero difference between two controlling vectors to $\mathbf{P}_{32,96}$, there are two possibilities to get $\Delta y = e_1$: one is that the controlling bits in the first $\mathbf{P}_{2,1}$ DDP-boxes of the six layers are all zero, which happens with a probability of 2^{-6}; the other is that the controlling bit in the first $\mathbf{P}_{2,1}$ DDP-box of the first layer is 1 and the controlling bits in the third $\mathbf{P}_{2,1}$ of the middle four layers and the first $\mathbf{P}_{2,1}$ of the last layer are all zero, which happens also with a probability of 2^{-6}. Therefore, we get that $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = 0) = 2^{-6} + 2^{-6} = 2^{-5}$.

(b) Given the difference e_1 between two inputs and the difference e_1 between two controlling vectors to $\mathbf{P}_{2,1}$, we can get either of the differences $\langle 01 \rangle_2$ and $\langle 01 \rangle_2$ between the two outputs $\mathbf{P}_{2,1}$ with a probability of 2^{-1}. For the case of $\langle 01 \rangle_2$, if the controlling bits in the third $\mathbf{P}_{2,1}$ of the middle four layers and the first $\mathbf{P}_{2,1}$ of the last layer are all zero, we can get $\Delta y = e_1$, which happens with a probability of 2^{-5}. For the case of $\langle 10 \rangle_2$, if the controlling bits in the third $\mathbf{P}_{2,1}$ of the last five layers are all zero, we can get $\Delta y = e_1$, which happens also with a probability of 2^{-5}. Hence, we get that $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = e_1) = 2^{-1} \times 2^{-5} + 2^{-1} \times 2^{-5} = 2^{-5}$. □

Property 7.5 *Let* Δx *be the difference between two inputs* x *and* x' *of* $\mathbf{P}^{(0)}_{32,32}$*,* Δv

be the difference between two controlling vectors v and v' of $\mathbf{P}^{(0)}_{32,32}$, and Δy be the difference between the two outputs $\mathbf{P}^{(0)}_{32,32}(x, v)$ and $\mathbf{P}^{(0)}_{32,32}(x', v')$. Then:

(a) $\Pr(\Delta y = 0 | \Delta x = 0, \Delta v = e_1) = 2^{-3}$.

(b) $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = 0) = 2^{-5}$.

(c) $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = e_1) = 2^{-7}$.

(d) $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = e_9) = 2^{-8}$.

(e) $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = e_{1,9}) = 2^{-10}$.

Proof. (a) As introduced in Section 7.2.2, $\mathbf{P}^{(0)}_{32,32}$ is the functional composition of $\mathbf{T}$, followed by $\mathbf{P}^{(0)}_{96,1}$ and then $\mathbf{P}_{32,96}$. A DDP-Box $\mathbf{P}_{32,96}$ consists of six layers of DDP-boxes $\mathbf{P}_{2,1}$. After the application of $\mathbf{T}$ and $\mathbf{P}^{(0)}_{96,1}$, the difference e_1 between two controlling vectors of $\mathbf{P}^{(0)}_{32,32}$ will produce a one difference in the following three controlling bits of the $\mathbf{P}_{32,96}$ in $\mathbf{P}^{(0)}_{32,32}$: the first $\mathbf{P}_{2,1}$ of the first layer, the 7th $\mathbf{P}_{2,1}$ of the second layer and the 13th $\mathbf{P}_{2,1}$ of the third layer, and a zero difference in the other controlling bits of $\mathbf{P}_{32,96}$. Thus, this property proves correct following Property 7.1(a).

(b) Similar to the proof of Property 7.4(a).

(c) As mentioned above, after the application of $\mathbf{T}$ and $\mathbf{P}^{(0)}_{96,1}$, the difference e_1 between two controlling vectors of $\mathbf{P}^{(0)}_{32,32}$ will produce a one difference in the following three controlling bits of the $\mathbf{P}_{32,96}$ in $\mathbf{P}^{(0)}_{32,32}$: the first $\mathbf{P}_{2,1}$ of the first layer, the 7th $\mathbf{P}_{2,1}$ of the second layer and the 13th $\mathbf{P}_{2,1}$ of the third layer, and a zero difference in the other controlling bits of $\mathbf{P}_{32,96}$. Thus, to get $\Delta y = e_1$ we require that the following requirements hold simultaneously.

- The two inputs to the 7th $\mathbf{P}_{2,1}$ of the second layer produce a zero output difference;
- The two inputs to the 13th $\mathbf{P}_{2,1}$ of the second layer produce a zero output difference;

- When the two inputs to the first $\mathbf{P}_{2,1}$ of the first layer produce the output differences $\langle 01\rangle_2$, the controlling bits in the third $\mathbf{P}_{2,1}$ of the middle four layers and the first $\mathbf{P}_{2,1}$ of the last layer are all zero, which happens with a probability of 2^{-5}; or when the two inputs to the first $\mathbf{P}_{2,1}$ of the first layer produce the output differences $\langle 10\rangle_2$, the controlling bits in the third $\mathbf{P}_{2,1}$ of the last five layers are all zero, which happens also with a probability of 2^{-5}.

By Property 7.1(a), we know that each of the first two requirements holds with a probability of 2^{-1}; similarly to Property 7.4(b) we know that the last requirement holds with a probability of $2^{-1} \times 2^{-5} + 2^{-1} \times 2^{-5} = 2^{-5}$. Therefore, we learn that $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = e_1) = 2^{-1} \times 2^{-1} \times 2^{-5} = 2^{-7}$.

(d) After the application of $\mathbf{T}$ and $\mathbf{P}^{(0)}_{96,1}$, the difference e_9 between two controlling vectors of $\mathbf{P}^{(0)}_{32,32}$ will produce a one difference in the following three controlling bits of the $\mathbf{P}_{32,96}$ in $\mathbf{P}^{(0)}_{32,32}$: the 9th $\mathbf{P}_{2,1}$ of the first layer, the 15th $\mathbf{P}_{2,1}$ of the second layer and the 5th $\mathbf{P}_{2,1}$ of the third layer, and a zero difference in the other controlling bits of $\mathbf{P}_{32,96}$. Thus, to get $\Delta y = e_1$ we require that the following requirements hold simultaneously.

- The two inputs to the 9th $\mathbf{P}_{2,1}$ of the first layer produce a zero output difference;
- The two inputs to the 15th $\mathbf{P}_{2,1}$ of the second layer produce a zero output difference;
- The two inputs to the 5th $\mathbf{P}_{2,1}$ of the third layer produce a zero output difference;
- When the two inputs to the first $\mathbf{P}_{2,1}$ of the first layer produce the output differences $\langle 01\rangle_2$, the controlling bits in the third $\mathbf{P}_{2,1}$ of the middle four layers and the first $\mathbf{P}_{2,1}$ of the last layer are all zero; or when the two inputs to the first $\mathbf{P}_{2,1}$ of the first layer produce the output differences $\langle 10\rangle_2$, the controlling bits in the third $\mathbf{P}_{2,1}$ of the last five layers are all zero.

By Property 7.1(a), we know that each of the first three requirements holds with a probability of 2^{-1}; similarly to Property 7.4(a) we know that the last requirement holds with a probability of $2^{-1} \times 2^{-5} + 2^{-1} \times 2^{-5} = 2^{-5}$. Therefore, we learn that $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = e_9) = 2^{-1} \times 2^{-1} \times 2^{-1} \times 2^{-5} = 2^{-8}$.

(e) After the application of $\mathbf{T}$ and $\mathbf{P}^{(0)}_{96,1}$, the difference $e_{1,9}$ between two controlling vectors of $\mathbf{P}^{(0)}_{32,32}$ will produce a one difference in the following six controlling bits of the $\mathbf{P}_{32,96}$ in $\mathbf{P}^{(0)}_{32,32}$: the 1st and 9th $\mathbf{P}_{2,1}$ of the first layer, the 7th and 15th $\mathbf{P}_{2,1}$ of the second layer and the 5th and 13th $\mathbf{P}_{2,1}$ of the third layer, and a zero difference in the other controlling bits of $\mathbf{P}_{32,96}$. Thus, to get $\Delta y = e_1$ we require that the following requirements hold simultaneously.

- The two inputs to the 9th $\mathbf{P}_{2,1}$ of the first layer produce a zero output difference;
- The two inputs to the 7th $\mathbf{P}_{2,1}$ of the second layer produce a zero output difference;
- The two inputs to the 15th $\mathbf{P}_{2,1}$ of the second layer produce a zero output difference;
- The two inputs to the 5th $\mathbf{P}_{2,1}$ of the third layer produce a zero output difference;
- The two inputs to the 13th $\mathbf{P}_{2,1}$ of the third layer produce a zero output difference;
- When the two inputs to the first $\mathbf{P}_{2,1}$ of the first layer produce the output differences $\langle 01 \rangle_2$, the controlling bits in the third $\mathbf{P}_{2,1}$ of the middle four layers and the first $\mathbf{P}_{2,1}$ of the last layer are all zero; or when the two inputs to the first $\mathbf{P}_{2,1}$ of the first layer produce the output differences $\langle 10 \rangle_2$, the controlling bits in the third $\mathbf{P}_{2,1}$ of the last five layers are all zero.

By Property 7.1(a), we know that each of the first five requirements holds with a probability of 2^{-1}; similarly to Property 7.4(b) we know that the last requirement holds with a probability of $2^{-1} \times 2^{-5} + 2^{-1} \times 2^{-5} = 2^{-5}$. Therefore, we learn that $\Pr(\Delta y = e_1 | \Delta x = e_1, \Delta v = e_{1,9}) = 2^{-1} \times 2^{-1} \times 2^{-1} \times 2^{-1} \times 2^{-1} \times 2^{-5} = 2^{-10}$. □

7.5 Related-Key Rectangle Attack on Cobra-F64a

In this section, we first describe a 15-round related-key rectangle distinguisher with probability $2^{-123.62}$ for Cobra-F64a. This then allows us to construct a related-key rectangle attack on the full Cobra-F64a. Note that in this section we are concerned exclusively with Cobra-F64a, and all statements made refer specifically to that cipher.

7.5.1 A 15-Round Related-Key Rectangle Distinguisher with Probability $2^{-123.62}$

Let $\mathbf{E}^0$ denote Rounds 2 to 9, and $\mathbf{E}^1$ denote Rounds 10 to 16 including the final transformation. The 15-round related-key rectangle distinguisher involves four cipher keys (TYPE 1 as described in Section 2.2.9), which we assume are K_A, K_B, K_C and K_D. The first part of this 15-round distinguisher is an 8-round related-key differential $\Delta\alpha \to \Delta\beta$ with probability 2^{-18} for $\mathbf{E}^0$. This has the form: $(e_1, 0) \to (0, e_1)$, where the key difference is $K_A \oplus K_B = K_C \oplus K_D = (e_1, 0, 0, 0)$. The second part of the 15-round related-key distinguisher differential is made up of a 7-round related-key differential $\Delta\gamma \to \Delta\delta$ with probability 2^{-12} for $\mathbf{E}^1$ (Rounds 10 to 16, and the final transformation). This has the form: $(e_1, 0) \to (0, 0)$, where the key difference is $K_A \oplus K_C = K_B \oplus K_D = (e_1, 0, 0, 0)$. Table 7.2 shows more details of the two related-key differentials, where the difference in a round is the input difference to this round.

In the following, we need to sum the square of the probabilities of all the differentials $\Delta\alpha \to \Delta\beta^*$ with the same input difference α through $\mathbf{E}^0$, which is computationally infeasible. Instead, we just count those 8-round related-key differentials $\Delta\alpha \to \Delta\beta^*$ in each of which only the difference propagation of the second $\mathbf{P}^{(0)}_{32,32}$ in Round 9 is different from the 8-round related-key differential $\Delta\alpha \to \Delta\beta$ in Table 7.2, that is, the input difference and the controlling vector difference of the second $\mathbf{P}^{(0)}_{32,32}$ in Round 9 is 0 and e_1, respectively, and its 32-bit output difference t has a hamming weight of 2 with one bit difference in the first byte and the other bit in the second byte (Case A) or one bit difference in the first two bytes and the other bit in the last two bytes (Case B). The contributions of the remaining 8-round related-key

Table 7.2: The related-key differentials in the 15-round related-key rectangle distinguisher

Round(i)	$(\Delta A_{i-1}, \Delta B_{i-1})$	$(\Delta K_i^1, \Delta K_i^2)$	Prob.
2	$(e_1, 0)$	$(0, 0)$	2^{-6}
3	$(0, e_1)$	$(0, e_1)$	1
4	$(0, 0)$	$(0, 0)$	1
5	$(0, 0)$	$(0, 0)$	1
6	$(0, 0)$	$(e_1, 0)$	2^{-6}
7	$(0, e_1)$	$(0, e_1)$	1
8	$(0, 0)$	$(0, 0)$	1
9	$(0, 0)$	$(e_1, 0)$	2^{-6}
output	$(0, e_1)$	/	/
10	$(e_1, 0)$	$(0, 0)$	2^{-6}
11	$(0, e_1)$	$(0, e_1)$	1
12	$(0, 0)$	$(0, 0)$	1
13	$(0, 0)$	$(e_1, 0)$	2^{-6}
14	$(0, e_1)$	$(0, e_1)$	1
15	$(0, 0)$	$(0, 0)$	1
16	$(0, 0)$	$(0, 0)$	1
FT	$(0, 0)$	$(0, 0)$	1
output	$(0, 0)$	/	/

differentials are negligible. We now analyse the probabilities corresponding to these two cases. Consider the second $\mathbf{P}_{32,32}^{(0)}$ in Round 9, where the controlling vector difference is e_1 and the input difference is 0. As shown in Figure 7.4, the controlling vector difference e_1 is propagated to V'_{1_1}, V'_{2_7} and $V'_{3_{13}}$ after the extension $\mathbf{T}$ and the transposition $\mathbf{P}_{96,1}^{(0)}$ in this $\mathbf{P}_{32,32}^{(0)}$.

- For Case A, there exist only the following two possible sources:
  1. The DDP-box $\mathbf{P}_{2,1}$ corresponding to $V'_{3_{13}}$ produces a difference $\langle 11 \rangle_2$, and the other two DDP-boxes $\mathbf{P}_{2,1}$ corresponding to V'_{1_1} and V'_{2_7} produce a difference $\langle 00 \rangle_2$. From Property 7.1(d), this holds with a probability of $2^{-1} \times 2^{-1} \times 2^{-1} = 2^{-3}$. Then, to get any specific difference in Case A, we have a probability of $2^{-3} \times 2^{-3} = 2^{-6}$, as there are three layers of DDP-boxes to reach each one-bit difference. As a result, the probability of getting any specific difference in Case A from this source is $2^{-3} \times 2^{-6} = 2^{-9}$.
  2. The DDP-box $\mathbf{P}_{2,1}$ corresponding to V'_{1_1} produces a difference $\langle 11 \rangle_2$, and the other two DDP-boxes $\mathbf{P}_{2,1}$ corresponding to V'_{2_7} and $V'_{3_{13}}$ produce a difference $\langle 00 \rangle_2$. Again, we can learn from Property 7.1(d) that this holds with a probability of 2^{-3}. Then, since there are two traces to reach any specific difference in Case A and there are five layers of DDP-boxes to

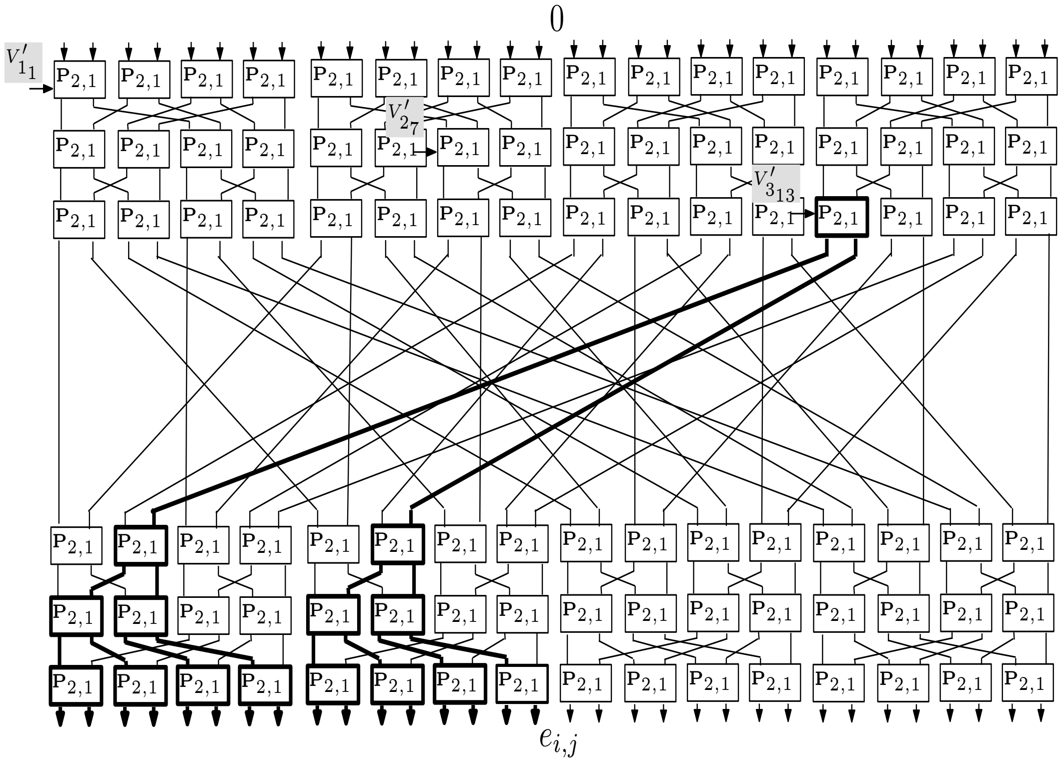


Figure 7.4: The $\mathbf{P}_{32,96}$ in $\mathbf{P}^{(0)}_{32,32}(\Delta X = 0, \Delta V = e_1)$

reach each one-bit difference, we have a probability of $2 \times 2^{-5} \times 2^{-5} = 2^{-9}$. As a result, the probability of getting any specific difference in Case A from this source is $2^{-3} \times 2^{-9} = 2^{-12}$.

Finally, we can conclude from the above analysis that the probability of getting any specific difference in Case A is $2^{-9} + 2^{-12}$.

- For Case B, there also exist only the following two possible sources:

  1. The DDP-box $\mathbf{P}_{2,1}$ corresponding to V'_{2_7} produces a difference $\langle 11 \rangle_2$, and the other two DDP-boxes $\mathbf{P}_{2,1}$ corresponding to V'_{1_1} and $V'_{3_{13}}$ produce a difference $\langle 00 \rangle_2$, which holds with a probability of $2^{-1} \times 2^{-1} \times 2^{-1} = 2^{-3}$. Then, as there are four layers of DDP-boxes to reach each one-bit difference of any specific difference in Case B, we have a probability of $2^{-4} \times 2^{-4} = 2^{-8}$. As a result, the probability of getting any specific difference in Case B from this source is $2^{-3} \times 2^{-8} = 2^{-11}$.
  2. The DDP-box $\mathbf{P}_{2,1}$ corresponding to V'_{1_1} produces a difference $\langle 11 \rangle_2$, and the other two DDP-boxes $\mathbf{P}_{2,1}$ corresponding to V'_{2_7} and $V'_{3_{13}}$ produce

a difference $\langle 00\rangle_2$, which holds with a probability of 2^{-3}. Then, since there are two traces to reach any specific difference in Case B and there are five layers of DDP-boxes to reach each one-bit difference, we have a probability of $2 \times 2^{-5} \times 2^{-5} = 2^{-9}$. As a result, the probability of getting any specific difference in Case B from this source is $2^{-3} \times 2^{-9} = 2^{-12}$.

Finally, we can conclude from the above analysis that the probability of getting any specific difference in Case B is $2^{-11} + 2^{-12}$.

Therefore, after considering the probability 2^{-3} incurred in the first $\mathbf{P}^{(0)}_{32,32}$ in Round 9, we can compute a square sum of at least $1 \times (2^{-18})^2 + \binom{8}{1} \cdot \binom{8}{1} \cdot [2^{-12} \times 2^{-3} \times (2^{-9} + 2^{-12})]^2 + \binom{16}{1} \cdot \binom{16}{1} \cdot [2^{-12} \times 2^{-3}(2^{-11} + 2^{-12})]^2 \approx 2^{-35.96}$ for the 321 possible 8-round related-key differentials $(e_1, 0) \rightarrow (t, e_1)$, where $t \in \{0$, Case A, Case B$\}$.

We also need to sum the square of the probabilities of all the differentials $\Delta\gamma^* \rightarrow \Delta\delta$ with the same output difference δ through $\mathbf{E}^1$, which is also computationally infeasible. Alternatively, we just count those 7-round related-key differentials $\Delta\gamma^* \rightarrow \Delta\delta$ in each of which only the difference propagation of the first $\mathbf{P}^{(0)}_{32,32}$ in Round 10 is different from the 7-round related-key differential $\Delta\gamma \rightarrow \Delta\delta$ in Table 7.2, that is, the output difference and the controlling vector difference of the first $\mathbf{P}^{(0)}_{32,32}$ in Round 10 (through the encryption direction) is 0 and e_1, respectively, and its 32-bit input difference s has a hamming weight of 2. After noting that the two one-bit differences of such a differential can only distribute in the input to one of the three DDP-boxes $\mathbf{P}_{2,1}$ corresponding to V'_{1_1}, V'_{2_7} and $V'_{3_{13}}$, we can similarly compute a square sum of at least $1 \times (2^{-12})^2 + 1 \times (2^{-13})^2 + \binom{2}{1} \cdot \binom{2}{1} \cdot (2^{-16})^2 + \binom{4}{1} \cdot \binom{4}{1} \cdot (2^{-18})^2 \approx 2^{-23.66}$ for the 22 possible 7-round related-key differentials $\Delta\gamma^* \rightarrow \Delta\delta$. As a result, the distinguisher has a probability of $2^{-64} \times 2^{-35.96} \times 2^{-23.66} = 2^{-123.62}$ for the correct key, while it has a probability of $(2^{-64})^2 = 2^{-128}$ for a wrong key.

7.5.2 Attack Description

We can use the 15-round distinguisher to mount a related-key rectangle attack on the full Cobra-F64a. The attack procedure is as follows.

1. Choose $2^{63.81}$ ciphertext pairs (C_i, C_i^*) with $C_i = C_i^*$, $(i = 1, \cdots, 2^{63.81})$. In a chosen-ciphertext attack scenario, obtain all the plaintexts for the $2^{63.81}$ ciphertexts C_i decrypted with K_A; we denote by P_i the plaintext for ciphertext C_i. In a chosen-ciphertext attack scenario, obtain all the plaintexts for the $2^{63.81}$ ciphertexts C_i^* decrypted with K_B; we denote by P_i^* the plaintext for ciphertext C_i^*, where $K_A \oplus K_B = (e_1, 0, 0, 0)$.

2. Guess a value for the 64-bit user key (W_1, W_4), and perform Steps (a) and (b) below.

   (a) Partially encrypt all the plaintexts P_i with (the guessed value for) (W_1, W_4) to get the corresponding values just after Round 1; we denote these values by T_i, respectively. Partially encrypt all the plaintexts P_i^* with $(W_1 \oplus e_1, W_4)$ to get the corresponding values just after Round 1; we denote them by T_i^*, respectively. Then, store all the values T_i and T_i^* into a hash table. Finally, choose only the quartets $(T_{i_1}, T_{i_1}^*, T_{i_2}, T_{i_2}^*)$ such that $T_{i_1} \oplus T_{i_2}^* = T_{i_1}^* \oplus T_{i_2} = (e_1, 0)$, where $1 \leq i_1 < i_2 \leq 2^{63.81}$. If six or more quartets $(T_{i_1}, T_{i_1}^*, T_{i_2}, T_{i_2}^*)$ pass this condition, execute Step 2(b) with the quartets meeting this condition; otherwise, repeat Step 2 with another guess.

   (b) Guess a value for the 64-bit user key (W_2, W_3). Partially encrypt all remaining quartets $(T_{i_1}, T_{i_1}^*, T_{i_2}, T_{i_2}^*)$ with (the guessed value for) (W_2, W_3) to get the corresponding values just after Round 2; we denote them by $(\overline{T}_{i_1}, \overline{T}_{i_1}^*, \overline{T}_{i_2}, \overline{T}_{i_2}^*)$, respectively. Finally, check whether $\overline{T}_{i_1} \oplus \overline{T}_{i_2}^* = \overline{T}_{i_1}^* \oplus \overline{T}_{i_2} = (0, e_1)$. If six or more quartets $(T_{i_1}, T_{i_1}^*, T_{i_2}, T_{i_2}^*)$ pass this condition, record the guessed value for (W_1, W_2, W_3, W_4), and execute Step 3; otherwise, repeat this step with another guess, (if all the 2^{64} possible values for (W_2, W_3) are tested, repeat Step 2 with another guess for (W_1, W_4)).

3. For every recorded value for (W_1, W_2, W_3, W_4), do a trial encryption with one known plaintext/ciphertext pair. If one is suggested, output it as the user key of Cobra-F64a; otherwise, go to Step 2.

7.5.3 Complexity Analysis

The attack requires $2^{64.81}$ related-key chosen ciphertexts, which have a time complexity of $2^{64.81}$ encryptions. The required memory for this attack is dominated by the encrypted plaintext pairs, which is approximately $2^{64.81} \times 8 = 2^{67.81}$ memory bytes.

Step 2(a) has a time complexity of about $2^{64} \times 2^{64.81} \times \frac{1}{2} \times \frac{1}{16} \approx 2^{123.81}$ 16-round Cobra-F64a encryptions, where $\frac{1}{2}$ means the average fraction of 64-bit key pairs that are tested in Step 2(a). In Step 2(a), a total of about $\binom{2^{63.81}}{2} \approx 2^{126.62}$ candidate quartets are yielded, and the probability that the number of the quartets for a wrong key is no less than six is approximately $\sum_{i=6}^{126.62}[\binom{126.62}{i} \cdot (2^{-64\times2})^i \cdot (1-2^{-64\times2})^{126.62-i}] \approx 2^{-17.77}$. Thus, about $2^{64} \times 2^{-17.77} \times \frac{1}{2} \approx 2^{45.23}$ keys pass Step 2(a) for every guess of (W_1, W_4).

Step 2(b) has a time complexity of $2^{45.23} \times 2^{64} \times 6 \times 4 \times \frac{1}{16} \approx 2^{108.65}$ 16-round Cobra-F64a encryptions. In Step 2(b), probability 2^{-6} is required to satisfy the one-round differential characteristic for Round 2, and the number of the quartets to be tested in this step is at least 6, so the probability that a wrong guess for (W_2, W_3) passes Step 2(b) is about $(2^{-6})^{6\times2} = 2^{-96}$. As a result, the expected number of the recorded values for (W_1, W_2, W_3, W_4) in Step 2(b) is $2^{45.23} \times 2^{64} \times 2^{-96} = 2^{13.23}$. As a consequence, Step 3 has a time complexity of $2^{13.23}$ 16-round Cobra-F64a encryptions.

Therefore, this attack requires a total time complexity of $2^{123.81}$ full-round Cobra-F64a encryptions.

The probability that a wrong 128-bit key is suggested in Step 3 is approximately 2^{-64}, thus the expected number of suggested wrong 128-bit keys is about $2^{-64} \times 2^{13.23} \approx 2^{-50.77}$, which is quite low. The expected number of quartets passing Step 2(b) for the right key pair is $2^{126.62} \times 2^{-123.62} = 8$, and the probability that the number of the quartets for the right subkey is no less than six is approximately $\sum_{i=6}^{2^{126.62}}[\binom{2^{126.62}}{i} \cdot (2^{-123.62})^i \times (1-2^{-123.62})^{2^{126.62}-i}] \approx 0.8$. Therefore, the related-key rectangle attack can break the full Cobra-F64a, with a success probability of 80%.

7.6 Related-Key Differential Attack on Cobra-F64b

In this section, we first describe a 19.5-round related-key differential characteristic with probability 2^{-57} of Cobra-F64b. This then enables us to construct a related-key differential attack on the full Cobra-F64b. Note that in his section we are concerned exclusively with Cobra-F64b, and all statements made refer specifically to that cipher.

7.6.1 A 19.5-Round Related-Key Differential Characteristic with Probability 2^{-57}

We describe a 19.5-round related-key differential characteristic $(0, e_1) \rightarrow (e_1, 0)$ with probability 2^{-57}, where the key difference is (e_1, e_1, e_1, e_1). See Table 7.3 for more details of the 19.5-round related-key differential characteristic. It is derived from the full-round related-key differential characteristic presented in [68].

Table 7.3: The 19.5-round related-key differential characteristic

Round(i)	$(\Delta A_{i-1}, \Delta B_{i-1})$	$(\Delta K_i^1, \Delta K_i^2)$	Prob.
1	$(0, e_1)$	(e_1, e_1)	2^{-3}
2	$(0, e_1)$	(e_1, e_1)	2^{-3}
3	$(0, e_1)$	(e_1, e_1)	2^{-3}
$\vdots$	$\vdots$	$\vdots$	$\vdots$
18	$(0, e_1)$	(e_1, e_1)	2^{-3}
19	$(0, e_1)$	(e_1, e_1)	2^{-3}
20(half)	$(0, e_1)$	(e_1, e_1)	$1^{\dagger}$
output	$(e_1, 0)$	/	/

†: This probability is just for the difference between the intermediate values XORed with the 20th round subkey

7.6.2 Attack Description

In order to reduce the time complexity of our attack, we use the following filtering property: some possible differences between a pair of ciphertexts can be partially determined from the output difference $(e_1, 0)$ of the 19.5-round related-key differential, for those ciphertext pairs that do not meet these differences can be discarded immediately. More specifically, as the input difference and the controlling vector difference of the DDP-box $\mathbf{P}_{32,32}^{(0)}$ in Round 20 are 0 and e_1, respectively, the output difference of this $\mathbf{P}^{(0)}_{32,32}$ should have a hamming weight of 0, 2, 4 or 6, which is caused by the three inherent DDP-boxes $\mathbf{P}_{2,1}$ corresponding to V'_{1_1}, V'_{2_7} and $V'_{3_{13}}$. After an analysis on the $\mathbf{P}^{(0)}_{32,32}$, we conclude that there are at most $\binom{32}{2}\cdot\binom{16}{1}\cdot\binom{16}{1}\cdot\binom{8}{1}\cdot\binom{8}{1} = 31\times 2^{18}$ possible values for those that have a hamming weight of 6, at most $\binom{32}{2}\cdot\binom{16}{1}\cdot\binom{16}{1}+\binom{32}{2}\cdot\binom{8}{1}\cdot\binom{8}{1}+\binom{16}{1}\cdot\binom{16}{1}\cdot\binom{8}{1}\cdot\binom{8}{1} = 31\times 2^{12}+31\times 2^{10}+2^{14}$ possible values for those that have a hamming weight of 4, at most $\binom{32}{2} = 31 \times 2^4$ possible values for those that have a hamming weight of 2, and only 1 with a hamming weight of 0. Therefore, the number of possible output differences of the $\mathbf{P}^{(0)}_{32,32}$ is totally $31 \times 2^{18} + 31 \times 2^{12} + 31 \times 2^{10} + 2^{14} + 31 \times 2^4 + 1 = 8302065$. After XORed with the subkey difference $\Delta W_3 = e_1$ in the final transformation, these 8302065 possible output differences of the $\mathbf{P}^{(0)}_{32,32}$ incur 8302065 possible output differences between the right halve of the pair of ciphertexts. We denote the resultant 8302065 possible output differences by the set Ω. We will not count the possible number for the left halve, for it seems infeasible due to the right rotation and addition modulo 2^{32} operations in Round 20.

Consequently, we can conduct the following related-key differential attack to break the full Cobra-F64b.

1. Choose 2^{60} pairs of plaintexts (P_i, P_i^*) with $P_i \oplus P_i^* = (0, e_1)$, $i = 1, \cdots, 2^{60}$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{60} plaintexts P_i encrypted with K_A; we denote by C_i the ciphertext for plaintext P_i. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{60} plaintexts P_i^* encrypted with K_B; we denote by C_i^* the ciphertext for plaintext P_i^*, where $K_A \oplus K_B = (e_1, e_1, e_1, e_1)$. Keep only the pairs (C_i, C_i^*) such that the right half of the difference $C_i \oplus C_i^*$ belongs to the set Ω.

2. Guess a value for the 64-bit key (W_2, W_3), and perform Steps (a) and (b) below.

   (a) Partially decrypt all the remaining ciphertexts C_i with (the guessed value for) (W_2, W_3) to get the corresponding values just after the data (A_{19}, B_{19}) XORed with the 20th round subkey (K^1_{20}, K^2_{20}) in Round 20 (i.e. just after the last 0.5 round in Round 20 through the backward direction); we denote them by T_i, respectively. Partially decrypt all the remaining ciphertexts C_i^* with $(W_2 \oplus e_1, W_3 \oplus e_1)$ to get the respective corresponding values just after the last 0.5 round in Round 20 through the backward direction; we denote them by T_i^*, respectively. Check whether $T_i \oplus T_i^* = (e_1, 0)$. If six or more pairs (T_i, T_i^*) pass this condition, execute Step 2(b) with the pairs meeting this condition; otherwise, repeat Step 2 with other guess.

(b) Guess a value for the 32-bit key W_1. For each remaining pair (T_i, T_i^*), partially decrypt T_i with (W_1, W_2) to get its corresponding value just after the data (A_{18}, B_{18}) XORed with the 19th round subkey (K_{19}^1, K_{19}^2) in Round 19 (i.e. just after the last 1.5 round in Rounds 20 and 19 through the backward direction); we denote them by $\overline{T}_i$, respectively. Partially decrypt T_i^* with $(W_1 \oplus e_1, W_2 \oplus e_1)$ to get its corresponding value just after the last 1.5 round in Rounds 20 and 19 through the backward direction; we denote them by $\overline{T}_i^*$, respectively. Check whether $\overline{T}_i \oplus \overline{T}_i^* = (e_1, 0)$. If six or more pairs (T_i, T_i^*) pass this condition, record the guessed value for (W_1, W_2, W_3), and execute Step 3; otherwise, repeat this step with another guess, (if all the 2^{32} possible values for W_1 are tested, repeat Step 2 with another guess for (W_2, W_3).

3. For every recorded value for (W_1, W_2, W_3), do an exhaustive search for the remaining 32-bit subkey W_4 using trial encryption. Two known pairs of plaintexts and ciphertexts are enough for this trial process. If a 128-bit key is suggested, output it as the user key of the full Cobra-F64b; otherwise, go to Step 2.

7.6.3 Complexity Analysis

This attack requires 2^{61} related-key chosen plaintexts, which have a time complexity of 2^{61} full-round Cobra-F64b encryptions. The required memory for this attack is dominated by the ciphertext pairs, which is approximately $2^{61} \times 8 = 2^{64}$ memory bytes.

Due to the filtering condition in Step 1, about $2^{60} \times \frac{8302065}{2^{32}} \approx 2^{50.99}$ pairs remain after Step 1.

Step 2(a) has a time complexity of about $2^{64} \times 2^{51.99} \times \frac{1}{2} \times \frac{1}{20} \approx 2^{110.67}$ full-round Cobra-F64b encryptions, where $\frac{1}{2}$ means the average fraction of 64-bit key pairs that are tested in Step 2(a). In Step 2(a), the expected number of pairs recorded for each guessed key is about $2^{-41.01} \times 2^{50.99} = 2^{9.98}$, for the probability that each decrypted pair passes the condition of Step 2(a) is about $2^{-64} \times 8302065 = 2^{-41.01}$, which is due to the fact that the filtering step holds $8302065 = 2^{22.99}$ ciphertext differences.

Step 2(b) has a time complexity of about $2^{9.98} \times 2 \times 2^{96}\frac{1}{2} \times \frac{1}{20} \approx 2^{101.66}$ full-round Cobra-F64b encryptions. In Step 2(b), probability 2^{-3} is required to satisfy the one-round differential characteristic for Round 19 (refer to Table 7.3), and the probability that a wrong guess for (W_1, W_2, W_3) passes Step 2(b) is about $\sum_{i=6}^{2^{9.98}}[\binom{2^{9.98}}{i} \cdot (2^{-3})^i \times (1-2^{-3})^{2^{9.98}-i}] \approx 2^{-53}$. Step 3 has a time complexity of $2^{32} \times 2^{96} \times 2^{-53} \times \frac{1}{2} = 2^{74}$ full-round Cobra-F64b encryptions.

Therefore, the attack requires a total time complexity of $2^{110.67}$ full-round Cobra-F64b encryptions.

Since the probability that a wrong 128-bit key is suggested in Step 3 is approximately 2^{-128}, the expected number of suggested wrong 128-bit keys is about $2^{-128} \times 2^{74} \approx 2^{-54}$, which is extremely low. One the other hand, the expected number of text pairs for the right key pair is $2^{60} \times 2^{-57} = 8$, and the probability that the number of the pairs for the right key guess is no less than six is approximately $\sum_{i=6}^{2^{60}}[\binom{2^{60}}{i} \cdot (2^{-57})^i \cdot (1-2^{-57})^{2^{60}-i}] \approx 0.8$. Therefore, the related-key differential attack can break the full Cobra-F64b, with a success probability of 0.8.

7.7 Summary

In this chapter we have presented a related-key rectangle attack on the full Cobra-F64a and a related-key differential attack on the full Cobra-F64b. Table 7.4 summarises the published cryptanalytic results on Cobra-F64a and Cobra-F64b, where RK-CP refers to the required numbers of related-key chosen plaintexts, and Encryptions refers to the required number of encryption operations of Cobra-F64a or Cobra-F64b.

Table 7.4: Cryptanalytic results on Cobra-F64a and Cobra-F64b

Cipher	Attack Type	Rounds	Data	Time	Source
Cobra-F64a	Related-key differential	11	2^{59}RK-CP	2^{107}Encryptions	[68]
	Related-key rectangle	full(16)	$2^{64.81}$RK-CP	$2^{123.81}$Encryptions	Sect. 7.5
Cobra-F64b	Related-key differential	18	2^{58}RK-CP	2^{122}Encryptions	[68]
		full(20)	2^{61}RK-CP	$2^{110.67}$Encryptions	Sect. 7.6

Chapter 8

Related-Key Rectangle Attack on 44-Round SHACAL-2

SHACAL-2 is a 64-round block cipher with a 256-bit block length and a variable length key of up to 512 bits, which was selected as one of the NESSIE-recommended algorithms in 2003. In this chapter, we present a related-key rectangle attack on 44 rounds of SHACAL-2. The attack requires 2^{233} *related-key chosen plaintexts, and has a time complexity of* $2^{497.2}$ *encryptions. This is better than any previously published cryptanalytic results on SHACAL-2 in terms of the number of attacked rounds.*

Contents

8.1 Introduction

In 2000, Handschuh and Naccache [31] proposed a 160-bit block cipher SHACAL, standardised hash function SHA-1 [92]. In 2001, they then elaborated their original proposal to give two schemes, SHACAL-1 and SHACAL-2 [32], where SHACAL-1 is the same as the original SHACAL, and SHACAL-2 is a 256-bit block cipher based on the compression function of the hash function SHA-256 [93]. In both cases, the block cipher encryption operation is simply the compression function of the hash function, with the chaining value input set equal to the plaintext block, and the message block input set equal to the key. Both SHACAL-1 and SHACAL-2 were submitted to the NESSIE project [89], and were both selected for the second phase of the evaluation. However, although SHACAL-2 became a member of the final set of NESSIE recommended algorithms, SHACAL-1 was rejected because of concerns regarding its key schedule.

In this chapter, we first describe a novel a 35-round related-key rectangle distinguisher with probability 2^{-460} for SHACAL-2. We then use this distinguisher to specify a related-key rectangle attack on 44 rounds of SHACAL-2, using the early abort technique described in Section 4.3. The attack requires 2^{233} related-key chosen plaintexts, and has a time complexity of $2^{497.2}$ encryptions. This is better than any previously published cryptanalytic results on SHACAL-2 in terms of the number of attacked rounds.

The remainder of this chapter is organised as follows. In Section 8.2 we describe SHACAL-2. In Section 8.3 we briefly review previous cryptanalytic results on SHACAL-2. In Section 8.4 we describe certain properties of SHACAL-2. In Section 8.5 we give a 35-round related-key rectangle distinguisher with probability 2^{-460}, which forms the basis for the related-key rectangle attack on 44-round SHACAL-2 described in Section 8.6. Section 8.7 summarises the results of this chapter.

8.2 The SHACAL-2 Block Cipher

In this section we briefly describe the SHACAL-2 block cipher [32].

8.2.1 Notation

In this chapter, the bits of a 32-bit value are numbered from 1 to 32 from left to right, where the least significant bit is referred as the 1st bit, and the most significant bit is referred as the 32nd bit. We use the following notation.

- $\boxplus$: addition modulo 2^{32}
- $\boxminus$: subtraction modulo 2^{32}
- e_j: a 32-bit word with zeros in all positions but bit j, $(1 \leq j \leq 32)$
- $e_{i_1,\cdots,i_j}$: the 32-bit word equal to $e_{i_1} \oplus \cdots \oplus e_{i_j}$, $(1 \leq i_1, \cdots, i_j \leq 32)$
- $e_{j,\sim}$: a 32-bit word that has zeros in bits 1 to $j-1$, a one in bit j and indeterminate values in bits $(j+1)$ to 32, $(1 \leq j \leq 31)$

8.2.2 Functions

SHACAL-2 uses a number of functions, namely $\mathbf{\Psi}_0$, $\mathbf{\Psi}_1$, $\mathbf{\Phi}_0$, $\mathbf{\Phi}_1$, **Ch** and **Maj**. These functions are as follows.

- $\mathbf{\Psi}_0 : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$. If X is a 32-bit block, then $\mathbf{\Psi}_0(X) = (X \ggg 7) \oplus (X \ggg 18) \oplus (X \gg 3)$.
- $\mathbf{\Psi}_1 : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$. If X is a 32-bit block, then $\mathbf{\Psi}_1(X) = (X \ggg 17) \oplus (X \ggg 19) \oplus (X \gg 10)$.
- $\mathbf{\Phi}_0 : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$. If X is a 32-bit block, then $\mathbf{\Phi}_0(X) = (X \ggg 2) \oplus (X \ggg 13) \oplus (X \ggg 22)$.

- $\mathbf{\Phi}_1 : \{0,1\}^{32} \rightarrow \{0,1\}^{32}$. If X is a 32-bit block, then $\mathbf{\Phi}_1(X) = (X \ggg 6) \oplus (X \ggg 11) \oplus (X \ggg 25)$.
- $\mathbf{Ch} : \{0,1\}^{32} \times \{0,1\}^{32} \times \{0,1\}^{32} \rightarrow \{0,1\}^{32}$. If X, Y and Z are 32-bit blocks, then $\mathbf{Ch}(X, Y, Z) = (X \& Y) \oplus (\neg X \& Z)$.
- $\mathbf{Maj} : \{0,1\}^{32} \times \{0,1\}^{32} \times \{0,1\}^{32} \rightarrow \{0,1\}^{32}$. If X, Y and Z are 32-bit blocks, then $\mathbf{Maj}(X, Y, Z) = (X \& Y) \oplus (X \& Z) \oplus (Y \& Z)$.

8.2.3 Generation of Subkeys

SHACAL-2 uses a total of 64 32-bit subkeys K_i, $(1 \leq i \leq 64)$, all derived from a variable length key of up to 512 bits. Shorter keys can be used by padding them with zeros to produce a 512-bit key string; however, the proposers recommend that the key should not be shorter than 128 bits. Let a 512-bit user key K be represented as a sequence of sixteen 32-bit words $K_1, K_2, \cdots, K_{16}$, then these words form the round keys for the first 16 rounds. The remaining round keys K_i $(17 \leq i \leq 64)$ are defined as follows.

$$K_i \quad = \quad \mathbf{\Psi}_1(K_{i-2}) \boxplus K_{i-7} \boxplus \mathbf{\Psi}_0(K_{i-15}) \boxplus K_{i-16}.$$

8.2.4 Encryption Procedure

SHACAL-2 takes as input a 256-bit plaintext block P, and has a total of 64 rounds. Its encryption procedure is as follows, where $A^i, B^i, C^i, D^i, E^i, F^i, G^i, H^i, T_1^i, T_2^i$ are 32-bit variables, and θ_i are public constants.

1. Represent P as eight 32-bit words $P = (A^0, B^0, C^0, D^0, E^0, F^0, G^0, H^0)$.
2. For $i = 1$ to 64:

   $T_1^i = K_i \boxplus \mathbf{\Phi}_1(E^{i-1}) \boxplus \mathbf{Ch}(E^{i-1}, F^{i-1}, G^{i-1}) \boxplus H^{i-1} \boxplus \theta_i$,

   $T_2^i = \mathbf{\Phi}_0(A^{i-1}) \boxplus \mathbf{Maj}(A^{i-1}, B^{i-1}, C^{i-1})$,

   $H^i = G^{i-1}$,

   $G^i = F^{i-1}$,

$$F^i = E^{i-1},$$

$$E^i = D^{i-1} \boxplus T_1^i,$$

$$D^i = C^{i-1},$$

$$C^i = B^{i-1},$$

$$B^i = A^{i-1},$$

$$A^i = T_1^i \boxplus T_2^i.$$

3. The ciphertext $= (A^{64}, B^{64}, C^{64}, D^{64}, E^{64}, F^{64}, G^{64}, H^{64})$.

The ith iteration of Step 2 in the above description is referred to below as Round i, $(1 \leq i \leq 64)$.

8.3 Previous Cryptanalytic Results

In this section we briefly review previous cryptanalytic attacks on SHACAL-2.

- In 2003, Hong, Kim, Kim, Sung, Lee and Lee [39] presented an impossible differential attack on 30-round SHACAL-2.
- In 2004, Shin, Kim, Kim, Hong and Lee [98] presented a square-nonlinear attack on 28-round SHACAL-2 and a differential-nonlinear attack on 32-round SHACAL-2.
- In 2004, Kim, Kim, Lee, Lim and Song [53] presented a related-key differential-nonlinear attack on 35-round SHACAL-2, and a related-key rectangle attack on 37-round SHACAL-2, where the latter is based on a 33-round related-key rectangle distinguisher.
- In 2006, Lu, Kim, Keller and Dunkelman [78] presented a related-key rectangle attack on 42-round SHACAL-2, exploiting a 34-round related-key rectangle distinguisher with probability $2^{-456.76}$ and then using an early abort technique.
- In 2007, Wang [104] presented a related-key rectangle attack on 43-round SHACAL-2, based on an extension of Lu et al.'s 34-round related-key rectangle distinguisher to a 35-round distinguisher with probability $2^{-474.76}$.

Table 8.1: Differential distribution of the functions **Ch** and **Maj**

x	0	0	0	1	0	1	1	1
y	0	0	1	0	1	0	1	1
z	0	1	0	0	1	1	0	1
Ch	0	0/1	0/1	0/1	1	0/1	0/1	0/1
Maj	0	0/1	0/1	0/1	0/1	0/1	0/1	1

This latter result is the best previously published cryptanalytic result on SHACAL-2 in terms of the number of attacked rounds.

8.4 Properties of SHACAL-2

We first give the following general result, which can be used to compute differential probabilities of the addition modulo 2^{32} in SHACAL-2.

Theorem 8.1 ([73]) *Let x, y and z be 32-bit words. If $\Pr_{X,Y\in\{0,1\}^{32}}((X \boxplus Y) \oplus ((X \oplus \Delta x) \boxplus (Y \oplus \Delta y)) = \Delta z) > 0$, then $\Pr_{X,Y\in\{0,1\}^{32}}((X \boxplus Y) \oplus ((X \oplus \Delta x) \boxplus (Y \oplus \Delta y)) = \Delta z) = 2^{-s}$, where s is the number of the least significant 31 bit positions that do not satisfy $x_i = y_i = z_i$, where x_i denotes the ith bit of x, and so on, ($1 \leq i \leq 31$).*

We next give two further differential properties of SHACAL-2.

The following observations are due to Shin at al. [98]. The functions **Ch** and **Maj** operate in a bit-by-bit manner, and hence they can be regarded as functions having a 3-bit input and a 1-bit output. It is thus simple to calculate the differential properties of these functions, and these properties are summarised in Table 8.1. In this table, for each possible 3-bit difference, the possible differences in the outputs of the two functions are indicated, where 0, 1, and 0/1 respectively indicate that the output is always, 0, always 1, or either 0 or 1 with probability $\frac{1}{2}$.

Property 8.1 ([78, 104]) *Suppose that K and $\widetilde{K}$ are cipher keys, P and $\widetilde{P}$ are plaintext blocks, and let K_i and $\widetilde{K}_i$ ($1 \leq i \leq 64$) denote the subkeys derived from K and $\widetilde{K}$, respectively. Also let $(A^i, B^i, \cdots, H^i)$ denote the values obtained at the end of Round i when encrypting P using the key K, and let $(\widetilde{A}^i, \widetilde{B}^i, \cdots, \widetilde{H}^i)$ denote the corresponding values when encrypting $\widetilde{P}$ using $\widetilde{K}$.*

Then, if $(A^i, B^i, \cdots, H^i)$, $(\widetilde{A}^i, \widetilde{B}^i, \cdots, \widetilde{H}^i)$ *and* $K_i \boxminus \widetilde{K}_i$ *are known* $(5 \le i \le 64)$*, then the following values can readily be computed:*

(i) $(A^{i-1}, B^{i-1}, \cdots, G^{i-1})$ *and* $(\widetilde{A}^{i-1}, \widetilde{B}^{i-1}, \cdots, \widetilde{G}^{i-1})$*;*

(ii) $H^{i-1} \boxminus \widetilde{H}^{i-1}$*;*

(iii) $(A^{i-5}, B^{i-5}, C^{i-5})$ *and* $(\widetilde{A}^{i-5}, \widetilde{B}^{i-5}, \widetilde{C}^{i-5})$*;*

(iv) $D^{i-5} \boxminus \widetilde{D}^{i-5}$*.*

8.5 A 35-Round Related-Key Rectangle Distinguisher with Probability 2^{-460}

In this section, we describe a 35-round related-key rectangle distinguisher with probability 2^{-460} for Rounds 0 to 34 of SHACAL-2. This distinguisher is an extension of those described in [78, 104]. These related-key rectangle distinguishers involve two cipher keys (TYPE 3 as described in Section 2.2.9), which we assume are K and $\widetilde{K}$. We also describe a flaw in Wang's attack on 43-round SHACAL-2.

8.5.1 A 34-Round Related-Key Rectangle Distinguisher with Probability $2^{-456.76}$

In 2006, Lu et al. [78] described a 24-round related-key differential characteristic for Rounds 2 to 25 of SHACAL-2. This is of the form $(0, 0, e_{7,10,19,21,26,30}, e_{32}, 0, e_{10,14,20}, e_{19,30}, e_{32}) \rightarrow (e_{14,25,29}, 0, 0, 0, e_{14,25,29}, 0, 0, 0)$ and has probability 2^{-38}.[1] They also give a 10-round differential characteristic for Rounds 25 to 34 of SHACAL-2, which has the form $(e_{32}, e_{32}, e_{7,10,19,21,26,30,32}, 0, 0, e_{10,14,20}, e_{19,30,32}, 0) \rightarrow (e_{7,10,19,21,26,30}, e_{32}, 0, 0, e_{7,21,26}, e_{32}, 0, 0)$ and has probability 2^{-65}.

Then, they computed a square sum of at least $2^{-74} (= 2^{-37\times 2})$ for the probabilities of all the 24-round related-key differentials for Rounds 2 to 25 with the input difference $(0, 0, e_{7,10,19,21,26,30}, e_{32}, 0, e_{10,14,20}, e_{19,30}, e_{32})$, and a square sum of at least $2^{-126.76}$(=

[1] Certain input bits are fixed.

$2^{-63.38\times 2}$) for the probabilities of all the 10-round differentials for Rounds 26 to 35 with the output difference $(e_{7,10,19,21,26,30}, e_{32}, 0, 0, e_{7,21,26}, e_{32}, 0, 0)$.

These two related-key differential characteristics were used to construct a 34-round related-key rectangle distinguisher with probability $2^{-456.76}(= 2^{-74} \times 2^{-126.76} \times 2^{-256})$ for Rounds 2 to 35 of SHACAL-2. This was finally used in conjunction with an early abort technique to break the first 42 rounds of SHACAL-2.

8.5.2 A 35-Round Related-Key Rectangle Distinguisher with Probability $2^{-474.76}$

In 2007, Wang [104] described a way of extending the 34-round related-key rectangle distinguisher given in Section 8.5 to a 35-round distinguisher by appending a one-round related-key differential with probability 1 at the beginning. The differential requires the pair of plaintext blocks to satisfy certain properties; specifically, suppose $P = (A^0, B^0, C^0, D^0, E^0, F^0, G^0, H^0)$ and $\widetilde{P} = (\widehat{A}^0, \widehat{B}^0, \widehat{C}^0, \widehat{D}^0, \widehat{E}^0, \widehat{F}^0, \widehat{G}^0, \widehat{H}^0)$ satisfy:

$$\begin{array}{ll} a^0_{32} = b^0_{32},\ a^0_i = c^0_i, & \text{for } i = 7, 10, 19, 21, 26, 30; \\ b^0_{10} = \neg e^0_{10},\ a^0_i = \neg f^0_i, & \text{for } i = 20, 31; \\ e^0_i = 0, & \text{for } i = 19, 30, 31; \\ f^0_i = g^0_i, & \text{for } i = 10, 14, 20, \end{array} \tag{8.1}$$

where a^0_i denotes the ith bit of A^0, and so on.

The 35-round distinguisher is made up of the following two related-key differentials. The following 25-round related-key differential with probability 2^{-47} is used for Rounds 1 to 25: $(0, e_{7,10,19,21,26,30}, e_{32}, 0, e_{10,14,20}, e_{19,30}, e_{32}, \Delta') \rightarrow (e_{14,25,29}, 0, 0, 0, e_{14,25,29}, 0, 0, 0)$, where $\Delta' = \mathbf{\Phi}_1(E^0) \boxminus \mathbf{\Phi}_1(E^0 \oplus e_{10,14,20})$ and the key difference $K \oplus \widetilde{K} = (e_{32}, 0, 0, 0, 0, 0, 0, 0, 0, e_{32}, 0, 0, 0, 0, 0, 0)$. See Table 8.2 for more details.

The second differential making up the 35-round distinguisher is the 10-round differential with probability 2^{-65} described in Section 8.5.1.

Wang used this 35-round related-key rectangle distinguisher with probability $(2^{-46})^2 \times 2^{-126.76} \times 2^{-256} = 2^{-474.76}$, to break the first 43 rounds of SHACAL-2. However, as described below, there is a flaw in the complexity analysis for Wang's attack

Table 8.2: The 25-round related-key differential characteristic for Rounds 1 to 25

Round(i)	ΔA^{i-1}	ΔB^{i-1}	ΔC^{i-1}	ΔD^{i-1}	ΔE^{i-1}	ΔF^{i-1}	ΔG^{i-1}	ΔH^{i-1}	ΔK_i	Prob.
1	0	$e_{7,10,19,21,26,30}$	e_{32}	0	$e_{10,14,20}$	$e_{19,30}$	e_{32}	Δ'	e_{32}	1
2	0	0	$e_{7,10,19,21,26,30}$	e_{32}	0	$e_{10,14,20}$	$e_{19,30}$	e_{32}	0	2^{-11}
3	e_{32}	0	0	$e_{7,10,19,21,26,30}$	0	0	$e_{10,14,20}$	$e_{19,30}$	0	2^{-10}
4	0	e_{32}	0	0	$e_{7,21,26}$	0	0	$e_{10,14,20}$	0	2^{-7}
5	0	0	e_{32}	0	0	$e_{7,21,26}$	0	0	0	2^{-4}
6	0	0	0	e_{32}	0	0	$e_{7,21,26}$	0	0	2^{-3}
7	0	0	0	0	e_{32}	0	0	$e_{7,21,26}$	0	2^{-4}
8	0	0	0	0	0	e_{32}	0	0	0	2^{-1}
9	0	0	0	0	0	0	e_{32}	0	0	2^{-1}
10	0	0	0	0	0	0	0	e_{32}	e_{32}	1
11	0	0	0	0	0	0	0	0	0	1
$\vdots$				$\vdots$					$\vdots$	$\vdots$
24	0	0	0	0	0	0	0	0	0	1
25	0	0	0	0	0	0	0	0	$\cdot$	2^{-6}
output	$e_{14,25,29}$	0	0	0	$e_{14,25,29}$	0	0	0	/	/

algorithm, which makes the attack infeasible.

8.5.2.1 A Flaw in Wang's Attack

Wang [104] claimed that the probability that six or more quartets pass the filtering condition in Step 6 of the attack is about $\sum_{i=6}^{2^{31.76}}[\binom{2^{31.76}}{i}\cdot(2^{-32\times2})^i\cdot(1-2^{-32\times2})^{2^{31.76}-i}]\approx 2^{-202.93}$. It is thus expected that about $2^{448}\times 2^{-202.93}=2^{245.07}$ guesses for $((K_{37},\cdots,K_{43}),\,(K^*_{37},\cdots,K^*_{43}))$ will be output by Step 6. As a result, Step 7 (which involves finding the 512-bit cipher key by exhaustively searching for the remaining 288 bits using the guesses output by Step 6) will have a complexity of around $2^{533.07}$, i.e. significantly larger than 2^{512}. Therefore, the attack is less efficient than an exhaustive key search.

8.5.3 A 35-Round Related-Key Rectangle Distinguisher with Probability 2^{-460}

We next describe a novel 35-round related-key rectangle distinguisher for Rounds 1-35 of SHACAL-2. This distinguisher incorporates a novel 10-round differential characteristic for Rounds 26 to 35: $(0,0,e_{7,10,19,21,26,30},e_{32},0,e_{10,14,20},e_{19,20},e_{32})\to(e_{7,10,19,21,26,30},e_{32},0,0,e_{7,21,26},e_{32},0,0)$, which has a probability of 2^{-56}. See Table 8.3 for more details.

Table 8.3: The 10-round differential characteristic for Rounds 26 to 35

Round(i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	ΔF^i	ΔG^i	ΔH^i	Prob.
26	0	0	$e_{7,10,19,21,26,30}$	e_{32}	0	$e_{10,14,20}$	$e_{14,19,30}$	$e_{14,32}$	2^{-11}
27	e_{32}	0	0	$e_{7,10,19,21,26,30}$	0	0	$e_{10,14,20}$	$e_{14,19,30}$	2^{-14}
28	0	e_{32}	0	0	$e_{7,21,26}$	0	0	$e_{10,14,20}$	2^{-7}
29	0	0	e_{32}	0	0	$e_{7,21,26}$	0	0	2^{-4}
30	0	0	0	e_{32}	0	0	$e_{7,21,26}$	0	2^{-3}
31	0	0	0	0	e_{32}	0	0	$e_{7,21,26}$	2^{-4}
32	0	0	0	0	0	e_{32}	0	0	2^{-1}
33	0	0	0	0	0	0	e_{32}	0	2^{-1}
34	0	0	0	0	0	0	0	e_{32}	1
35	e_{32}	0	0	0	e_{32}	0	0	0	2^{-11}
output	$e_{7,10,19,21,26,30}$	e_{32}	0	0	$e_{7,21,26}$	e_{32}	0	0	/

It also uses Wang's 25-round related-key differential characteristic with probability 2^{-47}. This means that the new 35-round related-key rectangle distinguisher has a probability of at least $(2^{-46} \times 2^{-56})^2 \times 2^{-256} = 2^{-460}$ for the correct key, while it has a probability of $(2^{-256})^2 = 2^{-512}$ for a wrong key.

8.6 Attacking the First 44 Rounds of SHACAL-2

In this section we describe an attack on the first 44 rounds of SHACAL-2. This attack exploits the novel related-key rectangle distinguisher described in Section 8.5.3. As mentioned before, we assume that the two related user keys are K and $\widetilde{K}$ with the relationship $K \oplus \widetilde{K} = (e_{32}, 0, 0, 0, 0, 0, 0, 0, 0, e_{32}, 0, 0, 0, 0, 0, 0)$.

8.6.1 Preliminary Remarks

Property 8.1 allows us to break more rounds of the cipher than would otherwise be the case by using the early abort technique described in Section 4.3. Because of the properties of the key schedule of SHACAL-2, it is impossible to determine the subkey differences of the last few rounds (to be attacked) from the difference between the two related cipher keys; thus it is necessary to guess the two different unknown subkeys in every round in order to conduct an early abort. In previously described related-key rectangle attacks on reduced-round SHACAL-2, such as those given in [78, 104], this is achieved by first guessing both the round subkeys, then partially decrypting every remaining candidate quartet to get the corresponding quartet just before this round, and finally checking whether it meets the difference requirements. However, we observe that an early abort can be conducted by checking the two pairs out of a candidate quartet in a staged way, as described in Section 4.4.

This observation enables us to use the 35-round distinguisher in Section 8.5 to conduct a related-key rectangle attack on the first 44 rounds of SHACAL-2. The early abort technique described in Section 4.4 plays a crucial role in the efficiency of our attack; otherwise, we would only be able to break only the first 43 rounds of SHACAL-2.

8.6.2 Attack Description

The attack procedure is as follows.

1. Choose a structure S, which is defined to be a set of 2^{232} plaintexts $P_i = (A_i^0, B_i^0, C_i^0, D_i^0, E_i^0, F_i^0, G_i^0, H_i^0)$ under the condition given in equation (8.1), $(i = 1, 2, \cdots, 2^{232})$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{232} plaintexts encrypted with K; let C_i be the ciphertext corresponding to plaintext P_i.

2. Compute another structure $\widetilde{S}$, which contains 2^{232} plaintexts $\widetilde{P_i} = (\widehat{A}^0, \widehat{B}^0, \widehat{C}^0, \widehat{D}^0, \widehat{E}^0, \widehat{F}^0, \widehat{G}^0, \widehat{H}^0) = (A_i^0, B_i^0 \oplus e_{7,10,19,21,26,30}, C_i^0 \oplus e_{32}, D_i^0, E_i^0 \oplus e_{10,14,20}, F_i^0 \oplus e_{19,30}, G_i^0 \oplus e_{32}, H_i^0 \boxplus \mathbf{\Phi}_1(E_i^0) \boxminus \mathbf{\Phi}_1(E_i^0 \oplus e_{10,14,20}))$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{232} plaintexts in $\widetilde{S}$ encrypted with $\widetilde{K} = K \oplus (e_{32}, 0, 0, 0, 0, 0, 0, 0, 0, e_{32}, 0, 0, 0, 0, 0, 0)$; let $\widetilde{C_i}$ be the ciphertext corresponding to plaintext $\widetilde{P_i}$ (encrypted with $\widetilde{K}$).

3. Guess a 128-bit subkey pair for $((K_{41}, K_{42}, K_{43}, K_{44}), (\widetilde{K}_{41}, \widetilde{K}_{42}, \widetilde{K}_{43}, \widetilde{K}_{44}))$. Then, partially decrypt all the ciphertexts C_i through Rounds 44 to 41 using the subkeys $(K_{44}, K_{43}, K_{42}, K_{41})$ to get the corresponding values just before Round 41; let C_i^{40} be the partially decrypted version of C_i. Partially decrypt all the ciphertexts $\widetilde{C_i}$ through Rounds 44 to 41 using the subkeys $(\widetilde{K}_{44}, \widetilde{K}_{43}, \widetilde{K}_{42}, \widetilde{K}_{41})$ to get the corresponding values just before Round 41; let $\widetilde{C}_i^{40}$ be the partially decrypted version of $\widetilde{C_i}$. Keep $(C_i^{40}, \widetilde{C}_i^{40})$ in a hash table. This process produces about $\frac{2^{232\times 2}}{2} = 2^{463}$ candidate quartets $(C_{i_0}^{40}, \widetilde{C}_{i_0}^{40}, C_{i_1}^{40},$

$\widetilde{C}^{40}_{i_1}$), where $1 \leq i_0 \leq i_1 \leq 2^{232}$. By Property 8.1, we can deduce $(A^{35}_{i_0}, B^{35}_{i_0}, C^{35}_{i_0})$, $(A^{35}_{i_1}, B^{35}_{i_1}, C^{35}_{i_1})$, $(\widehat{A}^{35}_{i_0}, \widehat{B}^{35}_{i_0}, \widehat{C}^{35}_{i_0})$, $(\widehat{A}^{35}_{i_1}, \widehat{B}^{35}_{i_1}, \widehat{C}^{35}_{i_1})$, $D^{35}_{i_0} \boxminus D^{35}_{i_1}$, and $\widehat{D}^{35}_{i_0} \boxminus \widehat{D}^{35}_{i_1}$. Finally, choose only the quartets $(C^{40}_{i_0}, \widetilde{C}^{40}_{i_0}, C^{40}_{i_1}, \widetilde{C}^{40}_{i_1})$ such that $(A^{35}_{i_0}, B^{35}_{i_0}, C^{35}_{i_0}) \oplus (A^{35}_{i_1}, B^{35}_{i_1}, C^{35}_{i_1}) = (e_{7,10,19,21,26,30}, e_{32}, 0)$, $(\widehat{A}^{35}_{i_0}, \widehat{B}^{35}_{i_0}, \widehat{C}^{35}_{i_0}) \oplus (\widehat{A}^{35}_{i_1}, \widehat{B}^{35}_{i_1}, \widehat{C}^{35}_{i_1}) = (e_{7,10,19,21,26,30}, e_{32}, 0)$, and $D^{35}_{i_0} \boxminus D^{35}_{i_1} = \widehat{D}^{35}_{i_0} \boxminus \widehat{D}^{35}_{i_1} = 0$. If six or more quartets $(C^{40}_{i_0}, \widetilde{C}^{40}_{i_0}, C^{40}_{i_1}, \widetilde{C}^{40}_{i_1})$ pass this condition, store the quartet and the associated information; otherwise, repeat this step with another guess.

4. Perform Steps (a) and (b) below for every remaining quartet $(C^{40}_{i_0}, C^{40}_{i_1}, \widetilde{C}^{40}_{i_0}, \widetilde{C}^{40}_{i_1})$.

   (a) Guess a value for the subkey K_{40}. Partially decrypt $C^{40}_{i_0}$ and $C^{40}_{i_1}$ through Round 40 with K_{40} to get the corresponding values just before Round 40; we denote them by $C^{39}_{i_0}$ and $C^{39}_{i_1}$, respectively. Thus, we can compute $H^{38}_{i_0} \boxminus H^{38}_{i_1}$ by Property 8.1; since $H^{38}_{i} = E^{35}_{i}$, we choose the quartets $(C^{40}_{i_0}, C^{40}_{i_1}, \widetilde{C}^{40}_{i_0}, \widetilde{C}^{40}_{i_1})$ such that $H^{38}_{i_0} \boxminus H^{38}_{i_1} \in \{\pm 2^6 \pm 2^{20} \pm 2^{25} \bmod 2^{32}\}$. If six or more quartets $(C^{40}_{i_0}, C^{40}_{i_1}, \widetilde{C}^{40}_{i_0}, \widetilde{C}^{40}_{i_1})$ pass this condition, execute Step 4(b) with the quartets meeting this condition; otherwise, repeat this step with another guess for K_{40}.

   (b) Guess a value for the subkey $\widetilde{K}_{40}$. Partially decrypt $\widetilde{C}^{40}_{i_0}$ and $\widetilde{C}^{40}_{i_1}$ through Round 40 with $\widetilde{K}_{40}$ to get the corresponding values just before Round 40; we denote them by $\widetilde{C}^{39}_{i_0}$ and $\widetilde{C}^{39}_{i_1}$, respectively. Similarly, we choose only the quartets $(C^{40}_{i_0}, C^{40}_{i_1}, \widetilde{C}^{40}_{i_0}, \widetilde{C}^{40}_{i_1})$ such that $\widehat{H}^{38}_{i_0} \boxminus \widehat{H}^{38}_{i_1} \in \{\pm 2^6 \pm 2^{20} \pm 2^{25} \bmod 2^{32}\}$. If six or more quartets $(C^{40}_{i_0}, C^{40}_{i_1}, \widetilde{C}^{40}_{i_0}, \widetilde{C}^{40}_{i_1})$ pass this condition, execute Step 5 with the quartets $(C^{39}_{i_0}, C^{39}_{i_1}, \widetilde{C}^{39}_{i_0}, \widetilde{C}^{39}_{i_1})$ that meet this condition; otherwise, repeat this step with another guess for $\widetilde{K}_{40}$.

5. Perform Steps (a) and (b) below for every remaining quartet $(C^{39}_{i_0}, C^{39}_{i_1}, \widetilde{C}^{39}_{i_0}, \widetilde{C}^{39}_{i_1})$.

   (a) Guess a value for the subkey K_{39}. Partially decrypt $C^{39}_{i_0}$ and $C^{39}_{i_1}$ through Round 39 with K_{39} to get the corresponding values just before Round 39; we denote them by $C^{38}_{i_0}$ and $C^{38}_{i_1}$, respectively. Thus, we can compute $E^{35}_{i_0}$, $E^{35}_{i_1}$, and $H^{37}_{i_0} \boxminus H^{37}_{i_1}$. We choose only the quartets $(C^{39}_{i_0}, C^{39}_{i_1}, \widetilde{C}^{39}_{i_0}, \widetilde{C}^{39}_{i_1})$ such that $E^{35}_{i_0} \oplus E^{35}_{i_1} = e_{7,21,26}$ and $H^{37}_{i_0} \boxminus H^{37}_{i_1} \in \{\pm 2^{31} \bmod 2^{32}\}$. If six or more quartets $(C^{39}_{i_0}, C^{39}_{i_1}, \widetilde{C}^{39}_{i_0}, \widetilde{C}^{39}_{i_1})$ pass this condition, execute Step 5(b) with the quartets meeting this condition; otherwise, repeat this step with another guess for K_{39}.

(b) Guess a value for the subkey $\widetilde{K}_{39}$. Partially decrypt $\widetilde{C}_{i_0}^{39}$ and $\widetilde{C}_{i_1}^{39}$ through Round 39 with $\widetilde{K}_{39}$ to get the corresponding values just before Round 39; we denote them by $\widetilde{C}_{i_0}^{38}$ and $\widetilde{C}_{i_1}^{38}$, respectively. Thus, we can compute $\widehat{E}_{i_0}^{35}$, $\widehat{E}_{i_1}^{35}$, and $\widehat{H}_{i_0}^{37} \boxminus \widehat{H}_{i_1}^{37}$. We choose only the quartets $(C_{i_0}^{40}, C_{i_1}^{40}, \widetilde{C}_{i_0}^{40}, \widetilde{C}_{i_1}^{40})$ such that $\widehat{E}_{i_0}^{35} \oplus \widehat{E}_{i_1}^{35} = e_{7,21,26}$ and $\widehat{H}_{i_0}^{37} \boxminus \widehat{H}_{i_1}^{37} \in \{\pm 2^{31} \bmod 2^{32}\}$. If six or more quartets $(C_{i_0}^{39}, C_{i_1}^{39}, \widetilde{C}_{i_0}^{39}, \widetilde{C}_{i_1}^{39})$ pass this test, execute Step 6 with the quartets $(C_{i_0}^{38}, C_{i_1}^{38}, \widetilde{C}_{i_0}^{38}, \widetilde{C}_{i_1}^{38})$ that meet this condition; otherwise, repeat this step with another guess for $\widetilde{K}_{39}$.

6. Perform Steps (a) and (b) below for every remaining quartet $(C_{i_0}^{38}, C_{i_1}^{38}, \widetilde{C}_{i_0}^{38}, \widetilde{C}_{i_1}^{38})$.

(a) Guess a value for the subkey K_{38}. Partially decrypt $C_{i_0}^{38}$ and $C_{i_1}^{38}$ through Round 38 with K_{38} to get the corresponding values just before Round 38; we denote them by $C_{i_0}^{37}$ and $C_{i_1}^{37}$, respectively. Thus, we can compute $F_{i_0}^{35}$, $F_{i_1}^{35}$, and $H_{i_0}^{36} \boxminus H_{i_1}^{36}$. We choose only the quartets $(C_{i_0}^{38}, C_{i_1}^{38}, \widetilde{C}_{i_0}^{38}, \widetilde{C}_{i_1}^{38})$ such that $F_{i_0}^{35} \oplus F_{i_1}^{35} = e_{32}$ and $H_{i_0}^{36} \boxminus H_{i_1}^{36} = 0$. If six or more quartets $(C_{i_0}^{38}, C_{i_1}^{38}, \widetilde{C}_{i_0}^{38}, \widetilde{C}_{i_1}^{38})$ pass this condition, execute Step 6(b) with the quartets meeting this condition; otherwise, repeat this step with another guess for K_{38}.

(b) Guess a value for the subkey $\widetilde{K}_{38}$. Partially decrypt $\widetilde{C}_{i_0}^{38}$ and $\widetilde{C}_{i_1}^{38}$ through Round 38 with $\widetilde{K}_{38}$ to get the corresponding values just before Round 38; we denote them by $\widetilde{C}_{i_0}^{37}$ and $\widetilde{C}_{i_1}^{37}$, respectively. Thus, we can compute $\widehat{F}_{i_0}^{35}$, $\widehat{F}_{i_1}^{35}$, and $\widehat{H}_{i_0}^{36} \boxminus \widehat{H}_{i_1}^{36}$. We choose only the quartets $(C_{i_0}^{38}, C_{i_1}^{38}, \widetilde{C}_{i_0}^{38}, \widetilde{C}_{i_1}^{38})$ such that $\widehat{F}_{i_0}^{35} \oplus \widehat{F}_{i_1}^{35} = e_{32}$ and $\widehat{H}_{i_0}^{36} \boxminus \widehat{H}_{i_1}^{36} = 0$. If six or more quartets $(C_{i_0}^{38}, C_{i_1}^{38}, \widetilde{C}_{i_0}^{38}, \widetilde{C}_{i_1}^{38})$ pass this test, execute Step 7 with the quartets $(C_{i_0}^{37}, C_{i_1}^{37}, \widetilde{C}_{i_0}^{37}, \widetilde{C}_{i_1}^{37})$ that meet this condition; otherwise, repeat this step with another guess for $\widetilde{K}_{38}$.

7. Perform Steps (a) and (b) below for every remaining quartet $(C_{i_0}^{37}, C_{i_1}^{37}, \widetilde{C}_{i_0}^{37}, \widetilde{C}_{i_1}^{37})$.

(a) Guess a value for the subkey K_{37}. Partially decrypt $C_{i_0}^{37}$ and $C_{i_1}^{37}$ through Round 37 with K_{37} to get the corresponding values just before Round 37; we denote them by $C_{i_0}^{36}$ and $C_{i_1}^{36}$, respectively. Thus, we can compute $H_{i_0}^{35} \boxminus H_{i_1}^{35}$. We choose only the quartets $(C_{i_0}^{37}, C_{i_1}^{37}, \widetilde{C}_{i_0}^{37}, \widetilde{C}_{i_1}^{37})$ such that $H_{i_0}^{35} \boxminus H_{i_1}^{35} = 0$. If six or more quartets $(C_{i_0}^{37}, C_{i_1}^{37}, \widetilde{C}_{i_0}^{37}, \widetilde{C}_{i_1}^{37})$ pass this condition, execute Step 7(b) with the quartets meeting this condition; otherwise, repeat this step with another guess for K_{37}.

(b) Guess a value for the subkey $\widetilde{K}_{37}$. Partially decrypt $\widetilde{C}^{37}_{i_0}$ and $\widetilde{C}^{37}_{i_1}$ through Round 37 with $\widetilde{K}_{37}$ to get the corresponding values just before Round 37; we denote them by $\widetilde{C}^{36}_{i_0}$ and $\widetilde{C}^{36}_{i_1}$, respectively. Thus, we can compute $\widehat{H}^{35}_{i_0} \boxminus \widehat{H}^{35}_{i_1}$. We choose only the quartets $(C^{37}_{i_0}, C^{37}_{i_1}, \widetilde{C}^{37}_{i_0}, \widetilde{C}^{37}_{i_1})$ such that $\widehat{H}^{35}_{i_0} \boxminus \widehat{H}^{35}_{i_1} = 0$. If six or more quartets $(C^{37}_{i_0}, C^{37}_{i_1}, \widetilde{C}^{37}_{i_0}, \widetilde{C}^{37}_{i_1})$ pass this test, then record $(K_{37}, K_{38}, \cdots, K_{44})$, and execute Step 8; otherwise, repeat this step with another guess for $\widetilde{K}_{37}$.

8. For a recorded value for $(K_{37}, K_{38}, \cdots, K_{44})$, exhaustively search for the remaining 256 bits using one known pair of plaintext and ciphertext. If a 512-bit key is suggested, output it as the user key of the 44-round SHACAL-2; otherwise, repeat Step 3 with another guess.

8.6.3 Complexity Analysis

This attack requires 2^{233} related-key chosen plaintexts. The required memory for this attack is dominated by the ciphertexts, which is approximately $2^{233} \times 32 \approx 2^{238}$ memory bytes.

Step 3 has a time complexity of about $2 \times 2^{232} \times 2^{32\times 8} \times \frac{8}{44} \approx 2^{486.54}$ 44-round SHACAL-2 encryptions, and it also requires about $2^{32\times 8} \times 2^{232} = 2^{488}$ memory accesses, which is negligible compared with the $2^{486.54}$ encryptions. Due to the 128-bit filtering condition in Step 3, it is expected that only about $2^{463} \times (2^{-128})^2 = 2^{207}$ candidate quartets remain after Step 3 for every key guess.

Step 4(a) has a time complexity about $2 \times 2^{207} \times 2^{32\times 9} \times \frac{1}{44} \approx 2^{490.54}$ encryptions. There is a filtering condition of $\frac{2^3}{2^{32}} = 2^{-29}$ in either of Steps 4(a) and (b). In Step 4(a), the probability that 6 or more quartets pass the test for a wrong guess is about 1, thus it follows that all the 2^{288} key guesses pass this step; and about $2^{207} \times 2^{-29} = 2^{178}$ candidate quartets remain after this step for every key guess. Step 4(b) has a time complexity about $2 \times 2^{178} \times 2^{32\times 10} \times \frac{1}{44} \approx 2^{493.54}$ encryptions. In Step 4(b), the probability that 6 or more quartets pass the test for a wrong guess is also about 1, thus it follows that all the 2^{320} key guesses pass this step; and about $2^{178} \times 2^{-29} = 2^{149}$ candidate quartets remain after this step for every key guess.

Step 5(a) has a time complexity about $2 \times 2^{149} \times 2^{32\times 11} \times \frac{1}{44} \approx 2^{496.54}$ encryptions. There is a filtering condition of $\frac{2}{2^{32}} \times \frac{1}{2^3} = 2^{-34}$ in either of Steps 5(a) and (b). In Step 5(a), the probability that 6 or more quartets pass the test for a wrong guess is about 1, so it follows that all the 2^{352} key guesses pass this step; and about $2^{149} \times 2^{-34} = 2^{115}$ candidate quartets remain after this step for every key guess. Step 5(b) has a time complexity about $2 \times 2^{115} \times 2^{32\times 12} \times \frac{1}{44} \approx 2^{494.54}$ encryptions. In Step 5(b), since the probability that 6 or more quartets pass the test for a wrong guess is also about 1, it follows that all the 2^{384} key guesses pass this step; and about $2^{115} \times 2^{-34} = 2^{81}$ candidate quartets remain after this step for every key guess.

Step 6(a) has a time complexity about $2 \times 2^{81} \times 2^{32\times 13} \times \frac{1}{44} \approx 2^{492.54}$ encryptions. There is a filtering condition of $\frac{1}{2^{32}} \times \frac{1}{2} = 2^{-33}$ in either of Steps 6(a) and (b). In Step 6(a), the probability that 6 or more quartets pass the test for a wrong guess is about 1 as well, thus it follows that all the 2^{416} key guesses pass this step; and about $2^{81} \times 2^{-33} = 2^{48}$ candidate quartets remain after this step for every key guess. Step 6(b) has a time complexity about $2 \times 2^{48} \times 2^{32\times 14} \times \frac{1}{44} \approx 2^{491.54}$ encryptions. In Step 6(b), the probability that 6 or more quartets pass the test for a wrong guess is about 1, thus it follows that all the 2^{448} key guesses pass this step; and about $2^{48} \times 2^{-33} = 2^{15}$ candidate quartets remain after this step for every key guess.

Step 7(a) has a time complexity about $2 \times 2^{15} \times 2^{32\times 15} \times \frac{1}{44} \approx 2^{490.54}$ encryptions. There is a filtering condition of 2^{-32} in either of Steps 7(a) and (b). In Step 7(a), the probability that six or more quartets pass the test for a wrong guess is about $\sum_{i=6}^{2^{15}}[\binom{2^{15}}{i} \cdot (2^{-32})^i \cdot (1-2^{-32})^{2^{15}-i}] \approx 2^{-111.49}$, thus it follows that about the $2^{480} \times 2^{-111.49} = 2^{368.51}$ key guesses pass this step. Step 7(b) has a time complexity about $2 \times 2^{368.51} \times 6 \times \frac{1}{44} \approx 2^{366.63}$ encryptions. In Step 7(b), the probability that six or more quartets pass the test for a wrong guess is about $(2^{-32})^6 = 2^{-192}$, so it is expected that only about $2^{368.51+32} \times 2^{-192} = 2^{208.51}$ guesses of $(K_{37}, K_{38}, \cdots, K_{44})$ pass Step 7(b), which result in $2^{464.51}$ trials in Step 8.

Therefore, the attack has a total time complexity of about $2^{497.2}$ 44-round SHACAL-2 encryptions.

As about 2^{463} quartets are tested in this attack and the 35-round related-key rectangle distinguisher has a probability of 2^{-460}, we can learn that the expected number of the qualified quartets for the correct key guess in Step 7(b) is about $2^{463} \times 2^{-460} = 8$. The probability that six or more quartets pass Step 7(b) is $\sum_{i=6}^{2^{463}}[\binom{2^{463}}{i} \cdot (2^{-460})^i \cdot (1-2^{-460})^{2^{463}-i}] \approx 0.8$, therefore the related-key rectangle attack works with a success probability of 80%.

8.7 Summary

In this chapter we have presented a related-key rectangle attack on 44 rounds of SHACAL-2. This is better than any previously published cryptanalytic result on SHACAL-2 in terms of the number of attacked rounds. Table 8.4 summarises the published cryptanalytic results on the 512-bit key version of SHACAL-2, where CP and RK-CP refer to the required numbers of chosen plaintexts and related-key chosen plaintexts, respectively; and Encryptions refers to the required number of encryption operations of the relevant reduced-round version of SHACAL-2.

Table 8.4: Cryptanalytic results on the 512-bit key version of SHACAL-2

Attack Type	Rounds	Data	Time	Source
Impossible differential	30	744CP	$2^{495.1}$Encryptions	[39]
Square-nonlinear	28	$2^{40.9}$CP	$2^{494.1}$Encryptions	[98]
Differential-nonlinear	32	$2^{43.4}$CP	$2^{504.2}$Encryptions	[98]
Related-key differential-nonlinear	35	$2^{42.4}$RK-CP	$2^{452.1}$Encryptions	[54]
Related-key rectangle	37	$2^{235.2}$RK-CP	2^{487}Encryptions	[54]
	42	$2^{243.4}$RK-CP	$2^{488.4}$Encryptions	[78]
	43†	$2^{240.4}$RK-CP	$2^{480.4}$Encryptions	[104]
	44	2^{233}RK-CP	$2^{497.2}$Encryptions	Section 8.6

†: there is a flaw, as shown in Section 8.5.2.

CHAPTER 9

Related-Key Rectangle Attack on 36-Round XTEA

XTEA is a 64-bit block cipher with a 128-bit user key. In this chapter, we present a related-key rectangle attack on 36 rounds of XTEA; the attack requires $2^{64.98}$ *related-key chosen plaintexts, and has a time complexity of* $2^{126.3}$ *encryptions. This is better than any previously published cryptanalytic results on XTEA in terms of the number of attacked rounds.*

Contents

9.1 Introduction

The block cipher TEA (Tiny Encryption Algorithm) was designed by Wheeler and Needham [106] in 1994 as a short C language program that would run safely on most machines. It has no preset tables or long set up times, and achieves a high performance by performing all its operations on 32-bit words, using only exclusive-or, addition modulo 2^{32}, multiplication modulo 2^{32} and shifts. TEA has a simple Feistel structure, but uses a large number (i.e. 64) of rounds to achieve the desired level of security. Although it was originally written in C, TEA can readily be implemented in a range of languages, including assembler. However, taking advantage of its simple key schedule, in 1997 Kelsey, Schneier and Wagner [50] described a related-key attack. To secure TEA against related-key attacks, Needham and Wheeler [88] presented an extended version of TEA in 1997, known as XTEA, which retains the original objectives of simplicity and efficiency.

In this chapter, we describe a 24-round related-key rectangle distinguisher with probability $2^{-124.92}$ for XTEA. We then apply it to mount a related-key rectangle attack on 36 rounds of XTEA, using the early abort technique described in Section 4.3. The attack requires $2^{64.98}$ related-key chosen plaintexts and has a time complexity of $2^{126.3}$ 36-round XTEA computations.

The remainder of this chapter is organised as follows. In Section 9.2 we describe XTEA. In Section 9.3 we briefly review previous cryptanalytic results on XTEA. In Section 9.4 we give the 24-round related-key rectangle distinguisher for XTEA. In 9.5 we present our cryptanalytic results on XTEA. Section 9.6 summarises the results of this chapter.

9.2 The XTEA Block Cipher

In this section we briefly describe the XTEA block cipher [88].

9.2.1 Notation

In this chapter, the bits of a 32-bit value are numbered from 1 to 32 from left to right, where the least significant bit is referred as the 1st bit, and the most significant bit is referred as the 32nd bit. We use the following notation.

- $\boxplus$: addition modulo 2^{32}
- $\boxtimes$: multiplication modulo 2^{32}
- e_j: a 32-bit word with zeros everywhere except for bit position j, $(1 \leq j \leq 32)$
- $e_{i_1,\cdots,i_j}$: the 32-bit word equal to $e_{i_1} \oplus \cdots \oplus e_{i_j}$, $(1 \leq i_1, \cdots, i_j \leq 32)$
- $e_{j,\sim}$: a 32-bit word that has zeros in bit positions 1 to $j-1$, a one in bit position j and indeterminate values in bit positions $(j+1)$ to 32, $(1 \leq j \leq 31)$
- $\star$: an arbitrary 32-bit value, where two values represented by the $\star$ symbol may be different
- ρ^l_j: an l-bit value with zeros everywhere except for bit position j, $(1 \leq j \leq l)$
- $\rho^l_{j,\sim}$: an l-bit value that has zeros in bit positions 1 to j, a one in bit position j and indeterminate values in the remaining positions, $(1 \leq j \leq l)$

9.2.2 Generation of Subkeys

XTEA uses a total of 64 32-bit subkeys K_i, $(1 \leq i \leq 64)$, all derived from a 128-bit key K. Let K be represented as a sequence of four 32-bit words $K = (W_1, W_2, W_3, W_4)$, then $K_i = W_{(\lfloor \frac{i}{2} \rfloor \boxtimes\ \theta >>11)\&3}$, where $\theta = 0x9e3779b9$. Table 9.1 lists the set of subkey values.

Table 9.1: The key schedule of XTEA

Round(i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
K_i	W_1	W_4	W_2	W_3	W_3	W_2	W_4	W_1	W_1	W_1	W_2	W_4	W_3	W_3	W_4	W_2
Round(i)	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
K_i	W_1	W_1	W_2	W_1	W_3	W_4	W_4	W_3	W_1	W_2	W_2	W_2	W_3	W_1	W_4	W_4
Round(i)	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
K_i	W_1	W_3	W_2	W_2	W_3	W_2	W_4	W_1	W_1	W_4	W_2	W_3	W_3	W_2	W_4	W_2
Round(i)	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
K_i	W_1	W_1	W_2	W_4	W_3	W_3	W_4	W_3	W_1	W_2	W_2	W_1	W_3	W_4	W_4	W_3

9.2.3 Encryption Procedure

XTEA takes as input a 64-bit plaintext block P, and has a total of 64 rounds. Its encryption procedure is as follows, where L_i and R_i are 32-bit variables.

1. Represent P as two 32-bit words $P = (L_0, R_0)$.
2. For $i = 1$ to 64:

   $R_i = L_{i-1} \boxplus (((R_{i-1} << 4 \oplus R_{i-1} >> 5) \boxplus R_{i-1}) \oplus (\lfloor \frac{i}{2} \rfloor \boxtimes \theta \boxplus K_i)),$

   $L_i = R_{i-1};$
3. Ciphertext $= (L_{64}, R_{64})$.

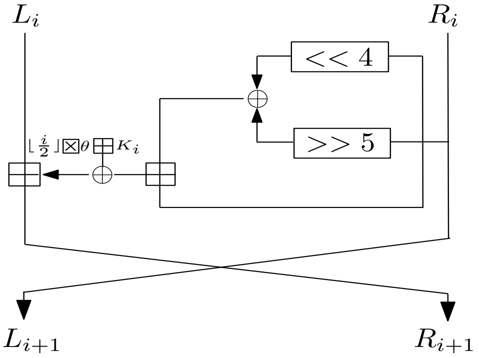


Figure 9.1: The ith encryption round of XTEA

The ith iteration of Step 2 in the above description is referred to below as Round i, $(1 \leq i \leq 64)$. Figure 9.1 depicts such a round.

Let $\widetilde{K}_i = (\lfloor \frac{i}{2} \rfloor \times \theta) \boxplus K_i$, $(1 \leq i \leq 64)$. We write $\widetilde{K}_{i,[l_1,l_2]}$ for bits $(l_1, \cdots, l_2)$ of $\widetilde{K}_i$, where $1 \leq l_1 \leq l_2 \leq 32$.

9.3 Previous Cryptanalytic Results

In this section we briefly review previously published cryptanalytic attacks on XTEA.

- In 2002, Moon, Hwang, Lee, Lee and Lim [86] presented an impossible differential attack on 14 rounds of XTEA.
- In 2003, Hong, Hong, Ko, Chang, Lee and Lee [38] presented a differential attack on 15 rounds of XTEA and a truncated differential attack on 23 rounds of XTEA, where the former attack uses a 13-round differential with probability $2^{-54.795}$, and the latter attack uses an 8-round truncated differential.

- In 2004, Ko, Hong, Lee, Lee and Kang [61] presented a related-key truncated differential attack on 27 rounds of XTEA, based on the 8-round truncated differential of Hong et al.
- In 2006, Lee, Hong, Chang, Hong and Lim [70] presented a related-key rectangle attack on 34 rounds of XTEA that works under the assumption that the key used is a member of a special class of weak keys.

In summary, the related-key truncated differential attack on 27-round XTEA of Ko et al. [61] is the best previously published cryptanalytic result on XTEA without making a weak key assumption.

9.4 A 24-Round Related-Key Rectangle Distinguisher with Probability $2^{-124.92}$

In this section, we describe a novel 24-round related-key rectangle distinguisher for XTEA.

The definition of a related-key rectangle distinguisher requires the part of the cipher $\mathbf{E}$ concerned to be decomposed into two sub-ciphers $\mathbf{E}^0$ and $\mathbf{E}^1$. Let $\mathbf{E}^0$ denote Rounds 21 to 36 of XTEA, and $\mathbf{E}^1$ denote Rounds 37 to 44 of XTEA. To define the distinguisher we need to specify related-key differentials for $\mathbf{E}^0$ and $\mathbf{E}^1$. The 24-round related-key rectangle distinguisher involves four cipher keys (TYPE 1 as described in Section 2.2.9), which we assume are K_A, K_B, K_C, K_D.

The first related-key differential making up the 24-round distinguisher is the following related-key differential $\Delta\alpha \rightarrow \Delta\beta$ with probability $2^{-32.49}$ for $\mathbf{E}^0$: $(e_{22,27,31}, e_{27}) \rightarrow (e_{12,17,21}, e_{7,25,27})$, where the relationship between the four cipher keys is $K_A \oplus K_B = K_C \oplus K_D = (0, 0, 0, e_{32})$. See Table 9.2 for further details of this differential. During the calculations of the probability of this related-key differential, we use the general result described in Theorem 8.1.

The second related-key differential making up the 24-round distinguisher is the following related-key differential $\Delta\gamma \rightarrow \Delta\delta$ with probability 1 for $\mathbf{E}^1$: $(e_{32}, 0) \rightarrow$

Table 9.2: The first related-key differential in the 24-round related-key rectangle distinguisher

Round(i)	$(\Delta L_{i-1}, \Delta R_{i-1})$	ΔK_i	Prob.	Round(i)	$(\Delta L_{i-1}, \Delta R_{i-1})$	ΔK_i	Prob.
21	$(e_{22,27,31}, e_{27})$	0	$2^{-4.16}$	31	$(0,0)$	e_{32}	1
22	(e_{27}, e_{32})	e_{32}	$2^{-1.52}$	32	$(0, e_{32})$	e_{32}	$2^{-1.52}$
23	$(e_{32}, 0)$	e_{32}	1	33	(e_{32}, e_{27})	0	$2^{-4.16}$
24	$(0,0)$	0	1	34	$(e_{27}, e_{22,27,31,32})$	0	$2^{-5.15}$
⋮	⋮	⋮	⋮	35	$(e_{22,27,31,32}, e_{17,27})$	0	$2^{-8.31}$
29	$(0,0)$	0	1	36	$(e_{17,27}, e_{12,17,21})$	0	$2^{-7.67}$
30	$(0,0)$	0	1	*output*	$(e_{12,17,21}, e_{7,25,27})$	/	/

$(0, e_{32})$, where the relationship between the four cipher keys is $K_A \oplus K_C = K_B \oplus K_D = (0, 0, e_{32}, 0)$.

In the following, we need to sum the squares of the probabilities of all the possible 16-round differentials $\Delta\alpha \to \Delta\beta^*$ with the same input difference α to $\mathbf{E}^0$, which is computationally infeasible. To address this problem, we just count some of those in which only the last one-round (Case A), two-round (Case B) or five-round (Case C) related-key differential characteristic is different from the 16-round related-key differential $\Delta\alpha \to \Delta\beta$ in Table 9.2:

Case A: The last one-round (i.e. Round 36) related-key differential characteristic has the form $(e_{17,27}, e_{12,17,21}) \to (e_{12,17,21}, \Delta R_{37})$. From an analysis of this one-round differential, we know that there exists at least 1 possible ΔR_{37} (i.e. $e_{7,25,27}$) with a lower bound probability of $2^{-7.67}$, at least 4 possible ΔR_{37} (i.e. $e_{7,8,25,27}, e_{7,18,25,27}, e_{7,25,26,27}, e_{7,25,26}$) with a lower bound probability of $2^{-8.67} + 2^{-9.72} \approx 2^{-8.10}$, at least 7 possible ΔR_{37} (i.e. $e_{7,8,9,25,27}, e_{7,8,18,25,27}, e_{7,8,25,26,27}, e_{7,8,25,26}, e_{7,18,19,25,27}, e_{7,18,25,26,27}, e_{7,18,25,26}$) with a lower bound probability of $2^{-9.67} + 2^{-11.86} \approx 2^{-9.38}$, at least 2 possible ΔR_{37} (i.e. $e_{7,25,26,28}, e_{7,25,26,27,28}$) with a lower bound probability of $2^{-9.67} + 2^{-12.85} \approx 2^{-9.52}$, at least 10 possible ΔR_{37} with a lower bound probability of $2^{-10.67}$, at least 15 possible ΔR_{37} with a lower bound probability of $2^{-11.67}$, at least 21 possible ΔR_{37} with a lower bound probability of $2^{-12.67}$ and at least 28 possible ΔR_{37} with a lower bound probability of $2^{-13.67}$. Thus, we can compute a square sum of at least $2^{-7.67\times 2} + 4 \times 2^{-8.1\times 2} + 7 \times 2^{-9.38\times 2} + 2 \times 2^{-9.52\times 2} + 10 \times 2^{-10.67\times 2} + 15 \times 2^{-11.67\times 2} + 21 \times 2^{-12.67\times 2} + 28 \times 2^{-13.67\times 2} \approx 2^{-13.25}$ for the probabilities of the one-round differentials $(e_{17,27}, e_{12,17,21}) \to (e_{12,17,21}, \Delta R_{37})$.

Case B: The last two-round (i.e. Rounds 35 and 36) related-key differential characteristic has the form $(e_{22,27,31,32}, e_{17,27}) \rightarrow (e_{17,27}, \Delta R_{36}) \rightarrow (\Delta R_{36}, \Delta R_{37})$. Here, we only consider $\Delta R_{36} \in \{e_{12,17,21}, e_{12,17,21,22}, e_{12,17,21,32}, e_{12,17,21,22,32}\}$; after an analysis we can learn that these four possibilities of ΔR_{36} have the same probability $2^{-8.31}$ for the one-round differential $(e_{22,27,31,32}, e_{17,27}) \rightarrow (e_{17,27}, \Delta R_{36})$. Similar to that described in Case A, we can compute a square sum of at least $2^{-14.04}$ for the case $\Delta R_{36} = e_{12,17,21,32}$, a square sum of at least $2^{-15.55}$ for the case $\Delta R_{36} = e_{12,17,21,22}$ and a square sum of at least $2^{-16.26}$ for the case $\Delta R_{36} = e_{12,17,21,22,32}$.

Case C: The last five-round (i.e. Rounds 32 to 36) related-key differential characteristic has the form $(0, e_{32}) \rightarrow (e_{32}, \Delta R_{33}) \rightarrow (\Delta R_{33}, \Delta R_{34}) \rightarrow (\Delta R_{34}, \Delta R_{35}) \rightarrow (\Delta R_{35}, \Delta R_{36}) \rightarrow (\Delta R_{36}, \Delta R_{37})$. Here, we only consider $(\Delta R_{34}, \Delta R_{35}) \in \{(e_{22,27,31,32}, e_{17,27}), (e_{22,27,31,32}, e_{17,27,32}), (e_{22,27,31}, e_{17,32}), (e_{22,27,31}, e_{17})\}$; we can know that the four possibilities of $(\Delta R_{34}, \Delta R_{35})$ have the same probability of at least $2^{-10.83} + 2^{-13.55} + 2^{-17.56} \approx 2^{-10.62}$ for the three-round differential $(0, e_{32}) \rightarrow (\Delta R_{34}, \Delta R_{35})$. Subsequently, a detailed analysis reveals that the one-round differential $(\Delta R_{34}, \Delta R_{35}) \rightarrow (\Delta R_{35}, \Delta R_{36})$ has a probability of at least $2^{-8.31}$ for the eight cases $\Delta R_{34} = e_{22,27,31,32}$ and $(\Delta R_{35}, \Delta R_{36}) \in \{(e_{17,27}, e_{12,17,21}), (e_{17,27}, e_{12,17,21,22}), (e_{17,27}, e_{12,17,21,32}), (e_{17,27}, e_{12,17,21,22,32}), (e_{17,27,32}, e_{12,17,21,27}), (e_{17,27,32}, e_{12,17,21,22,27}), (e_{17,27,32}, e_{12,17,21,27,32}), (e_{17,27,32}, e_{12,17,21,22,27,32})\}$, and has a probability of at least $2^{-7.46}$ for the eight cases $\Delta R_{34} = e_{22,27,31}$ and $(\Delta R_{35}, \Delta R_{36}) \in \{(e_{17}, e_{12,17,21,27,31}), (e_{17}, e_{12,17,21,22,27,31}), (e_{17}, e_{12,17,21,27,31,32}), (e_{17}, e_{12,17,21,22,27,31,32}), (e_{17,32}, e_{12,17,21,31}), (e_{17,32}, e_{12,17,21,22,31}), (e_{17,32}, e_{12,17,21,31,32}), (e_{17,32}, e_{12,17,21,22,31,32})\}$. Then, similar to that described in Case A, for the one-round differentials $(\Delta R_{35}, \Delta R_{36}) \rightarrow (\Delta R_{36}, \Delta R_{37})$, we can compute a square sum of at least $2^{-17.11}$ for the probabilities of the differentials from either of the two cases $(\Delta R_{34}, \Delta R_{35}, \Delta R_{36}) \in \{(e_{22,27,31,32}, e_{17,27,32}, e_{12,17,21,27}), (e_{22,27,31,32}, e_{17,27,32}, e_{12,17,21,27,32})\}$, a square sum of at least $2^{-18.13}$ for the probabilities of the differentials from either of the two cases $(\Delta R_{34}, \Delta R_{35}, \Delta R_{36}) \in \{(e_{22,27,31,32}, e_{17,27,32}, e_{12,17,21,22,27}), (e_{22,27,31,32}, e_{17,27,32}, e_{12,17,21,22,27,32})\}$, and a square sum of at least $2^{-18.22}$ for the probabilities of the differentials from each of the eight cases with $\Delta R_{34} = e_{22,27,31}$.

Thus, with the three cases above, we can compute a square sum for the probabilities of the differentials $\alpha \rightarrow \beta^*$ of at least $(2^{-4.16} \times 2^{-1.52} \times 2^{-10.62})^2 \times (2^{-8.31\times 2} \times 2^{-13.25} + 2^{-8.31\times 2} \times 2^{-14.04} + 2^{-8.31\times 2} \times 2^{-15.55} + 2^{-8.31\times 2} \times 2^{-16.26} + 2 \times 2^{-8.31\times 2} \times 2^{-17.11} + 2 \times 2^{-8.31\times 2} \times 2^{-18.13} + 8 \times 2^{-7.46\times 2} \times 2^{-18.22}) \approx 2^{-60.92}$.

As the 8-round related-key differential $\Delta\gamma \rightarrow \Delta\delta$ for $\mathbf{E}_1$ has a probability of 1, this distinguisher has a probability of at least $\sum_{\beta^*}[\Pr(\Delta\alpha \rightarrow \Delta\beta^*)^2 \times 2^{-64}] = 2^{-60.92} \times 2^{-64} = 2^{-124.92}$ for the correct key, while it has a probability of $(2^{-64})^2 = 2^{-128}$ for a wrong key.

9.5 Attacking Rounds 16 to 51 of XTEA

In this section we describe a related-key rectangle attack on 36 rounds of XTEA.

9.5.1 Preliminary Results

We first give three properties of XTEA.

The following result follows from inspection of Table 9.1.

Property 9.1 *In the key schedule of XTEA, only 64 user key bits* (W_1, W_2) *are used in Rounds 16 to 20 and 48 to 51.*

The following property follows from the structure of the XTEA round function.

Property 9.2 *Suppose two blocks are encrypted using XTEA with a pair of keys for which the subkeys for Rounds* $i, i+1, i+2$ *and* $i+3$ *are either the same or differ by* e_{32} *(for some* i*,* $1 \leq i \leq 61$*). Then, if the difference just after Round* i *is* $(0, e_{32})$*, then the difference just after Round* $(i+1)$ *has the form* $(e_{32}, e_{27,\sim})$*, the difference just after Round* $(i+2)$ *has the form* $(e_{27,\sim}, e_{22,\sim})$*, and the difference just after Round* $(i+3)$ *has the form* $(e_{22,\sim}, e_{17,\sim})$*.*

We know that the addition modulo operation definitely preserves the least significant differences in the original positions, and may preserve the other differences in the original positions or propagate them to the more significant positions, but never to the less significant positions. Thus, we can get the following property.

Property 9.3 *Given a pair of 64-bit values* (x_l, x_r) *and* $(\widehat{x}_l, \widehat{x}_r)$ *with difference* $(e_{j+5,\sim}, e_{j,\sim})$ *after Round* i $(1 \leq j \leq 27)$*, to determine whether it could produce a difference with the form* $(\xi, e_{j+5,\sim})$ *just before Round* i*, we only need to guess the most significant* $(32 - j)$ *bits of* $\widetilde{K}_i$ *and the carry bit occurred in the* $(j-1)$*th bit of the left addition modulo* 2^{32} *operation in Round* i*, where* ξ *denotes a (possible) specific 32-bit difference.*

Property 9.1 enables us to travel through the nine rounds from Rounds 16 to 20 and Rounds 48 to 51 by guessing only 64 user key bits (W_1, W_2). Properties 9.2 and 9.3 allow us to break Rounds 45 and 47 by using the early abort technique. We guess only part of the 32 bits of an unknown $\widetilde{K}_i$ when conducting an early abort; otherwise, our attack would be impossible.

We use plaintext structures in our attack. For a plaintext pair to produce the difference $(e_{22,27,31}, e_{27})$ just before Round 21, the input difference to Round 16 should have the form $(\star, e_{2,\sim})$.

9.5.2 Attack Description

As a result, the above analysis enables us to give the following attack procedure to break the 36 rounds from Rounds 16 to 51 of XTEA. The attack procedure is as follows.

1. Choose a structure S, which is defined to be a set of $2^{62.96}$ plaintexts P_l with the second rightmost bits fixed, $(l = 1, \cdots, 2^{62.96})$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the $2^{62.96}$ plaintexts encrypted with K_A and K_C, respectively; let C_l and C'_l be the ciphertexts for plaintext P_l encrypted with K_A and K_C, respectively. Choose another structure $\widehat{S}$, which contains the 2^{63} plaintexts $\widehat{P}_j$ with the second rightmost bits fixed to be the complement of the second rightmost bit value in S, $(j = 1, \cdots, 2^{63})$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{63}

plaintexts in $\widehat{S}$ encrypted with K_B and K_D; we denote by $\widehat{C}_j^*$ and $\widehat{C}_j^{\prime *}$ the ciphertexts for plaintext $\widehat{P}_j$, encrypted with K_B and K_D, respectively. Here, $K_A \oplus K_B = K_C \oplus K_D = (0,0,0,e_{32})$, and $K_A \oplus K_C = K_B \oplus K_D = (0,0,e_{32},0)$.

2. Guess a value for the 64-bit user key (W_1, W_2), compute the subkeys $(K_{16}, \cdots, K_{20})$, and perform Steps (a) and (b) below.

   (a) Partially encrypt every plaintext P_l in S with $(K_{16}, \cdots, K_{20})$ through Rounds 16 to 20 to get its corresponding value just after Round 20; we denote it by ε_l. Then, partially decrypt $\varepsilon_l \oplus (e_{22,27,31}, e_{27})$ with $(K_{16}, \cdots, K_{20})$ through Rounds 16 to 20 to get its plaintext; we denote it by $\widetilde{P}_l$. Find $\widetilde{P}_l$ in $\widehat{S}$. We denote by $\widetilde{C}_l^*$ and $\widetilde{C}_l^{\prime *}$ the corresponding ciphertexts of $\widetilde{P}_l$ encrypted under K_B and K_D, respectively. This step generates a total of $2^{62.96}$ plaintext pairs with difference $(e_{22,27,31}, e_{27})$ after Round 20 for every guess for (W_1, W_2), which can propose about $\binom{2^{62.96}}{2} \approx \frac{2^{62.96 \times 2}}{2} = 2^{124.92}$ candidate quartets.

   (b) Compute the subkeys $(K_{48}, \cdots, K_{51})$ with the guessed value for (W_1, W_2). Partially decrypt all the 2^{64} ciphertexts with $(K_{48}, \cdots, K_{51})$ through Rounds 48 to 51 to get the corresponding values just before Round 48; we denote the corresponding values for the ciphertexts C_l, $\widetilde{C}_l^*$, C_l' and $\widetilde{C}_l^{\prime *}$ by T_l, $\widetilde{T}_l^*$, T_l' and $\widetilde{T}_l^{\prime *}$, respectively. Store $(T_l, T_l', \widetilde{T}_l^*, \widetilde{T}_l^{\prime *})$ in a hash table. Finally, choose only the quartets $(T_{l_1}, \widetilde{T}_{l_1}^*, T_{l_2}', \widetilde{T}_{l_2}^{\prime *})$ such that both $T_{l_1} \oplus T_{l_2}'$ and $\widetilde{T}_{l_1}^* \oplus \widetilde{T}_{l_2}^{\prime *}$ have the form $(e_{22,\sim}, e_{17,\sim})$, where $1 \leq l_1 \leq l_2 \leq 2^{62.96}$. If one or more quartets $(T_{l_1}, \widetilde{T}_{l_1}^*, T_{l_2}', \widetilde{T}_{l_2}^{\prime *})$ pass this test, execute Step 3 with the quartets meeting this condition; otherwise, repeat Step 2 with another guess.

3. Guess a value for the most significant 16 bits $\widetilde{K}_{47,[17,32]}$ of the 32-bit value $\widetilde{K}_{47}$, and perform Steps (a) and (b) below.

   (a) For each remaining quartet $(T_{l_1}, \widetilde{T}_{l_1}^*, T_{l_2}', \widetilde{T}_{l_2}^{\prime *})$, partially decrypt T_{l_1} and T_{l_2}' with $\widetilde{K}_{47,[17,32]}$ under the two possibilities 0 and 1 of the carry bit occurred in bit (16) of the left add modulo operation to get the corresponding values for the most significant 16 bits of both the left and right halves just before Round 47; we denote them by Q_{m,l_1} and Q_{m,l_2}', respectively, where $m \in \{0,1\}$ denotes the two possibilities of the carry bit; and check whether $Q_{m,l_1} \oplus Q_{m,l_2}'$ has the form $(\rho_{11,\sim}^{16}, \rho_{6,\sim}^{16})$. If not, repeat this step with another quartet; otherwise, partially decrypt $\widetilde{T}^{*}_{l_1}$ and $\widetilde{T}'^{*}_{l_2}$ with $\widetilde{K}_{47,[17,32]} \oplus \rho^{16}_{16}$ under the two possibilities 0 and 1 of the carry bit occurred in bit (16) of the left add modulo operation to get the corresponding values for the most significant 16 bits of both the left and right halves just before Round 47; we denote them by $\widetilde{Q}^{*}_{n,l_1}$ and $\widetilde{Q}'^{*}_{n,l_2}$, respectively, where $n \in \{0, 1\}$ denotes the two possibilities of the carry bit. Finally, check whether $\widetilde{Q}^{*}_{n,l_1} \oplus \widetilde{Q}'^{*}_{n,l_2}$ has the form $(\rho^{16}_{11,\sim}, \rho^{16}_{6,\sim})$. If one or more quartets $(T_{l_1}, \widetilde{T}^{*}_{l_1}, T'_{l_2}, \widetilde{T}'^{*}_{l_2})$ pass this test, record the quartets $(Q_{m,l_1}, \widetilde{Q}^{*}_{n,l_1}, Q'_{m,l_2}, \widetilde{Q}'^{*}_{n,l_2})$, and execute Step 3(b) with the quartets $(T_{l_1}, \widetilde{T}^{*}_{l_1}, T'_{l_2}, \widetilde{T}'^{*}_{l_2})$ that meet this condition; otherwise, repeat Step 3 with another guess for $\widetilde{K}_{47,[17,32]}$.

(b) Guess a value for the least significant 16 bits $\widetilde{K}_{47,[1,16]}$ of $\widetilde{K}_{47}$. For every remaining quartet $(T_{l_1}, \widetilde{T}^{*}_{l_1}, T'_{l_2}, \widetilde{T}'^{*}_{l_2})$, partially decrypt T_{l_1} and T'_{l_2} with $\widetilde{K}_{47}(= \widetilde{K}_{47,[1,16]}||\widetilde{K}_{47,[17,32]})$ to get the corresponding values just before Round 47; we denote them by Q_{l_1} and Q'_{l_2}, respectively; and check whether $Q_{l_1} \oplus Q'_{l_2}$ has the form $(e_{27,\sim}, e_{22,\sim})$. If not, repeat this step with another quartet; otherwise, partially decrypt $\widetilde{T}^{*}_{l_1}$ and $\widetilde{T}'^{*}_{l_2}$ with $\widetilde{K}_{47} \oplus e_{32}$ to get the corresponding values just before Round 47; we denote them by $\widetilde{Q}^{*}_{l_1}$ and $\widetilde{Q}'^{*}_{l_2}$, respectively. Finally, check whether $\widetilde{Q}^{*}_{l_1} \oplus \widetilde{Q}'^{*}_{l_2}$ has the form $(e_{27,\sim}, e_{22,\sim})$. If one or more quartets $(T_{l_1}, \widetilde{T}^{*}_{l_1}, T'_{l_2}, \widetilde{T}'^{*}_{l_2})$ pass this test, execute Step 4 with the quartets $(Q_{l_1}, \widetilde{Q}^{*}_{l_1}, Q'_{l_2}, \widetilde{Q}'^{*}_{l_2})$ that meet this condition; otherwise, repeat this step with another guess for $\widetilde{K}_{47,[1,16]}$.

4. Compute the subkey $\widetilde{K}_{46}$ with the W_2 guessed in Step 2. For every remaining quartet $(Q_{l_1}, \widetilde{Q}^{*}_{l_1}, Q'_{l_2}, \widetilde{Q}'^{*}_{l_2})$, partially decrypt (Q_{l_1}, Q'_{l_2}) with $\widetilde{K}_{46}$ to get the corresponding values just before Round 46; we denote them by (R_{l_1}, R'_{l_2}), respectively; and check whether $R_{l_1} \oplus R'_{l_2}$ has the form $(e_{32}, e_{27,\sim})$. If not, repeat this step with another quartet; otherwise, partially decrypt $(\widetilde{Q}^{*}_{l_1}, \widetilde{Q}'^{*}_{l_2})$ with $\widetilde{K}_{46}$ to get the corresponding values just before Round 46; we denote them by $(\widetilde{R}^{*}_{l_1}, \widetilde{R}'^{*}_{l_2})$, respectively. Finally, check whether $\widetilde{R}^{*}_{l_1} \oplus \widetilde{R}'^{*}_{l_2}$ has the form $(e_{32}, e_{27,\sim})$. If one or more quartets $(Q_{l_1}, \widetilde{Q}^{*}_{l_1}, Q'_{l_2}, \widetilde{Q}'^{*}_{l_2})$ pass this test, execute Step 5 with the quartets $(R_{l_1}, \widetilde{R}^{*}_{l_1}, R'_{l_2}, \widetilde{R}'^{*}_{l_2})$ that meet this condition; otherwise, repeat Step 3(b) with another guess for $\widetilde{K}_{47,[1,16]}$.

5. Guess a value for the most significant 6 bits $\widetilde{K}_{45,[27,32]}$ of the 32-bit value $\widetilde{K}_{45}$, and perform Steps (a) and (b) below.

(a) For each remaining quartet $(R_{l_1}, \widetilde{R}^*_{l_1}, R'_{l_2}, \widetilde{R}'^*_{l_2})$, partially decrypt R_{l_1} and R'_{l_2} with $\widetilde{K}_{45,[27,32]}$ and $\widetilde{K}_{45,[27,32]} \oplus \rho^6_6$, respectively, under the two possibilities 0 and 1 of the carry bit occurred in bit (26) of the left add modulo operation to get the corresponding values for the most significant 6 bits of the left and right halves just before Round 45; we denote them by U_{s,l_1} and U'_{s,l_2}, respectively, and partially decrypt $\widetilde{R}^*_{l_1}$ and $\widetilde{R}'^*_{l_2}$ with $\widetilde{K}_{45,[27,32]}$ and $\widetilde{K}_{45,[27,32]} \oplus \rho^6_6$, respectively, under the two possibilities 0 and 1 of the carry bit occurred in bit (26) of the left add modulo operation to get the corresponding values for the most significant 6 bits of the left and right halves just before Round 45; we denote them by $\widetilde{U}^*_{t,l_1}$ and $\widetilde{U}'^*_{t,l_2}$, respectively, where $s, t \in \{0, 1\}$ denote the two possibilities of the carry bit. Finally, check whether $U_{s,l_1} \oplus U'_{s,l_2} = \widetilde{U}^*_{t,l_1} \oplus \widetilde{U}'^*_{t,l_2} = (0, \rho^6_6)$. If one or more quartets $(R_{l_1}, \widetilde{R}^*_{l_1}, R'_{l_2}, \widetilde{R}'^*_{l_2})$ pass this test, execute Step 5(b) with the quartets $(R_{l_1}, \widetilde{R}^*_{l_1}, R'_{l_2}, \widetilde{R}'^*_{l_2})$ that meet this condition; otherwise, repeat Step 5 with another guess for $\widetilde{K}_{45,[27,32]}$.

(b) Guess a value for the least significant 26 bits $\widetilde{K}_{45,[1,26]}$ of $\widetilde{K}_{45}$. For every remaining $(R_{l_1}, \widetilde{R}^*_{l_1}, R'_{l_2}, \widetilde{R}'^*_{l_2})$, partially decrypt R_{l_1} and R'_{l_2} with $\widetilde{K}_{45}(= \widetilde{K}_{45,[1,26]} || \widetilde{K}_{45,[27,32]})$ and $\widetilde{K}_{45} \oplus e_{32}$, respectively, to get the corresponding values just before Round 45; we denote them by U_{l_1} and U'_{l_2}, respectively; and check whether $U_{l_1} \oplus U'_{l_2} = (0, e_{32})$. If not, repeat this step with another quartet; otherwise, partially decrypt $\widetilde{R}^*_{l_1}$ and $\widetilde{R}'^*_{l_2}$ with $\widetilde{K}_{45}$ and $\widetilde{K}_{45} \oplus e_{32}$, respectively, to get the corresponding values just before Round 45; we denote them by $\widetilde{U}^*_{l_1}$ and $\widetilde{U}'^*_{l_2}$, respectively. Finally, check whether $\widetilde{U}^*_{l_1} \oplus \widetilde{U}'^*_{l_2} = (0, e_{32})$. If one or more quartets $(R_{l_1}, \widetilde{R}^*_{l_1}, R'_{l_2}, \widetilde{R}'^*_{l_2})$ pass this test, execute Step 6 with the quartets $(U_{l_1}, \widetilde{U}^*_{l_1}, U'_{l_2}, \widetilde{U}'^*_{l_2})$ that meet this condition; otherwise, repeat this step with another guess for $\widetilde{K}_{45,[1,26]}$.

6. Compute the subkey $\widetilde{K}_{21}$ with the W_3 indicated by $\widetilde{K}_{45}$. For every plaintext quartet $(P_{l_1}, \widetilde{P}^*_{l_1}, P'_{l_2}, \widetilde{P}'^*_{l_2})$ corresponding to a remaining $(U_{l_1}, \widetilde{U}^*_{l_1}, U'_{l_2}, \widetilde{U}'^*_{l_2})$, partially encrypt ε_{l_1} and $\varepsilon_{l_1} \oplus (e_{22,27,31}, e_{27})$ with $\widetilde{K}_{21}$ to get the corresponding values just after Round 21; we denote them by V_{l_1} and $\widetilde{V}^*_{l_1}$, respectively; and check whether $V_{l_1} \oplus \widetilde{V}^*_{l_1} = (e_{27}, e_{32})$. If not, repeat this step with another quartet; otherwise, partially encrypt ε_{l_2} and $\varepsilon_{l_2} \oplus (e_{22,27,31}, e_{27})$ with $\widetilde{K}_{21} \oplus e_{32}$ to get the corresponding values just after Round 21; we denote them by V_{l_2} and $\widetilde{V}^*_{l_2}$, respectively. Finally, check whether $V_{l_2} \oplus \widetilde{V}^*_{l_2} = (e_{27}, e_{32})$. If one or more

quartets $(P_{l_1}, \widetilde{P}^*_{l_1}, P'_{l_2}, \widetilde{P}'^*_{l_2})$ pass this test, then record $(W_1, W_2, \widetilde{K}_{47}, \widetilde{K}_{45})$, and execute Step 7; otherwise, repeat Step 5(b) with another guess for $\widetilde{K}_{45,[1,26]}$.

7. For a recorded value for $(W_1, W_2, \widetilde{K}_{47}, \widetilde{K}_{45})$, do a trial encryption with three plaintext/ciphertext pairs to determine the correct user key of the 36-round XTEA, (If all the possible guesses during any of Steps 3 to 5 are tested, repeat its previous steps with other guess).

9.5.3 Complexity Analysis

The attack requires $2 \times (2^{62.96} + 2^{63}) \approx 2^{64.98}$ related-key chosen plaintexts. The required memory for this attack is dominated by the ciphertexts, which is approximately $2^{64.98} \times 8 = 2^{67.98}$ memory bytes.

Step 2(a) has a time complexity of about $2 \times 2^{62.96} \times 2^{64} \times \frac{5}{36} \approx 2^{125.12}$ 36-round XTEA encryptions. The time complexity of Step 2(b) is dominated by the partial decryptions, which is about $2^{64} \times 2^{64} \times \frac{4}{36} \approx 2^{124.83}$ 36-round XTEA computations. Besides, Step 2(b) requires about $2^{64} \times 2^{62.96} \approx 2^{126.96}$ memory accesses, which is negligible compared with the $2^{124.83}$ computations (actually it can be done more efficiently using computers of today). In Step 2(b), the probability that a quartet meets the filtering condition is $(\frac{1}{2^{22}} \times \frac{1}{2^{17}})^2 = 2^{-78}$, so it follows that the expected number of the quartets passing the test for each guess is $2^{124.92} \times 2^{-78} = 2^{46.92}$. The probability that one or more quartets pass the test for a wrong guess is about $\sum_{i=1}^{2^{124.92}}[\binom{2^{124.92}}{i} \cdot (2^{-78})^i \cdot (1 - 2^{-78})^{2^{124.92}-i}] \approx 1$, thus, almost all the 2^{64} possible values of (W_1, W_2) pass Step 2(b).

In Step 3(a), the probability that a remaining quartet meets either of the filtering conditions is $\frac{1}{2^{10}} + \frac{1}{2^{10}} = 2^{-9}$, thus the time complexity of Step 3(a) is about $2 \times 2^{64} \times 2^{16} \times 2^{46.92} \times 2 \times \frac{1}{2} \times \frac{1}{36} + 2 \times 2^{64} \times 2^{16} \times 2^{37.92} \times 2 \times \frac{1}{2} \times \frac{1}{36} \approx 2^{122.75}$, where $\frac{1}{2}$ means the average fraction of the key bits that are tested. In this step, the probability that a remaining quartet meets both the filtering conditions is $(\frac{1}{2^{10}} + \frac{1}{2^{10}})^2 = 2^{-18}$, so the expected number of the quartets passing the test for each guess is $2^{46.92} \times 2^{-18} = 2^{28.92}$, and the probability that one or more quartets pass the test for a wrong guess is about 1. Thus, it is expected that almost all the 2^{80} possible $(W_1, W_2, \widetilde{K}_{47,17-32})$ pass this step. In Step 3(b), the probability that a remaining quartet meets either of the filtering conditions is 2^{-1}, because both the pairs in a remaining quartet should produce the required carry bits occurred in bit 15 of the left add modulo operation; thus the time complexity of Step 3(b) is about $2 \times 2^{80} \times 2^{16} \times 2^{28.92} \times 2 \times \frac{1}{2} \times \frac{1}{36} + 2 \times 2^{80} \times 2^{16} \times 2^{27.92} \times 2 \times \frac{1}{2} \times \frac{1}{36} \approx 2^{121.34}$; the probability that a remaining quartet meets both the filtering conditions is $2^{-1\times 2} = 2^{-2}$, so the expected number of the quartets passing the test for each guess is $2^{28.92} \times 2^{-2} = 2^{26.92}$, and almost all the 2^{96} possible values of $(W_1, W_2, \widetilde{K}_{47})$ pass Step 3(b).

In Step 4, the probability that a remaining quartet meets either of the filtering conditions is 2^{-10}, thus the time complexity of Step 4 is about $2 \times 2^{96} \times 2^{26.92} \times \frac{1}{2} \times \frac{1}{36} + 2 \times 2^{96} \times 2^{16.92} \times \frac{1}{2} \times \frac{1}{36} \approx 2^{117.75}$. In this step, the probability that a remaining quartet meets both the filtering conditions is $2^{-10\times 2} = 2^{-20}$, so the expected number of the quartets passing the test for each guess is $2^{26.92} \times 2^{-20} = 2^{6.92}$, and the probability that one or more quartets pass the test for a wrong guess is about 1. Thus, it is expected that almost all the 2^{96} possible values of $(W_1, W_2, \widetilde{K}_{47})$ pass this step.

In Step 5(a), the time complexity is about $4 \times 2^{96} \times 2^{6} \times 2^{6.92} \times 2 \times \frac{1}{2} \times \frac{1}{36} \approx 2^{105.75}$, and the probability that a remaining quartet meets the filtering condition is $(\frac{1}{2^5} + \frac{1}{2^5})^2 = 2^{-8}$, so the expected number of the quartets passing the test for each guess is $2^{6.92} \times 2^{-8} = 2^{-1.08}$. The probability that one or more quartets pass the test for a wrong guess is about $\sum_{i=1}^{2^{6.92}} [\binom{2^{6.92}}{i} \cdot (2^{-8})^i \cdot (1 - 2^{-8})^{2^{6.92}-i}] \approx 2^{-1.08}$. Hence, it is expected that about $2^{96} \times 2^{6} \times 2^{-1.08} = 2^{100.92}$ possible values of $(W_1, W_2, \widetilde{K}_{47}, \widetilde{K}_{45,[27,32]})$ pass Step 5(a). In Step 5(b), the probability that a remaining quartet meets either of the filtering conditions is 2^{-1}; as a result, it is expected that $2^{100.92} \times 2^{26} \times 2^{-1} = 2^{125.92}$ possible values of $(W_1, W_2, \widetilde{K}_{47}, \widetilde{K}_{45})$ pass the first filtering condition in Step 5(b). Therefore, the time complexity of Step 5(b) is about $2 \times 2^{100.92} \times 2^{26} \times \frac{1}{2} \times \frac{1}{36} + 2 \times 2^{125.92} \times \frac{1}{2} \times \frac{1}{36} \approx 2^{122.34}$. In this step, it is expected that about $2^{126.92} \times 2^{-2} = 2^{124.92}$ possible values of $(W_1, W_2, \widetilde{K}_{47}, \widetilde{K}_{45})$ pass Step 5(b).

In Step 6, as the probability that a remaining quartet meets either of the filtering conditions is $2^{-4.16}$, it follows that about $2^{124.92} \times 2^{-4.16} = 2^{120.76}$ possible values of $(W_1, W_2, \widetilde{K}_{47}, \widetilde{K}_{45})$ are expected to pass the first filtering condition in this step. The time complexity of this step is about $2 \times 2^{124.92} \times \frac{1}{2} \times \frac{1}{36} + 2 \times 2^{120.76} \times \frac{1}{2} \times \frac{1}{36} \approx 2^{119.83}$. In this step, the probability that a remaining quartet meets both the filtering conditions is $2^{-4.16\times 2} = 2^{-8.32}$, so it follows that about $2^{124.92} \times 2^{-8.32} = 2^{116.6}$ possible values of $(W_1, W_2, \widetilde{K}_{47}, \widetilde{K}_{45})$ are expected to pass this step, which result in about $2^{116.6}$ trials in Step 7.

Therefore, this attack has a total of approximately $2^{126.3}$ 36-round XTEA computations.

The probability that a wrong key is suggested in Step 7 is approximately 2^{-192}, so the expected number of suggested wrong 128-bit keys is about $2^{-192} \times 2^{116.6} = 2^{-75.4}$, which is extremely low. In Step 6, the expected number of quartets for the correct key guess is $2^{124.92} \times 2^{-124.92} = 1$, and the probability that one or more quartets pass the test for the correct key guess is approximately $\sum_{i=1}^{2^{124.92}} [\binom{2^{124.92}}{i} \cdot (2^{-124.92})^i \cdot (1 - 2^{-124.92})^{2^{124.92}-i}] \approx 0.63$. Therefore, with a success probability of 63%, the related-key rectangle attack can break the 36-round XTEA, marginally faster than exhaustive key search.

9.6 Summary

In this chapter we have presented a related-key rectangle attack on 36 rounds of XTEA. This is better than any previously published cryptanalytic results on XTEA in terms of the number of attacked rounds. Table 9.3 summarises the published cryptanalytic results on XTEA, where CP and RK-CP refer to the required numbers of chosen plaintexts and related-key chosen plaintexts, respectively; and Encryptions refers to the required number of encryption operations of the relevant reduced-round version of XTEA.

Table 9.3: Cryptanalytic results on XTEA

Attack Type	Rounds	Data	Time	Source
Impossible differential	14	$2^{62.5}$CP	2^{85}Encryptions	[86]
Differential	15	2^{59}CP	2^{120}Encryptions	[38]
Truncated differential	23	$2^{20.55}$CP	$2^{120.65}$Encryptions	[38]
Related-key truncated differential	25	116RK-CP	$2^{110.05}$Encryptions	[61]
	27	$2^{20.5}$RK-CP	$2^{115.15}$Encryptions	[61]
Related-key rectangle	34†	2^{62}RK-CP	$2^{31.94}$Encryptions	[70]
	36	$2^{64.98}$RK-CP	$2^{126.3}$Encryptions	Section 9.5

†: Under weak key assumptions

Chapter 10

Cryptanalysis of Reduced HIGHT

HIGHT is a 64-bit block cipher with a 128-bit user key. In this chapter, we present an impossible differential attack on 25-round HIGHT, a related-key rectangle attack on 26-round HIGHT, and a related-key impossible differential attack on 28-round HIGHT. The 25-round HIGHT attack requires 2^{60} *chosen plaintexts and has a time complexity of* $2^{126.78}$ *encryptions; the 26-round HIGHT attack requires* $2^{49.7}$ *related-key chosen plaintexts and has a time complexity of* $2^{120.41}$ *encryptions; the 28-round HIGHT attack requires* 2^{60} *related-key chosen plaintexts and has a time complexity of* $2^{125.54}$ *encryptions. These attacks are better than any previously published cryptanalytic results on HIGHT in terms of the number of attacked rounds.*

Contents

10.1 Introduction

Recently, cryptographic techniques suitable for use in embedded and ubiquitous computing systems has received extensive attention. In 2006, Hong, Sung, Hong, Lim, Lee, Koo, Lee, Chang, Lee, Jeong, Kim, Kim and Chee [37] proposed a 64-bit block cipher known as HIGHT, meaning "high security and light weight". HIGHT has a Feistel structure with four branches, a 128-bit user key, and a total of 32 rounds. It is especially efficient in hardware implementations, and is most suitable for various real-life resource-constrained application environments, such as RFID (Radio-Frequency IDentification) [47].

In this chapter we describe certain 16-round impossible differentials for HIGHT, and use them to mount an impossible differential attack on 25-round HIGHT requiring 2^{60} chosen plaintexts and has a time complexity of $2^{126.78}$ encryptions. We next describe an 18-round related-key rectangle distinguisher with probability $2^{-92.4}$ for HIGHT, and then use it to construct a related-key rectangle attack on 26-round HIGHT which requires $2^{49.7}$ related-key chosen plaintexts and has a time complexity of $2^{120.41}$ encryptions. Finally, we describe certain 19-round related-key impossible differentials for HIGHT, and use them to mount a related-key impossible differential attack on 28-round HIGHT which requires 2^{60} related-key chosen plaintexts and has a time complexity of $2^{125.54}$ encryptions. The attacks use the early abort technique described in Chapter 4.

The remainder of this chapter is organised as follows. In Section 10.2 we describe HIGHT. In Section 10.3 we briefly review previous cryptanalytic results on HIGHT. In Section 10.4 we introduce two properties of HIGHT. In Sections 10.5, 10.6 and 10.7 we present our cryptanalytic results on HIGHT. Section 10.8 summarises the results of this chapter.

10.2 The HIGHT Block Cipher

In this section we briefly describe the HIGHT block cipher [37]. Note that in order to maintain consistency of presentation throughout the thesis, the description below is different from (but equivalent to) that given in [37]; in particular, we use a different numbering of the bits of a value.

10.2.1 Notation

In this chapter, a 64-bit value is represented as a sequence of eight bytes, numbered from 1 to 8 from left to right; and the bits of a byte are numbered from 1 to 8 from left to right, where the least significant bit is referred as the 1st bit, and the most significant bit is referred as the 8th bit. We use the following notation.

- $\boxplus$: addition modulo 2^8
- e_j : an 8-bit value with zeros everywhere except for bit position j $(1 \leq j \leq 8)$
- $e_{i_1,\cdots,i_j}$: the 8-bit value equal to $e_{i_1} \oplus \cdots \oplus e_{i_j}$ $(1 \leq i_1, \cdots, i_j \leq 8)$
- $e_{j,\sim}$: an 8-bit value that has zeros in bit positions 1 to $j-1$, a one in bit position j and indeterminate values in bit positions $(j+1)$ to 8, $1 \leq j \leq 8$
- $\overline{e}_{j,\sim}$: an 8-bit value that has zeros in bit positions 1 to j and indeterminate values in bit positions $(j+1)$ to 8, $1 \leq j \leq 8$
- $\star$: an arbitrary 8-bit value, where two values represented by the $\star$ symbol may be different

10.2.2 Functions

The HIGHT round function uses the following two elementary functions:

- $\mathbf{F}_0 : \{0,1\}^8 \rightarrow \{0,1\}^8$. If X is a 8-bit block, then $\mathbf{F}_0(X) = (X \ggg 1) \oplus (X \ggg 2) \oplus (X \ggg 7)$.

- $\mathbf{F}_1 : \{0,1\}^8 \to \{0,1\}^8$. If X is a 8-bit block, then $\mathbf{F}_1(X) = (X \ggg 3) \oplus (X \ggg 4) \oplus (X \ggg 6)$.

10.2.3 Generation of Subkeys

HIGHT uses a total of eight 8-bit whitening subkeys KW_j $(1 \leq j \leq 8)$, and 128 8-bit round subkeys KS_i, $(1 \leq i \leq 128)$, all derived from a 128-bit user key K. Let K be represented as a sequence of as sixteen bytes $(W_1, W_2, \cdots, W_{16})$.

The whitening subkeys KW_j are defined as follows.

$$KW_j = W_{j+12} \text{ for } j = 1, 2, 3, 4;$$
$$KW_j = W_{j-4} \text{ for } j = 5, 6, 7, 8.$$

The round subkeys KS_i are as follows, where $\lambda_{16\cdot l+j-16}$ and $\lambda_{16\cdot l+j-8}$ are public constants, $(1 \leq l, j \leq 8)$.

$$KS_{16\cdot l+j-16} = W_{j-l \bmod 8+1} \boxplus \lambda_{16\cdot l+j-16};$$
$$KS_{16\cdot l+j-8} = W_{(j-l \bmod 8)+9} \boxplus \lambda_{16\cdot l+j-8}.$$

Table 10.1 lists the user key byte used to compute the round subkey KS_i for every i, $(1 \leq i \leq 128)$.

We write $W_{i,l}$ for the lth bit of W_i, $W_{i,[l_1,l_2]}$ for bits $(l_1, \cdots, l_2)$ of W_i, $KS_{i,l}$ for the lth bit of KS_i, and $KS_{i,[l_1,l_2]}$ for bits $(l_1, \cdots, l_2)$ of KS_i, where $1 \leq l \leq 8, 1 \leq l_1 \leq l_2 \leq 8$.

10.2.4 Encryption Procedure

HIGHT takes as input a 64-bit plaintext block P, and has a total of 32 rounds. Its encryption procedure is as follows, where $X_{0,1}, X_{0,2}, \cdots, X_{0,8}, X_{i,1}, \cdots, X_{i,8}$ are 8-bit variables.

1. Represent P as eight bytes $P = (P_1, P_2, \cdots, P_8)$.

Table 10.1: The key byte used to generate the round subkey KS_i

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
KS_i	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_9	W_{10}	W_{11}	W_{12}	W_{13}	W_{14}	W_{15}	W_{16}
i	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
KS_i	W_8	W_1	W_2	W_3	W_4	W_5	W_6	W_7	W_{16}	W_9	W_{10}	W_{11}	W_{12}	W_{13}	W_{14}	W_{15}
i	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
KS_i	W_7	W_8	W_1	W_2	W_3	W_4	W_5	W_6	W_{15}	W_{16}	W_9	W_{10}	W_{11}	W_{12}	W_{13}	W_{14}
i	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
KS_i	W_6	W_7	W_8	W_1	W_2	W_3	W_4	W_5	W_{14}	W_{15}	W_{16}	W_9	W_{10}	W_{11}	W_{12}	W_{13}
i	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
KS_i	W_5	W_6	W_7	W_8	W_1	W_2	W_3	W_4	W_{13}	W_{14}	W_{15}	W_{16}	W_9	W_{10}	W_{11}	W_{12}
i	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
KS_i	W_4	W_5	W_6	W_7	W_8	W_1	W_2	W_3	W_{12}	W_{13}	W_{14}	W_{15}	W_{16}	W_9	W_{10}	W_{11}
i	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
KS_i	W_3	W_4	W_5	W_6	W_7	W_8	W_1	W_2	W_{11}	W_{12}	W_{13}	W_{14}	W_{15}	W_{16}	W_9	W_{10}
i	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
KS_i	W_2	W_3	W_4	W_5	W_6	W_7	W_8	W_1	W_{10}	W_{11}	W_{12}	W_{13}	W_{14}	W_{15}	W_{16}	W_9

2. $(X_{0,1}, X_{0,2}, X_{0,3}, X_{0,4}, X_{0,5}, X_{0,6}, X_{0,7}, X_{0,8}) = (P_1 \boxplus KW_1, P_2, P_3 \oplus KW_2, P_4, P_5 \boxplus KW_3, P_6, P_7 \oplus KW_4, P_8)$.

3. For $i = 1$ to 32:

$$\begin{aligned}
X_{i,1} &= X_{i-1,8} \oplus (\mathbf{F}_0(X_{i-1,7}) \boxplus KS_{4i}),\\
X_{i,2} &= X_{i-1,1},\\
X_{i,3} &= X_{i-1,2} \boxplus (\mathbf{F}_1(X_{i-1,1}) \oplus KS_{4i-1}),\\
X_{i,4} &= X_{i-1,3},\\
X_{i,5} &= X_{i-1,4} \oplus (\mathbf{F}_0(X_{i-1,3}) \boxplus KS_{4i-2}),\\
X_{i,6} &= X_{i-1,5},\\
X_{i,7} &= X_{i-1,6} \boxplus (\mathbf{F}_1(X_{i-1,5}) \oplus KS_{4i-3}),\\
X_{i,8} &= X_{i-1,7}.
\end{aligned}$$

4. Ciphertext $= (X_{32,2} \boxplus KW_5, X_{32,3}, X_{32,4} \oplus KW_6, X_{32,5}, X_{32,6} \boxplus KW_7, X_{32,7}, X_{32,8} \oplus KW_8, X_{32,1})$.

The ith iteration of Step 3 in the above description is referred to below as Round i, $(1 \leq i \leq 32)$, the transformation in Step 2 is referred to below as the initial transformation, and the transformation in Step 4 is referred to below as the final transformation. Figure 10.1 depicts an encryption round.

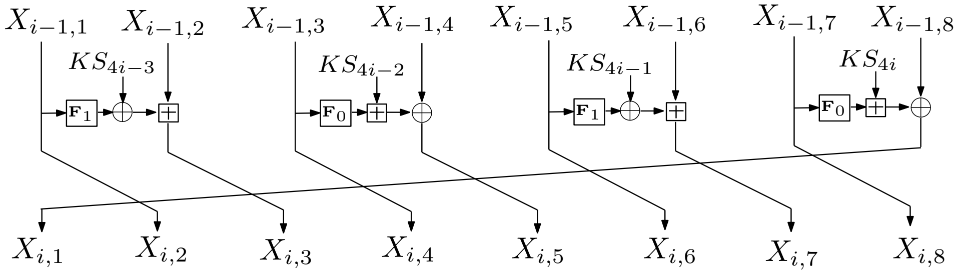


Figure 10.1: The ith encryption round of HIGHT

10.3 Previous Cryptanalytic Results

The HIGHT proposers Hong et al. [37] describe a differential attack, a linear attack and a boomerang attack on 13-round HIGHT, a truncated differential attack and a saturation attack on 16-round HIGHT, an impossible differential attack on 18-round HIGHT, and a related-key boomerang attack on 19-round HIGHT. These are the only previously published cryptanalytic results on HIGHT.

10.4 Properties of HIGHT

We first give the following general property of the $\boxplus$ and $\oplus$ operations.

Property 10.1 *The $\boxplus$ operation definitely preserves the least significant differences in the original positions, and may preserve the other differences in the original positions or propagate them to the more significant positions, but never to the less significant positions, while the $\oplus$ operation always preserves all the differences in their original positions.*

HIGHT has a Feistel-like round structure with four branches, which can be efficiently implemented. However, we observe that this round structure is much less effective in diffusing bit/byte changes than other commonly used Feistel structures. This property of limited diffusion can be formalised in the following way.

Property 10.2 *A byte value (or difference) input to Round i will affect at most two bytes of the output of Round i; two byte value (or difference) input to Round*

i will affect at most four bytes of the output of Round i; and three byte value (or difference) input to Round i will affect at most six bytes of the output of Round i, $(1 \le i \le 31)$.

Property 10.2 implies that, in order to learn a byte value (or difference) input to a round, we need not guess all the twelve 8-bit subkeys in the following three rounds. Also we can determine whether a candidate pair is useful in a byte by byte way, and even bit by bit, because of the round structure and the operations involved. This observation is another example of the application of the early abort technique.

10.5 Impossible Differential Attack on 25-Round HIGHT

In this section, we describe certain 16-round impossible differentials of HIGHT, and then use them to conduct an impossible differential attack on 25-round HIGHT.

10.5.1 16-Round Impossible Differentials

We describe certain 16-round impossible differentials: $(0,0,0,0,0,0,0,e_{i,\sim}) \nrightarrow (e_8,0,0,0,0,0,0,e_{1,4,6,7,8})$, where $2 \le i \le 8$. Note that the 16-round differentials $(0,0,0,0,0,0,e_{1,4,6,7,8},e_8) \rightarrow (0,0,0,0,0,0,e_{i,\sim},0)$ are also impossible. These 16-round impossible differentials arise because of Property 10.1.

The 16-round impossible differentials are built in a miss-in-the-middle manner [5]: a 8-round differential $(0,0,0,0,0,0,0,e_{i,\sim}) \rightarrow (\star,\star,\star,\star,\star,\star,\star,e_{i,\sim})$ with probability 1 is concatenated with another 8-round differential $(\star,\star,\star,\star,\star,\star,0,e_{1,\sim}) \leftarrow (e_8,0,0,0,0,0,0,e_{1,4,6,7,8})$ with probability 1, but the rightmost bytes of the intermediate differences of these two differentials contradict one another. Table 10.2 shows more details of the two 8-round differentials.

The input difference $(0,0,0,0,0,0,0,e_{i,\sim})$ of the first 8-round differential propagates to a difference $(e_{i,\sim},0,0,0,0,0,0,0)$ after one round of HIGHT, which then propagates to a difference $(0,e_{i,\sim},\star,0,0,0,0,0)$ after another round. As a result, the difference $(0,e_{i,\sim},\star,0,0,0,0,0)$ finally propagates to a difference $(\star,\star,\star,\star,\star,\star,\star,e_{i,\sim})$

Table 10.2: The two 8-round differentials in the 16-round impossible differential

Round(i)	$\Delta X_{i-1,1}$	$\Delta X_{i-1,2}$	$\Delta X_{i-1,3}$	$\Delta X_{i-1,4}$	$\Delta X_{i-1,5}$	$\Delta X_{i-1,6}$	$\Delta X_{i-1,7}$	$\Delta X_{i-1,8}$
1	0	0	0	0	0	0	0	$e_{i,\sim}$
2	$e_{i,\sim}$	0	0	0	0	0	0	0
3	0	$e_{i,\sim}$	$\star$	0	0	0	0	0
4	0	0	$e_{i,\sim}$	$\star$	$\star$	0	0	0
5	0	0	0	$e_{i,\sim}$	$\star$	$\star$	$\star$	0
6	$\star$	0	0	0	$e_{i,\sim}$	$\star$	$\star$	$\star$
7	$\star$	$\star$	$\star$	0	0	$e_{i,\sim}$	$\star$	$\star$
8	$\star$	$\star$	$\star$	$\star$	$\star$	0	$e_{i,\sim}$	$\star$
output	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$e_{i,\sim}$
9	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	0	$e_{1,\sim}$
10	$e_{1,\sim}$	$\star$	$\star$	$\star$	$\star$	$\star$	0	0
11	0	$e_{1,\sim}$	$\star$	$\star$	$\star$	$\star$	0	0
12	0	0	$e_{1,\sim}$	$\star$	$\star$	$\star$	0	0
13	0	0	0	$e_{1,\sim}$	$\star$	$\star$	0	0
14	0	0	0	0	$e_{1,\sim}$	$\star$	0	0
15	0	0	0	0	0	$e_{1,\sim}$	0	0
16	0	0	0	0	0	0	$e_{1,4,6,7,8}$	0
output	e_8	0	0	0	0	0	0	$e_{1,4,6,7,8}$

after a further six rounds.

On the other hand, when we roll back the difference $(e_8, 0, 0, 0, 0, 0, 0, e_{1,4,6,7,8})$ through one round of HIGHT in the reverse direction, we definitely get the difference $(0, 0, 0, 0, 0, 0, e_{1,4,6,7,8}, 0)$, as the difference $e_{1,4,6,7,8}$ becomes $(e_{1,4,6,7,8} \ggg 1) \oplus (e_{1,4,6,7,8} \ggg 2) \oplus (e_{1,4,6,7,8} \ggg 7) = e_{1,2,5,7,8} \oplus e_{1,2,3,6,8} \oplus e_{3,5,6,7,8} = e_8$ after the $\mathbf{F}_0$ function. The difference $(0, 0, 0, 0, 0, 0, e_{1,4,6,7,8}, 0)$ propagates to a difference $(\star, \star, \star, \star, \star, \star, 0, e_{1,\sim})$ when we roll it back through seven more rounds.

We now have a contradiction if $i \neq 1$, as the rightmost byte difference of one of the two intermediate differences is $e_{i,\sim}$ while the leftmost byte difference of the other is $e_{1,\sim}$.

These impossible differentials apply to any set of sixteen consecutive rounds of HIGHT.

10.5.2 Attacking Rounds 6 to 30

We can use the 16-round impossible differentials to break 25-round HIGHT. We attack Rounds 6 to 30 of HIGHT with only the final transformation. We use the 16-round impossible differentials described in the previous section applied to Rounds

11 to 26. The attack procedure is as follows.

10.5.2.1 Attack Description

1. Choose 2^{13} structures S_i, $(i = 1, 2, \cdots, 2^{13})$, where a structure S_i is defined to be a set of 2^{47} plaintexts $P_{i,j}$ with the first two bytes and bit (1) of the third byte fixed, and the other 47 bit positions taking all the possible values, $(j = 1, 2, \cdots, 2^{47})$. In a chosen-plaintext attack scenario, obtain all the 2^{60} ciphertexts for the 2^{47} plaintexts in each of the 2^{13} structures; let $C_{i,j}$ be the ciphertext for plaintext $P_{i,j}$. Choose only the ciphertext pairs (C_{i,j_1}, C_{i,j_2}) such that $C_{i,j_1} \oplus C_{i,j_2} = (0, 0, e_{1,\sim}, \star, \star, \star, \star, \star)$, where $1 \leq j_1 \neq j_2 \leq 2^{47}$.

2. Guess a value for the two key bytes (W_1, W_4), then compute the subkeys (KW_8, KS_{120}), and perform Steps (a)–(h) below.

   (a) Partially decrypt every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) with (KW_8, KS_{120}) to get the corresponding values for bytes (7,8) just before Round 30, and check whether they have a difference $(\star, 0)$. Keep only the pairs that meet this condition.

   (b) Guess a value for the two key bytes (W_3, W_8), then compute the subkeys (KW_7, KS_{119}), and compute the subkey KS_{115} with the W_4 guessed above. Partially decrypt every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) with $(KW_7, KS_{115}, KS_{119})$ to get the corresponding values for bytes (5,6) just before Round 29.[1] Check whether they have a difference $(\star, 0)$. Keep only the pairs that meet this condition.

   (c) Guess a value for the key byte W_2, compute the subkey KW_6, and perform the following two sub-steps.

      i. Guess a value for the least significant bit $W_{7,1}$ of the key byte W_7, and compute the least significant bit $KS_{118,1}$ of the subkey KS_{118}. Partially decrypt every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) with $(KW_6, KS_{118,1})$ to get the corresponding values for bit (1) of byte (4) just before Round 30, and check whether they have a non-zero difference. Keep only the pairs that meet this condition.

[1] The other required corresponding values have been obtained in the previous steps. The same statement applies to certain subsequent steps, as well as the attacks in the next two sections, although we do not make any further explicit statements.

ii. Guess a value for the most significant seven bits $W_{7,[2,8]}$ of W_7, and compute the subkey KS_{118} (together with the $W_{7,1}$ guessed above). Partially decrypt every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) with (KW_6, KS_{118}) to get the corresponding values for bytes (3,4) just before Round 30.

(d) Compute the subkey KS_{114} with the W_3 guessed above. For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for bytes (4,5) just before Round 30 with KS_{114} to get the corresponding values for bytes (3,4) just before Round 29, and check whether they have a difference $(e_{1,\sim}, e_{3,\sim})$. Keep only the pairs that meet this condition.

(e) For $l = 1$ to 8:

- Guess a value for the lth bit $W_{16,l}$ of the key byte W_{16}, and compute the l-bit subkey $KS_{110,[1,l]}$ of the subkey KS_{110}.
- For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for bytes (4,5) just before Round 29 with $KS_{110,[1,l]}$ to get the corresponding values for bits $(1, 2, \cdots, l)$ of byte (4) just before Round 28, and check whether they have a zero difference. Keep only the pairs that meet this condition.

(f) Guess a value for the key byte W_6, compute the subkey KS_{117}, and compute the subkeys (KW_5, KS_{113}) with the (W_1, W_2) guessed above. Partially decrypt every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) with $(KW_5, KS_{113}, KS_{117})$ to get the corresponding values for bytes (1,2) just before Round 29, and check whether they have the difference $(0, e_{1,4,6,7,8})$. Keep only the pairs that meet this condition.

(g) Guess a value for the least significant bit $W_{15,1}$ of the key byte W_{15}. For $l = 2$ to 8, perform the following two sub-steps.

- Guess a value for the lth bit $W_{15,l}$ of W_{15}, and compute the l-bit subkey $KS_{109,[1,l]}$ of the subkey KS_{109}.
- For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for bytes (2,3) just before Round 29 with $KS_{109,[1,l]}$ to get the corresponding values for bits $(1, 2, \cdots, l)$ of byte (2) just before Round 28. If $l \neq 8$, check whether they have a zero

difference; if $l = 8$, check whether they have difference e_8. Keep only the pairs that meet this condition.

(h) Guess a value for the least significant 3 bits $W_{11,[1,3]}$ of the key byte W_{11}. For $l = 4$ to 8, perform the following two sub-steps.

- Guess a value for the lth bit $W_{11,l}$ of W_{11}, and compute the l-bit subkey $KS_{105,[1,l]}$ of the subkey KS_{105}.
- For every remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}), partially decrypt the corresponding values for bytes (2,3) just before Round 28 with $KS_{105,[1,l]}$ to get the corresponding values for bits $(1, 2, \cdots, l)$ of byte (2) just before Round 27, and check whether they have a zero difference. Keep only the pairs that meet this condition.

3. Compute the subkey KS_{24} with the W_7 guessed in Step 2, and perform Steps (a)–(e) below.

(a) Partially encrypt every plaintext pair (P_{i,j_1}, P_{i,j_2}) corresponding to a remaining ciphertext pair (C_{i,j_1}, C_{i,j_2}) with KS_{24} to get the corresponding values for bytes (1,8) just after Round 6, and check whether they have a difference $(0, \star)$. Keep only the plaintext pairs that meet this condition.

(b) Compute the subkeys (KS_{23}, KS_{28}) with the (W_6, W_{11}) guessed in Step 2. Partially encrypt every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) with (KS_{23}, KS_{28}) to get the corresponding values for bytes (1,8) just after Round 7, and check whether they have a difference $(0, \star)$. Keep only the plaintext pairs that meet this condition.

(c) Guess a value for the two key bytes (W_5, W_{10}), compute the subkeys (KS_{22}, KS_{27}), and compute the subkey KS_{32} with the W_{15} guessed in Step 2. Partially encrypt every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) with $(KS_{22}, KS_{27}, KS_{32})$ to get the corresponding values for bytes (1,8) just after Round 8, and check whether they have a difference $(0, \star)$. Keep only the plaintext pairs that meet this condition.

(d) Guess a value for the two key bytes (W_9, W_{14}), compute the subkeys (KS_{26}, KS_{31}), and compute the subkeys (KS_{21}, KS_{36}) with the (W_2, W_4) guessed in Step 2. Partially encrypt every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) with $(KS_{21}, KS_{26}, KS_{31}, KS_{36})$ to get the corresponding values for bytes (1,8) just after Round 9, and check whether they have a difference $(0, \star)$. Keep only the plaintext pairs that meet this condition.

(e) Guess a value for the key byte W_{13}, compute the subkey KS_{30}, and compute the subkeys $(KS_{25}, KS_{35}, KS_{40})$ with the (W_1, W_6, W_{16}) guessed in Step 2. Partially encrypt every remaining plaintext pair (P_{i,j_1}, P_{i,j_2}) with $(KS_{25}, KS_{30}, KS_{35}, KS_{40})$ to get the corresponding values for bytes (1,8) just after Round 10, and check whether they have a difference $(0, \overline{e}_{1,\sim})$. If none of the remaining plaintext pairs meets this condition, record the guessed value for $(W_1, \cdots, W_{11}, W_{13}, \cdots, W_{16})$, and execute Step 4; otherwise, discard this guess, and try another.

4. For a recorded value for $(W_1, \cdots, W_{11}, W_{13}, \cdots, W_{16})$, exhaustively search for the remaining 8 key bits using three known pairs of plaintexts and ciphertexts. If a 128-bit key is suggested, output it as the user key of the 25-round HIGHT; otherwise, go to Step 2.

10.5.2.2 Complexity Analysis

The attack requires 2^{60} chosen plaintexts, which take a time complexity of 2^{60} 25-round HIGHT encryptions.

In Step 1, a structure S_i yields $\binom{47}{2} \approx \frac{2^{47 \times 2}}{2} = 2^{93}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}) with difference $(0, 0, \overline{e}_{1,\sim}, \star, \star, \star, \star, \star)$, $(i = 1, 2, \cdots, 2^{13}, 1 \leq j_1 \neq j_2 \leq 2^{47})$, thus the 2^{13} structures yield a total of 2^{106} plaintext pairs (P_{i,j_1}, P_{i,j_2}). There is a 17-bit filtering condition over the candidate ciphertext pairs, so it follows that about $2^{106} \times 2^{-17} = 2^{89}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) remain after Step 1.

In Step 2(a) there is an 8-bit filtering condition over the candidate ciphertext pairs, so it follows that about $2^{89} \times 2^{-8} = 2^{81}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) pass Step 2(a) for every guess of (W_1, W_4). Step 2(a) has a time complexity of $2 \times 2^{89} \times 2^{16} \times \frac{1}{4} \times \frac{1}{25} \approx 2^{99.36}$ 25-round HIGHT encryptions.

In Step 2(b) there is an 8-bit filtering condition over the candidate ciphertext pairs, so it follows that about $2^{81} \times 2^{-8} = 2^{73}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) pass Step 2(b) for every guess of (W_1, W_3, W_4, W_8). Step 2(b) has a time complexity of $2 \times 2^{81} \times 2^{32} \times \frac{1}{4} \times \frac{2}{25} \approx 2^{108.36}$ 25-round HIGHT encryptions.

In Step 2(c) there is a 1-bit filtering condition over the candidate ciphertext pairs (in Step 2(c)-i), so it follows that about $2^{73} \times 2^{-1} = 2^{72}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) pass Step 2(c) for every guess of $(W_1, W_2, W_3, W_4, W_7, W_8)$. Step 2(c) has a time complexity of $2 \times 2^{73} \times 2^{41} \times \frac{1}{4} \times \frac{1}{25} + 2 \times 2^{72} \times 2^{48} \times \frac{1}{4} \times \frac{1}{25} \approx 2^{114.36}$ 25-round HIGHT encryptions.

In Step 2(d) there is a 3-bit filtering condition over the candidate ciphertext pairs, so it follows that about $2^{72} \times 2^{-3} = 2^{69}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) pass Step 2(d) for every guess of $(W_1, W_2, W_3, W_4, W_7, W_8)$. Step 2(d) has a time complexity of $2 \times 2^{72} \times 2^{48} \times \frac{1}{4} \times \frac{1}{25} \approx 2^{114.36}$ 25-round HIGHT encryptions.

In Step 2(e) there is a 1-bit filtering condition over the candidate ciphertext pairs in every iteration, so it follows that about $2^{69} \times 2^{-8} = 2^{61}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) pass Step 2(e) for every guess of $(W_1, W_2, W_3, W_4, W_7, W_8, W_{16})$. Step 2(e) has a time complexity of $\sum_{l=0}^{7}(2 \times 2^{69-l} \times 2^{48+l+1} \times \frac{1}{4} \times \frac{1}{25}) \approx 2^{115.36}$ 25-round HIGHT encryptions.

In Step 2(f) there is a 7-bit filtering condition over the candidate ciphertext pairs, so it follows that about $2^{61} \times 2^{-7} = 2^{54}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) pass Step 2(f) for every guess of $(W_1, W_2, W_3, W_4, W_6, W_7, W_8, W_{16})$. Step 2(f) has a time complexity of $2 \times 2^{61} \times 2^{64} \times \frac{1}{4} \times \frac{2}{25} \approx 2^{120.36}$ 25-round HIGHT encryptions.

In Step 2(g) there is a 1-bit filtering condition over the candidate ciphertext pairs in every iteration, so it follows that about $2^{54} \times 2^{-7} = 2^{47}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) pass Step 2(g) for every guess of $(W_1, W_2, W_3, W_4, W_6, W_7, W_8, W_{15}, W_{16})$. Step 2(g) has a time complexity of $\sum_{l=0}^{6}(2 \times 2^{54-l} \times 2^{64+2+l} \times \frac{1}{4} \times \frac{1}{25}) \approx 2^{117.16}$ 25-round HIGHT encryptions.

In Step 2(h) there is a 1-bit filtering condition over the candidate ciphertext pairs in every iteration, so it follows that about $2^{47} \times 2^{-5} = 2^{42}$ ciphertext pairs (C_{i,j_1}, C_{i,j_2}) pass Step 2(h) for every guess of $(W_1, W_2, W_3, W_4, W_6, W_7, W_8, W_{11}, W_{15}, W_{16})$. Step 2(h) has a time complexity of $\sum_{l=0}^{4}(2 \times 2^{47-l} \times 2^{72+4+l} \times \frac{1}{4} \times \frac{1}{25}) \approx 2^{119.68}$ 25-round HIGHT encryptions.

In Step 3(a) there is an 8-bit filtering condition over the candidate plaintext pairs, so it follows that about $2^{42}\times 2^{-8} = 2^{34}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}) pass Step 3(a) for every guess of $(W_1, W_2, W_3, W_4, W_6, W_7, W_8, W_{11}, W_{15}, W_{16})$. Step 3(a) has a time complexity of $2 \times 2^{42} \times 2^{80} \times \frac{1}{4} \times \frac{1}{25} \approx 2^{116.36}$ 25-round HIGHT encryptions.

In Step 3(b) there is an 8-bit filtering condition over the candidate plaintext pairs, so it follows that about $2^{34}\times 2^{-8} = 2^{26}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}) pass Step 3(b) for every guess of $(W_1, W_2, W_3, W_4, W_6, W_7, W_8, W_{11}, W_{15}, W_{16})$. Step 3(b) has a time complexity of $2 \times 2^{34} \times 2^{80} \times \frac{1}{4} \times \frac{2}{25} \approx 2^{109.36}$ 25-round HIGHT encryptions.

In Step 3(c) there is an 8-bit filtering condition over the candidate plaintext pairs, so it follows that about $2^{26}\times 2^{-8} = 2^{18}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}) pass Step 3(c) for every guess of $(W_1, \cdots, W_8, W_{10}, W_{11}, W_{15}, W_{16})$. Step 3(c) has a time complexity of $2 \times 2^{26} \times 2^{96} \times \frac{1}{4} \times \frac{3}{25} \approx 2^{117.94}$ 25-round HIGHT encryptions.

In Step 3(d) there is an 8-bit filtering condition over the candidate plaintext pairs, so it follows that about $2^{18} \times 2^{-8} = 2^{10}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}) pass Step 3(d) for every guess of $(W_1, \cdots, W_{11}, W_{14}, W_{15}, W_{16})$. Step 3(d) has a time complexity of $2 \times 2^{18} \times 2^{112} \times \frac{1}{4} \times \frac{4}{25} \approx 2^{126.36}$ 25-round HIGHT encryptions.

In Step 3(e) there is an 8-bit filtering condition over the candidate plaintext pairs, so it follows that about $2^{120} \times (1 - 2^{-8})^{2^{10}} \approx 2^{120} \times \mathtt{e}^{-2^2} \approx 2^{114.24}$ guesses for $(W_1, \cdots, W_{11}, W_{13}, \cdots, W_{16})$ are recorded in Step 3(e), where $\mathtt{e}(= 2.71828\ldots)$ is the base of the natural logarithm. Thus, the expected number of wrong keys in Step 4 is about $2^{114.24} \times 2^8 \times 2^{-192} = 2^{-73.76}$. Therefore, it is very likely that we can find the correct key guess. Step 3(e) has a time complexity of about $2 \times 2^{120} \times [1 + (1 - 2^{-8}) + \cdots + (1 - 2^{-8})^{2^{10}}] \times \frac{1}{4} \times \frac{4}{25} \approx 2^{124.36}$ 25-round HIGHT encryptions. Step 4 has a time complexity of about $2^{122.24}$ 25-round HIGHT encryptions.

Therefore, the attack has a total time complexity of approximately $2^{126.78}$ 25-round HIGHT encryptions.

10.6 Related-Key Rectangle Attack on 26-Round HIGHT

In this section, we describe certain 18-round related-key rectangle distinguishers with probability $2^{-92.4}$ of HIGHT, such that we can mount a related-key rectangle attack on 26-round HIGHT.

10.6.1 18-Round Related-Key Rectangle Distinguishers with Probability $2^{-92.4}$

Let $\mathbf{E}^0$ denote Rounds 3 to 12 of HIGHT, and $\mathbf{E}^1$ denote Rounds 13 to 20 of HIGHT. The 18-round related-key rectangle distinguisher involves four cipher keys (TYPE 1 as described in Section 2.2.9), which we assume are K_A, K_B, K_C, K_D. The first related-key differential making up this 18-round distinguisher is the related-key differential $\Delta\alpha \rightarrow \Delta\beta$ with probability 2^{-12} for $\mathbf{E}^0$: $(0,0,0,0,0,e_8,e_{1,2,7},e_{2,4,6}) \rightarrow (0,0,0,0,e_8,e_{1,7,8},e_{2,6,7},0)$, where the key difference $K_A \oplus K_B = K_C \oplus K_D = (0,0,e_8,0,\cdots,0)$. The second related-key differential making up this 18-round distinguisher is the related-key differential $\Delta\gamma \rightarrow \Delta\delta$ with probability 2^{-9} for $\mathbf{E}^1$: $(0,e_8,e_{1,7,8},e_{3,6,7},0,0,0,0) \rightarrow (e_{1,2,7},0,0,0,0,0,0,e_8)$, where the key difference $K_A \oplus K_C = K_B \oplus K_D = (0,\cdots,0,e_8,0)$. See 10.3 for details of two related-key differentials.

We can compute a square sum of at least $6 \times (2^{-12})^2 + 20 \times (2^{-13})^2 + 20 \times (2^{-14})^2 + 72 \times (2^{-15})^2 \approx 2^{-19.98}$ for the probabilities of all the possible 10-round related-key differentials $\Delta\alpha \rightarrow \Delta\beta'$ for $\mathbf{E}^0$, as there are at least 6 possible β' with probability 2^{-12}, at least 20 possible β' with probability 2^{-13}, at least 20 possible β' with probability 2^{-14}, and at least 72 possible β' with probability 2^{-15}. We can also compute a square sum of at least $5 \times (2^{-9})^2 + 18 \times (2^{-10})^2 + 40 \times (2^{-11})^2 \approx 2^{-14.42}$ for the probabilities of all the possible 8-round related-key differentials $\Delta\gamma' \rightarrow \Delta\delta$ for $\mathbf{E}^1$, as there are at least 5 possible γ' with probability 2^{-9}, at least 18 possible γ' with probability 2^{-10}, and at least 40 possible γ' with probability 2^{-11}.

Therefore, this 18-round related-key rectangle distinguisher has a probability of at least $2^{-19.98} \times 2^{-14.42} \times 2^{-64} = 2^{-98.4}$ for the correct key, while it has a probability of $(2^{-64})^2 = 2^{-128}$ for a wrong key. We can further improve it by counting

Table 10.3: The two related-key differentials in the 18-round related-key rectangle distinguisher

Round(i)	$\Delta X_{i-1,1}$	$\Delta X_{i-1,2}$	$\Delta X_{i-1,3}$	$\Delta X_{i-1,4}$	$\Delta X_{i-1,5}$	$\Delta X_{i-1,6}$	$\Delta X_{i-1,7}$	$\Delta X_{i-1,8}$	subkey difference	Prob.
3	0	0	0	0	0	e_8	$e_{1,2,7}$	$e_{2,4,6}$	$(0,0,0,0)$	2^{-3}
4	0	0	0	0	0	0	e_8	$e_{1,2,7}$	$(0,0,0,0)$	2^{-3}
5	0	0	0	0	0	0	0	e_8	$(0,0,0,e_8)$	1
6	0	0	0	0	0	0	0	0	$(0,0,0,0)$	1
7	0	0	0	0	0	0	0	0	$(0,0,0,0)$	1
8	0	0	0	0	0	0	0	0	$(0,0,0,0)$	1
9	0	0	0	0	0	0	0	0	$(0,0,0,0)$	1
10	0	0	0	0	0	0	0	0	$(e_8,0,0,0)$	1
11	0	0	e_8	0	0	0	0	0	$(0,0,0,0)$	2^{-3}
12	0	0	0	e_8	$e_{1,7,8}$	0	0	0	$(0,0,0,0)$	2^{-3}
output	0	0	0	0	e_8	$e_{1,7,8}$	$e_{2,6,7}$	0	/	/
13	0	e_8	$e_{1,7,8}$	$e_{3,6,7}$	0	0	0	0	$(0,0,0,0)$	2^{-3}
14	0	0	e_8	$e_{1,7,8}$	0	0	0	0	$(0,0,0,0)$	2^{-3}
15	0	0	0	e_8	0	0	0	0	$(0,e_8,0,0)$	1
16	0	0	0	0	0	0	0	0	$(0,0,0,0)$	1
17	0	0	0	0	0	0	0	0	$(0,0,0,0)$	1
18	0	0	0	0	0	0	0	0	$(0,0,0,0)$	1
19	0	0	0	0	0	0	0	0	$(0,0,e_8,0)$	1
20	0	0	0	0	0	0	e_8	0	$(0,0,0,0)$	2^{-3}
output	$e_{1,2,7}$	0	0	0	0	0	0	e_8	/	/

many possible 8-round related-key differentials $\Delta\gamma' \rightarrow \Delta\delta'$ for every related-key differential $\Delta\gamma' \rightarrow \Delta\delta$ for $\mathbf{E}^1$. We count those that only have the output difference $(\Delta X_{20,0}, 0, 0, 0, 0, 0, 0, e_8)$ different from the 8-round differential $\Delta\gamma' \rightarrow \Delta\delta$; an analysis of this one-round differentials reveals that there are 4 possible $\Delta X_{20,0}$ (i.e. $e_{1,2,7}, e_{1,7}, e_{1,7,8}, e_{1,2,7,8}$) with probability 2^{-3}, 4 possible $\Delta X_{20,0}$ with probability 2^{-4}, 4 possible $\Delta X_{20,0}$ with probability 2^{-5}, 4 possible $\Delta X_{20,0}$ with probability 2^{-6}, and 8 possible $\Delta X_{20,0}$ with probability 2^{-7}. Actually, these are all the 24 possible output differences of the last one-round differentials; we denote them by the set Ω. As a result, the distinguisher now has a probability of at least $2^{-19.98} \times (4 \times 2^{-7.21} + 4 \times 2^{-8.21} + 4 \times 2^{-9.21} + 4 \times 2^{-10.21} + 8 \times 2^{-11.21})^2 \times 2^{-64} = 2^{-92.4}$ for the correct key, while it has a probability of $(24 \times 2^{-64})^2 \approx 2^{-118.83}$ for a wrong key.

We note that this distinguisher can be extended to a distinguisher that operates on more rounds, by appending one or more rounds at the ends; however, we will conduct a key recovery on these rounds such that a less data complexity is required. Similar related-key rectangle distinguishers exist for some other series of 18 rounds.

10.6.2 Attacking Rounds 1 to 26

The output difference $(x, 0, 0, 0, 0, 0, 0, e_8)$ of this distinguisher will propagate to a difference $(e_8, x, \star, 0, 0, 0, 0, 0)$ just after Round 21, where $x \in \Omega$, which will then propagate to a difference $(0, e_8, e_{1,\sim}, \star, \star, 0, 0, 0)$ just after Round 22, to a difference $(e_8, 0, e_8, e_{1,\sim}, \star, \star, \star, 0)$ just after Round 23 (due to the subkey difference in Round 23), and a difference $(\star, e_8, e_{3,\sim}, e_8, \overline{e}_{1,\sim}, \star, \star, \star)$ just after Round 24. This property allows us to use the early abort technique described in Section 4.4 to break Rounds 21 and 24.

The above analysis enables us to give the following related-key rectangle attack on the first 26 rounds of HIGHT with the final transformation only. Note that the same 64 user key bits are used in Rounds 1, 2, 25 and 26 as well as the final transformation. To get the difference $(0, 0, 0, 0, 0, e_8, e_{1,2,7}, e_{2,4,6})$ just before Round 3, the input difference to Round 1 must have the form $(0, 0, 0, e_8, e_{1,\sim}, \star, e_{1,\sim}, \star)$, with 31 bits definitely being zero. We conduct the early abort in an optimized order, according to the output differences of the distinguishier.

10.6.2.1 Attack Description

1. Choose $2^{16.2}$ structures S_i, $(i = 1, 2, \cdots, 2^{16.2})$, where a structure S_i is defined to be a set of 2^{33} plaintexts $P_{i,l}$ with the first three bytes and bits (1,2,$\cdots$,7) of the fourth byte fixed, and the remaining 33 bit positions taking all the possible values, $(l = 1, 2, \cdots, 2^{33})$. In a chosen-plaintext attack scenario, obtain all the ciphertexts for the 2^{33} plaintexts in each of the $2^{16.2}$ structures encrypted with K_A, K_B, K_C, K_D, where $K_A \oplus K_B = K_C \oplus K_D = (0, 0, e_8, 0, \cdots, 0)$ and $K_A \oplus K_C = K_B \oplus K_D = (0, \cdots, 0, e_8, 0)$. We denote by $C_{i,l}, C^*_{i,l}, C'_{i,l}, C'^*_{i,l}$ the ciphertexts for plaintext $P_{i,l}$ encrypted respectively with K_A, K_B, K_C, K_D.

2. Guess a value for the 8 key bytes $(W_1, \cdots, W_8)$, compute the subkeys $(KS_1, \cdots, KS_8)$, and perform Steps (a) and (b) below.

   (a) Partially encrypt every plaintext $P_{i,l}$ through Rounds 1 and 2 with $(KS_1, \cdots, KS_8)$ to get the corresponding value just after Round 2; we denote it by $x_{i,l}$. Then, partially decrypt $x_{i,l} \oplus (0, 0, 0, 0, 0, e_8, e_{1,2,7}, e_{2,4,6})$ through

   Rounds 1 and 2 with $(KS_1, KS_2, KS_3 \oplus e_8, KS_4, \cdots, KS_8)$ to get its plaintext; we denote it by $\widetilde{P}_{i,l}$. Find $\widetilde{P}_{i,l}$ in S_i. We denote by $\widetilde{C}_{i,l}, \widetilde{C}^*_{i,l}, \widetilde{C}'_{i,l}$ and $\widetilde{C}'^*_{i,l}$ the corresponding ciphertexts for $\widetilde{P}_{i,l}$ encrypted under K_A, K_B, K_C and K_D, respectively.

   (b) Compute the subkeys $(KS_{97}, \cdots, KS_{104}, KW_5, \cdots, KW_8)$ with the $(W_1, \cdots, W_8)$ guessed above. Then, partially decrypt all the $C_{i,l}$ and $C^{'}_{i,l}$ with these subkeys to get the corresponding values just before Round 25; we denote them by $T_{i,l}$ and $T^{'}_{i,l}$, respectively. Partially decrypt all the $\widetilde{C}^*_{i,l}$ and $\widetilde{C}^{*'}_{i,l}$ with the related subkeys $(KS_{97} \oplus e_8, KS_{98}, KS_{99}, \cdots, KS_{104}, KW_5, KW_6, KW_7 \oplus e_8, KW_8)$ to get the corresponding values just before Round 25; we denote them by $\widetilde{T}^*_{i,l}$ and $\widetilde{T}^{*'}_{i,l}$, respectively. Store $(T_{i,l}, T^{'}_{i,l}, \widetilde{T}^*_{i,l}, \widetilde{T}^{*'}_{i,l})$ in a hash table. Finally, choose only the quartets $(T_{i_1,l_1}, \widetilde{T}^*_{i_1,l_1}, T^{'}_{i_2,l_2}, \widetilde{T}'^*_{i_2,l_2})$ such that both $T_{i_1,l_1} \oplus T^{'}_{i_2,l_2}$ and $\widetilde{T}^*_{i_1,l_1} \oplus \widetilde{T}'^*_{i_2,l_2}$ have the form $(\star, e_8, e_{3,\sim}, e_8, \overline{e}_{1,\sim}, \star, \star, \star)$, where $1 \leq i_1 \leq i_2 \leq 2^{16.2}$ and $1 \leq l_1, l_2 \leq 2^{33}$. If six or more quartets $(T_{i_1,l_1}, \widetilde{T}^*_{i_1,l_1}, T^{'}_{i_2,l_2}, \widetilde{T}^{*'}_{i_2,l_2})$ pass this test, execute Step 3 with the quartets meeting this condition; otherwise, repeat Step 2 with another guess.

3. Perform Steps (a) and (b) below for $j = 1$ to 8:

   (a) Guess a value for the jth bit $W_{11,j}$ of the key byte W_{11}, and compute the j-bit subkey $KS_{96,[1,j]}$ of the subkey KS_{96}.

   (b) For every remaining quartet $(T_{i_1,l_1}, \widetilde{T}^*_{i_1,l_1}, T^{'}_{i_2,l_2}, \widetilde{T}^{*'}_{i_2,l_2})$, partially decrypt bytes (1,8) of T_{i_1,l_1} and $T^{'}_{i_2,l_2}$ with $KS_{96,[1,j]}$ to get the corresponding values for bits $(1, 2, \cdots, j)$ of byte (8) just before Round 24, and check whether they have a zero difference. If not, repeat this step with another quartet; otherwise, partially decrypt bytes (1,8) of $\widetilde{T}^*_{i_1,l_1}$ and $\widetilde{T}^{*'}_{i_2,l_2}$ with $KS_{96,[1,j]}$ to get the corresponding values for bits $(1, 2, \cdots, j)$ of byte (8) just before Round 24, and check whether they have a zero difference as well. If six or more quartets pass this condition, execute next iteration (Step 4 when $j = 8$) with the quartets meeting this condition; otherwise, repeat Step 3(a) with another guess.

4. Guess a value for the key byte W_{10}, and compute the subkey KS_{95}. Perform Steps (a) and (b) below for $j = 1$ to 8.

   (a) Guess a value for the jth bit $W_{14,j}$ of the key byte W_{14}, and compute the

j-bit subkey $KS_{91,[1,j]}$ of the subkey KS_{91}.

(b) For every remaining quartet $(T_{i_1,l_1}, \widetilde{T}^*_{i_1,l_1}, T^{'}_{i_2,l_2}, \widetilde{T}^{*'}_{i_2,l_2})$, partially decrypt bytes (6,7) of T_{i_1,l_1} and $T^{'}_{i_2,l_2}$ with $(KS_{95}, KS_{91,[1,j]})$ to get the corresponding values for bits $(1, 2, \cdots, j)$ of byte (6) just before Round 23, and check whether they have a zero difference. If not, repeat this step with another quartet; otherwise, partially decrypt bytes (6,7) of $\widetilde{T}^*_{i_1,l_1}$ and $\widetilde{T}^{*'}_{i_2,l_2}$ with $(KS_{95}, KS_{91,[1,j]})$ to get the corresponding values for bits $(1, 2, \cdots, j)$ of byte (6) just before Round 23, and check whether they have a zero difference as well. If six or more quartets pass this condition, execute next iteration (Step 5 when $j = 8$) with the quartets meeting this condition; otherwise, repeat Step 4(a) with another guess, (if all the guesses for $W_{14,j}$ are tested, repeat Step 4 with another guess for W_{10}).

5. Guess a value for the least significant 3 bits $W_{16,[1,3]}$ of the key byte W_{16}. Perform Steps (a) and (b) below for $j = 4$ to 8.

   (a) Guess a value for the jth bit $W_{16,j}$ of W_{16}, and compute the j-bit subkey $KS_{93,[1,j]}$ of the subkey KS_{93}.

   (b) For every remaining quartet $(T_{i_1,l_1}, \widetilde{T}^*_{i_1,l_1}, T^{'}_{i_2,l_2}, \widetilde{T}^{*'}_{i_2,l_2})$, partially decrypt bytes (2,3) of T_{i_1,l_1} and $T^{'}_{i_2,l_2}$ with $KS_{93,[1,j]}$ to get the corresponding values for bits $(1, 2, \cdots, j)$ of byte (2) just before Round 24, and check whether they have a zero difference. If not, repeat this step with another quartet; otherwise, partially decrypt bytes (2,3) of $\widetilde{T}^*_{i_1,l_1}$ and $\widetilde{T}^{*'}_{i_2,l_2}$ with $KS_{93,[1,j]}$ to get the corresponding values for bits $(1, 2, \cdots, j)$ of byte (2) just before Round 24, and check whether they have a zero difference as well. If six or more quartets pass this condition, execute next iteration (Step 6 when $j = 8$) with the quartets meeting this condition; otherwise, repeat Step 5(a) with another guess.

6. Guess a value for the key bytes (W_9, W_{13}), compute the subkeys (KS_{90}, KS_{94}), and compute the subkey KS_{86} with the W_1 guessed in Step 2. For every remaining quartet $(T_{i_1,l_1}, \widetilde{T}^*_{i_1,l_1}, T^{'}_{i_2,l_2}, \widetilde{T}^{*'}_{i_2,l_2})$, partially decrypt bytes (4,5) of T_{i_1,l_1} and $T^{'}_{i_2,l_2}$ with $(KS_{94}, KS_{90}, KS_{86})$ to get the corresponding values for bytes (3,4) just before Round 22, and check whether they have a difference $(\star, 0)$. If not, repeat this step with another quartet; otherwise, partially decrypt bytes (4,5) of $\widetilde{T}^*_{i_1,l_1}$ and $\widetilde{T}^{*'}_{i_2,l_2}$ with $(KS_{94}, KS_{90}, KS_{86})$ to get the corresponding val-

ues for bytes (3,4) just before Round 22, and check whether they have a difference $(\star, 0)$ as well. If six or more quartets pass this condition, execute Step 7 with the quartets meeting this condition; otherwise, repeat this step with another guess.

Now, for every remaining quartet $(T_{i_1,l_1}, \widetilde{T}^*_{i_1,l_1}, T^{'}_{i_2,l_2}, \widetilde{T}^{*'}_{i_2,l_2})$, we obtain the corresponding values just before Round 24 under the guess for $(W_1, \cdots, W_{11}, W_{13}, W_{14}, W_{16})$; we denote them by $(Q_{i_1,l_1}, \widetilde{Q}^*_{i_1,l_1}, Q^{'}_{i_2,l_2}, \widetilde{Q}^{*'}_{i_2,l_2})$, respectively.

7. Guess a value for the key byte W_{12}, compute the subkey KS_{89}, and compute the subkey KS_{85} with the W_8 guessed in Step 2. For every quartet $(Q_{i_1,l_1}, \widetilde{Q}^*_{i_1,l_1}, Q^{'}_{i_2,l_2}, \widetilde{Q}^{*'}_{i_2,l_2})$, partially decrypt bytes (2,3) of Q_{i_1,l_1} and $Q^{'}_{i_2,l_2}$ with (KS_{89}, KS_{85}) to get the corresponding values for bytes (1,2) just before Round 22, and check whether they have a difference belonging to the set $\{(e_8, x)|x \in \Omega\}$. If not, repeat this step with another quartet; otherwise, partially decrypt bytes (2,3) of $\widetilde{Q}^*_{i_1,l_1}$ and $\widetilde{Q}^{*'}_{i_2,l_2}$ with (KS_{89}, KS_{85}) to get the corresponding values for bytes (1,2) just before Round 22, and check whether they have a difference belonging to the set $\{(e_8, x)|x \in \Omega\}$. If six or more quartets $(Q_{i_1,l_1}, \widetilde{Q}^*_{i_1,l_1}, Q^{'}_{i_2,l_2}, \widetilde{Q}^{*'}_{i_2,l_2})$ pass this test, execute Step 8 with the quartets meeting this condition; otherwise, repeat this step with another guess for W_{12}.

8. Compute the subkey KS_{81} with the W_4 guessed in Step 2. For every remaining quartet $(Q_{i_1,l_1}, \widetilde{Q}^*_{i_1,l_1}, Q^{'}_{i_2,l_2}, \widetilde{Q}^{*'}_{i_2,l_2})$, since we already obtain the corresponding values for bytes (1,2) just before Round 22, we can partially decrypt them with KS_{81} to check whether the corresponding values for byte (2) just before Round 21 for $(Q_{i_1,l_1}, Q^{'}_{i_2,l_2})$ have a zero difference, and check whether the corresponding values for byte (2) just before Round 21 for $(\widetilde{Q}^*_{i_1,l_1}, \widetilde{Q}^{*'}_{i_2,l_2})$ have a zero difference as well. If six or more quartets $(Q_{i_1,l_1}, \widetilde{Q}^*_{i_1,l_1}, Q^{'}_{i_2,l_2}, \widetilde{Q}^{*'}_{i_2,l_2})$ pass this test, record the guessed value for $(W_1, \cdots, W_{14}, W_{16})$, and go to Step 9; otherwise, repeat Step 7 with another guess for W_{12}.

9. For a recorded value for $(W_1, \cdots, W_{14}, W_{16})$, exhaustively search for the remaining 8 key bits using a known plaintext/ciphertext pair. If a 128-bit key is suggested, output it as the user key of the 26-round HIGHT; otherwise, go to Step 2 (If all the guesses are tested during any of Steps 3 to 8, repeat its previous steps with another guess).

10.6.2.2 Complexity Analysis

The attack requires $2^{51.2}$ (related-key) chosen plaintexts, which take a time complexity of $2^{51.2}$ 26-round HIGHT encryptions.

In Step 2(a), about $2^{16.2} \times \frac{2^{33}}{2} = 2^{48.2}$ plaintext pairs are yielded for every guess of $(W_1, \cdots, W_8)$, which produce the difference $(0, 0, 0, 0, 0, e_8, e_{1,2,7}, e_{2,4,6})$ just before Round 3 under the key guess, thus about $\binom{2^{48.2}}{2} \approx \frac{2^{48.2 \times 2}}{2} = 2^{95.4}$ candidate quartets are constructed for every guess of $(W_1, \cdots, W_8)$. To produce the output difference $(x, 0, 0, 0, 0, 0, 0, e_8)$ just before Round 21, where $x \in \Omega$, the two pairs $(T_{i_1,l_1}, T^{'}_{i_2,l_2})$ and $(\widetilde{T}^{*}_{i_1,l_1}, \widetilde{T}^{'*}_{i_2,l_2})$ in a candidate quartet $(T_{i_1,l_1}, \widetilde{T}^{*}_{i_1,l_1}, T^{'}_{i_2,l_2}, \widetilde{T}^{'*}_{i_2,l_2})$ must have a difference of the form $(\star, e_8, e_{3,\sim}, e_8, \overline{e}_{1,\sim}, \star, \star, \star)$ just before Round 25, so a candidate quartet that does not meet this filtering condition is an incorrect quartet. Step 2(a) has about $2 \times 2^{49.2} \times 2^{64} \times \frac{1}{2} \times \frac{2}{26} \approx 2^{109.5}$ 26-round HIGHT encryptions, where $\frac{1}{2}$ means the average fraction of the guessed keys that are tested in the step.

In Step 2(b), either of the pairs $(T_{i_1,l_1}, T^{'}_{i_2,l_2})$ and $(\widetilde{T}^{*}_{i_1,l_1}, \widetilde{T}^{'*}_{i_2,l_2})$ meets the condition with a probability of 2^{-20}, thus about $2^{95.4} \times (2^{-20})^2 = 2^{55.4}$ candidate quartets $(T_{i_1,l_1}, \widetilde{T}^{*}_{i_1,l_1}, T^{'}_{i_2,l_2}, \widetilde{T}^{'*}_{i_2,l_2})$ remain after 2(b) for every guess of $(W_1, \cdots, W_8)$. The probability that 6 or more quartets pass the condition is $\sum_{i=6}^{2^{95.4}} [\binom{2^{95.4}}{i} \cdot (2^{-40})^i \cdot (1 - 2^{-40})^{2^{95.4}-i}] \approx 1$, so it is expected that almost all the 2^{64} guesses for $(W_1, \cdots, W_8)$ will pass Step 2(b). The time complexity of Step 2(b) is dominated by the partial decryptions, which is about $4 \times 2^{49.2} \times 2^{64} \times \frac{1}{2} \times \frac{2}{26} \approx 2^{110.5}$ 26-round HIGHT encryptions.

In Step 3(b), the probability that a quartet meets either of the filtering conditions in every iteration is 2^{-1}, so it follows that all the 2^{72} guesses for $(W_1, \cdots, W_8, W_{11})$ will past Step 3, and for a wrong guess it is expected about $2^{55.4} \times 2^{-1 \times 2 \times 8} = 2^{39.4}$ quartets remain after Step 3. Step 3 has a time complexity of about $\sum_{l=0}^{7}(2 \times 2^{55.4-2 \times l} \times 2^{65+l} \times \frac{1}{2} \times \frac{1}{4} \times \frac{1}{26} + 2 \times 2^{55.4-(2 \times l+1)} \times 2^{65+l} \times \frac{1}{2} \times \frac{1}{4} \times \frac{1}{26}) \approx 2^{115.28}$ 26-round HIGHT encryptions.

In Step 4(b), the probability that a quartet meets either of the filtering conditions in every iteration is 2^{-1}, so it is expected that all the 2^{88} guesses for $(W_1, \cdots, W_8, W_{10}, W_{11}, W_{14})$ will past this step, and for a wrong guess about $2^{39.4} \times 2^{-1 \times 2 \times 8} =$

$2^{23.4}$ quartets remain after Step 4. Step 4 has a time complexity of about $\sum_{l=0}^{7}(2 \times 2^{39.4-2\times l} \times 2^{81+l} \times \frac{1}{2} \times \frac{1}{4} \times \frac{2}{26} + 2 \times 2^{39.4-(2\times l+1)} \times 2^{81+l} \times \frac{1}{2} \times \frac{1}{4} \times \frac{2}{26}) \approx 2^{116.28}$ 26-round HIGHT encryptions.

In Step 5(b), the probability that a quartet meets either of the filtering conditions in every iteration is 2^{-1}, so it is expected that all the 2^{96} guesses for $(W_1, \cdots, W_8, W_{10}, W_{11}, W_{14}, W_{16})$ will past this step, and for a wrong guess about $2^{23.4} \times 2^{-1\times 2\times 5} = 2^{13.4}$ quartets remain after Step 5. Step 5 has a time complexity of about $\sum_{l=0}^{4}(2 \times 2^{23.4-2\times l} \times 2^{92+l} \times \frac{1}{2} \times \frac{1}{4} \times \frac{1}{26} + 2 \times 2^{23.4-(2\times l+1)} \times 2^{92+l} \times \frac{1}{2} \times \frac{1}{4} \times \frac{1}{26}) \approx 2^{110.24}$ 26-round HIGHT encryptions.

In Step 6, the probability that a quartet meets the filtering conditions is $2^{-8\times 2} = 2^{-16}$, so for a wrong guess about $2^{13.4} \times 2^{-16} = 2^{-2.6}$ quartets remain after Step 6, and the probability that 6 or more quartets pass the tests for a wrong guess is approximately $\sum_{i=6}^{2^{13.4}}[\binom{2^{13.4}}{i} \cdot (2^{-16})^i \cdot (1-2^{-16})^{2^{13.4}-i}] \approx 2^{-25.09}$, thus it is expected that about $2^{112} \times 2^{-25.09} = 2^{86.91}$ guesses for $(W_1, \cdots, W_{11}, W_{13}, W_{14}, W_{16})$ pass Step 6. Step 6 has a time complexity of about $2 \times 2^{13.4} \times 2^{112} \times \frac{1}{2} \times \frac{1}{4} \times \frac{3}{26} + 2 \times 2^{5.4} \times 2^{112} \times \frac{1}{2} \times \frac{1}{4} \times \frac{3}{26} \approx 2^{120.28}$ 26-round HIGHT encryptions.

In Step 7, the probability that a quartet meets the filtering conditions is $(\frac{24}{2^7})^2 = 2^{-4.83}$, and the probability that 6 or more quartets pass the tests for a wrong guess is approximately $(2^{-4.83})^6 \approx 2^{-28.98}$, so it is expected about $2^{86.91+8} \times 2^{-28.98} = 2^{65.93}$ guesses for $(W_1, \cdots, W_{14}, W_{16})$ pass Step 7. Step 7 has a time complexity of about $2 \times 6 \times 2^{94.91} \times \frac{1}{2} \times \frac{1}{4} \times \frac{2}{26} + 2 \times 6 \times 2^{92.5} \times \frac{1}{2} \times \frac{1}{4} \times \frac{2}{26} \approx 2^{92.05}$ 26-round HIGHT encryptions.

In Step 8, the probability that 6 or more quartets pass the tests for a wrong guess is approximately $(2^{-8\times 2})^6 = 2^{-96}$, thus it is expected about $2^{65.93} \times 2^{-96} = 2^{-30.07}$ guesses for $(W_1, \cdots, W_{14}, W_{16})$ pass Step 8. Therefore, it is expected that we can find the correct user key with 2^8 trials in Step 9. Step 8 has a time complexity of about $4 \times 6 \times 2^{65.93} \times \frac{1}{2} \times \frac{1}{4} \times \frac{1}{26} \approx 2^{62.81}$ 26-round HIGHT encryptions.

Therefore, the attack has a total time complexity of about $2^{120.41}$ 26-round HIGHT encryptions.

In Step 8, it is expected about $2^{95.4} \times 2^{-92.4} = 8$ quartets pass the filtering condition for the correct key, and the probability that 6 or more quartets pass the test for the correct key guess is approximately $\sum_{i=6}^{2^{95.4}} [\binom{2^{95.4}}{i} \cdot (2^{-92.4})^i \cdot (1 - 2^{-92.4})^{2^{95.4}-i}] \approx 0.8$. Therefore, the related-key rectangle attack can break the 26-round HIGHT with a success probability of 80%.

10.7 Related-Key Impossible Differential Attack on 28-Round HIGHT

In this section, we describe certain 19-round related-key impossible differentials of HIGHT, which enable us to conduct a related-key impossible differential attack on 28-round HIGHT.

10.7.1 19-Round Related-Key Impossible Differentials

We describe certain 19-round related-key impossible differentials: $(0, 0, 0, 0, 0, 0, 0, e_8) \nrightarrow (e_{1,\sim}, 0, 0, 0, 0, 0, 0, 0)$, where the key difference $(\Delta W_1, \Delta W_2, \cdots, \Delta W_{16})$ is $(0, \cdots, 0, e_8, 0, 0, 0, 0, 0)$, which start from Round 7 and end at Round 25.

They are also built in a miss-in-the-middle manner: a 12-round related-key differential with probability 1 is concatenated with a 7-round related-key differential with probability 1, where the second byte of the output difference of the 12-round related-key differential is $\overline{e}_{1,\sim}$, and the second byte of the difference of the 7-round related-key differential is $e_{1,\sim}$, which contradict with each other. See 10.4 for more details of the two related-key differentials.

Due to the key difference, the input difference $(0, 0, 0, 0, 0, 0, 0, e_8)$ to Round 7 will be canceled to zero by the subkey difference in Round 7. The zero difference will be kept until the input of Round 12, for the subkey differences in Rounds 8 to 11 are all zero. Since the subkey difference $(\Delta KS_{45}, \Delta KS_{46}, \Delta KS_{47}, \Delta KS_{48})$ in Round 12 is $(e_8, 0, 0, 0)$, the input difference to Round 13 is $(0, 0, e_8, 0, 0, 0, 0, 0)$, which propagates to a difference $(0, 0, 0, e_8, e_{1,\sim}, 0, 0, 0)$ just after Round 13. Then, the difference $(0, 0, 0, e_8, e_{1,\sim}, 0, 0, 0)$ propagates to a difference $(\star, 0, 0, 0, 0, e_8, e_{1,\sim}, \star)$ just before

Table 10.4: The two related-key differentials in the 19-round related-key impossible differential

Round(i)	$\Delta X_{i-1,1}$	$\Delta X_{i-1,2}$	$\Delta X_{i-1,3}$	$\Delta X_{i-1,4}$	$\Delta X_{i-1,5}$	$\Delta X_{i-1,6}$	$\Delta X_{i-1,7}$	$\Delta X_{i-1,8}$	subkey difference
7	0	0	0	0	0	0	0	e_8	$(0, 0, 0, e_8)$
8	0	0	0	0	0	0	0	0	$(0, 0, 0, 0)$
$\vdots$				$\vdots$					$\vdots$
11	0	0	0	0	0	0	0	0	$(0, 0, 0, 0)$
12	0	0	0	0	0	0	0	0	$(e_8, 0, 0, 0)$
13	0	0	e_8	0	0	0	0	0	$(0, 0, 0, 0)$
14	0	0	0	e_8	$e_{1,\sim}$	0	0	0	$(0, 0, 0, 0)$
15	0	0	0	0	e_8	$e_{1,\sim}$	$\star$	0	$(0, 0, 0, 0)$
16	$\star$	0	0	0	0	e_8	$e_{1,\sim}$	$\star$	$(0, e_8, 0, 0)$
17	$\star$	$\star$	$\star$	0	e_8	0	e_8	$e_{1,\sim}$	$(0, 0, 0, 0)$
18	$\overline{e}_{1,\sim}$	$\star$	$\star$	$\star$	$\star$	e_8	$e_{3,\sim}$	e_8	$(0, 0, 0, 0)$
output	$\star$	$\overline{e}_{1,\sim}$	$\star$	$\star$	$\star$	$\star$	$\star$	$e_{3,\sim}$	$(0, 0, 0, 0)$
19	0	$e_{1,\sim}$	$\star$	$\star$	$\star$	$\star$	$\star$	$\star$	$(0, 0, 0, 0)$
20	0	0	$e_{1,\sim}$	$\star$	$\star$	$\star$	$\star$	$\star$	$(0, 0, e_8, 0)$
21	0	0	0	$e_{1,\sim}$	$\star$	$\star$	$\star$	$\star$	$(0, 0, 0, 0)$
22	0	0	0	0	$e_{1,\sim}$	$\star$	$\star$	$\star$	$(0, 0, 0, 0)$
23	0	0	0	0	0	$e_{1,\sim}$	$\star$	$\star$	$(0, 0, 0, 0)$
24	0	0	0	0	0	0	$e_{1,\sim}$	$\star$	$(0, 0, 0, e_8)$
25	0	0	0	0	0	0	0	$e_{1,\sim}$	$(0, 0, 0, 0)$
output	$e_{1,\sim}$	0	0	0	0	0	0	0	$(0, 0, 0, 0)$

Round 16. Since the subkey difference $(\Delta KS_{61}, \Delta KS_{62}, \Delta KS_{63}, \Delta KS_{64})$ in Round 16 is $(0, e_8, 0, 0)$, the output difference of Round 16 is $(\star, \star, \star, 0, e_8, 0, e_8, e_{1,\sim})$, which propagates to a difference $(\overline{e}_{1,\sim}, \star, \star, \star, \star, e_8, e_{3,\sim}, e_8)$ just after Round 17. Finally, we can learn that the output difference of Round 18 has the form $(\star, \overline{e}_{1,\sim}, \star, \star, \star, \star, \star, e_{3,\sim})$.

On the other hand, when we roll back the output difference $(e_{1,\sim}, 0, 0, 0, 0, 0, 0, 0)$ of Round 25 through seven rounds of HIGHT in the reverse direction, we will definitely get an input difference $(0, e_{1,\sim}, \star, \star, \star, \star, \star, \star)$ to Round 19.

Now, a contradiction occurs between the intermediate differences of these two differentials, because the second byte of the output difference of the 12-round related-key differential is $\overline{e}_{1,\sim}$, while the second byte of the difference of the 7-round related-key differential is $e_{1,\sim}$. Therefore, these 19-round related-key differentials are impossible.

10.7.2 Attack Rounds 2 to 29

The 19-round related-key impossible differentials can be used to break the 28 rounds from Rounds 2 to 29 of HIGHT with only the final transformation, similar to that given in Section 10.5.2. The main difference between them lies in that here we compute the related-key difference between a pair of data. The attack procedure is as follows.

10.7.2.1 Attack Description

1. Choose 2^{19} structures S_i, $(i = 1, 2, \cdots, 2^{19})$, where a structure is defined to be a set of 2^{40} plaintexts $P_{i,j}$ with the first two bytes, bits (1,2,$\cdots$,7) of the third byte and bit (1) of the fourth byte fixed, and the other 40 bit positions taking all the possible values, $(j = 1, 2, \cdots, 2^{40})$. In a chosen-plaintext attack scenario, obtain all the ciphertexts of the 2^{40} plaintexts in each of the 2^{19} structures encrypted with K_A and K_B, where $K_A \oplus K_B = (0, \cdots, 0, e_8, 0, 0, 0, 0, 0)$; we denote by $C_{i,j}$ and $\widetilde{C}_{i,j}$ for the ciphertexts for $P_{i,j}$ encrypted respectively with K_A and K_B. Choose only the ciphertext pairs $(C_{i,j_1}, \widetilde{C}_{i,j_2})$ with difference $(0, 0, 0, e_{1,\sim}, \star, \star, \star, \star)$, where $1 \leq j_1 \neq j_2 \leq 2^{40}$.

2. Guess a value for the key bytes (W_4, W_5), compute the subkeys (KW_8, KS_{116}), and perform Steps (a)–(c) below.

   (a) Partially decrypt every remaining ciphertext pair $(C_{i,j_1}, \widetilde{C}_{i,j_2})$ with (KW_8, KS_{116}) to get the corresponding values for bytes (7,8) just before Round 29, and check whether they have a difference $(\star, 0)$. Keep only the pairs that meet this condition.

   (b) Guess a value for the two key bytes (W_3, W_9), compute the subkeys (KW_7, KS_{111}), and compute the subkey KS_{115} with the W_4 guessed above. Partially decrypt every remaining ciphertext pair $(C_{i,j_1}, \widetilde{C}_{i,j_2})$ with $(KW_7, KS_{111}, KS_{115})$ to get the corresponding values for bytes (5,6) just before Round 28, and check whether they have a difference $(\star, 0)$. Keep only the pairs that meet this condition.

   (c) Guess a value for the three key bytes (W_2, W_{12}, W_{16}), compute the subkeys $(KW_6, KS_{106}, KS_{110})$, and compute the subkey KS_{114} with the W_3

guessed above. Partially decrypt every remaining ciphertext pair $(C_{i,j_1}, \widetilde{C}_{i,j_2})$ with $(KW_6, KS_{106}, KS_{110}, KS_{114})$ to get the corresponding values for bytes (3,4) just before Round 27, and check whether they have a difference $(\star, 0)$. Keep only the pairs that meet this condition.

3. Perform Steps (a)–(c) below for a plaintext pair $(P_{i,j_1}, \widetilde{P}_{i,j_2})$ corresponding to a remaining ciphertext pair $(C_{i,j_1}, \widetilde{C}_{i,j_2})$.

   (a) Guess a value for the key byte W_8, and compute the subkey KS_8. Partially encrypt every plaintext pair $(P_{i,j_1}, \widetilde{P}_{i,j_2})$ with KS_8 to get the corresponding values for bytes (1,8) just after Round 2, and check whether they have a difference $(0, \star)$. Keep only the pairs that meet this condition.

   (b) Guess a value for the key byte W_7, compute the subkey KS_7, and compute the subkey KS_{12} with the W_{12} guessed above. Partially encrypt every remaining plaintext pair $(P_{i,j_1}, \widetilde{P}_{i,j_2})$ with (KS_7, KS_{12}) to get the corresponding values for bytes (1,8) just after Round 3, and check whether they have a difference $(0, \star)$. Keep only the pairs that meet this condition.

   (c) Guess a value for the two key bytes (W_6, W_{11}), compute the subkeys (KS_6, KS_{11}), and compute the subkey KS_{16} with the W_{16} guessed above. For every remaining pair plaintext $(P_{i,j_1}, \widetilde{P}_{i,j_2})$, partially encrypt P_{i,j_1} with (KS_6, KS_{11}, KS_{16}) to get the corresponding value for bytes (1,8) just after Round 4, and partially decrypt $\widetilde{P}_{i,j_2}$ with $(KS_6, KS_{11} \oplus e_8, KS_{16})$ to get the corresponding value for bytes (1,8) just after Round 4. Check whether they have a difference $(0, \star)$. Keep only the pairs that meet this condition.

4. Guess a value for the key bytes (W_1, W_{15}), compute the subkeys (KW_5, KS_{109}), and compute the subkeys $(KS_{101}, KS_{105}, KS_{113})$ with the (W_2, W_7, W_{11}) guessed above. For every ciphertext pair $(C_{i,j_1}, \widetilde{C}_{i,j_2})$ corresponding to a remaining plaintext pair $(P_{i,j_1}, \widetilde{P}_{i,j_2})$, partially decrypt C_{i,j_1} with $(KW_5, KS_{101}, KS_{105}, KS_{109}, KS_{113})$ to get the corresponding value for bytes (1,2) just before Round 26, and partially decrypt $\widetilde{C}_{i,j_2}$ with $(KW_5, KS_{101}, KS_{105} \oplus e_8, KS_{109}, KS_{113})$ to get the corresponding value for bytes (1,2) just before Round 26. Check whether they have a difference $(e_{1,\sim}, 0)$. Keep only the pairs that meet this condition.

5. Perform Steps (a) and (b) below for a plaintext pair $(P_{i,j_1}, \widetilde{P}_{i,j_2})$ corresponding

to a remaining ciphertext pair $(C_{i,j_1}, \widetilde{C}_{i,j_2})$.

(a) Guess a value for the key byte W_{10}, compute the subkey KS_{10}, and compute the subkeys (KS_5, KS_{15}, KS_{20}) with the (W_3, W_5, W_{15}) guessed above. Partially encrypt every remaining plaintext pair $(P_{i,j_1}, \widetilde{P}_{i,j_2})$ with $(KS_5, KS_{10}, KS_{15}, KS_{20})$ to get the corresponding values for bytes (1,8) just after Round 5, and check whether they have a difference $(0, e_{1,\sim})$. Keep only the pairs that meet this condition.

(b) Guess a value for the key byte W_{14}, compute the subkey KS_{14}, and compute the subkeys (KS_9, KS_{19}, KS_{24}) with the (W_2, W_7, W_9) guessed above. For every remaining plaintext pair $(P_{i,j_1}, \widetilde{P}_{i,j_2})$, partially encrypt the corresponding values for bytes (1,2) just after Round 2 with $(KS_9, KS_{14}, KS_{19}, KS_{24})$ to get the corresponding values for bytes (1,8) just after Round 6. Check whether they have a difference $(0, e_8)$. If none of the plaintext pairs meet this condition, record the guessed value for $(W_1, \cdots, W_{12}, W_{14}, W_{15}, W_{16})$, and execute Step 6; otherwise, discard this guess, and try another.

6. For a recorded value for $(W_1, \cdots, W_{12}, W_{14}, W_{15}, W_{16})$, exhaustively search for the remaining 8 key bits using three known pairs of plaintexts and ciphertexts. If a 128-bit key is suggested, output it as the user key of the 28-round HIGHT; otherwise, go to Step 2.

10.7.2.2 Complexity Analysis

The attack requires 2^{60} (related-key) chosen plaintexts, which take a time complexity of 2^{60} 28-round HIGHT encryptions.

In Step 1, a structure S_i yields $\binom{2^{40}}{2} \approx 2^{79}$ plaintext pairs (P_{i,j_1}, P_{i,j_2}) with difference $(0, 0, e_8, \overline{e}_{1,\sim}, \star, \star, \star, \star)$, thus the 2^{19} structures yield a total of 2^{98} plaintext pairs (P_{i,j_1}, P_{i,j_2}) with difference $(0, 0, e_8, \overline{e}_{1,\sim}, \star, \star, \star, \star)$, $(1 \leq j_1 \neq j_2 \leq 2^{40})$. There is a 25-bit filtering condition over the candidate ciphertext pairs, so it follows that about $2^{98} \times 2^{-25} = 2^{73}$ ciphertext pairs $(C_{i,j_1}, \widetilde{C}_{i,j_2})$ remain after Step 1.

In Step 2(a) there is an 8-bit filtering condition over the candidate ciphertext pairs, so it follows that about $2^{73} \times 2^{-8} = 2^{65}$ ciphertext pairs $(C_{i,j_1}, \widetilde{C}_{i,j_2})$ pass Step 2(a) for every guess of (W_4, W_5). Step 2(a) has a time complexity of $2 \times 2^{73} \times 2^{16} \times \frac{1}{4} \times \frac{1}{28} \approx 2^{83.2}$ 28-round HIGHT encryptions.

In Step 2(b) there is an 8-bit filtering condition over the candidate ciphertext pairs, so it follows that about $2^{65} \times 2^{-8} = 2^{57}$ ciphertext pairs $(C_{i,j_1}, \widetilde{C}_{i,j_2})$ pass Step 2(b) for every guess of (W_3, W_4, W_5, W_9). Step 2(b) has a time complexity of $2 \times 2^{65} \times 2^{32} \times \frac{1}{4} \times \frac{2}{28} \approx 2^{92.2}$ 28-round HIGHT encryptions.

In Step 2(c) there is an 8-bit filtering condition over the candidate ciphertext pairs, so it follows that about $2^{57} \times 2^{-8} = 2^{49}$ ciphertext pairs $(C_{i,j_1}, \widetilde{C}_{i,j_2})$ pass Step 2(c) for every guess of $(W_2, W_3, W_4, W_5, W_9, W_{12}, W_{16})$. Step 2(c) has a time complexity of $2 \times 2^{57} \times 2^{56} \times \frac{1}{4} \times \frac{3}{28} \approx 2^{108.78}$ 28-round HIGHT encryptions.

In Step 3(a) there is an 8-bit filtering condition over the candidate plaintext pairs, so it follows that about $2^{49} \times 2^{-8} = 2^{41}$ plaintext pairs $(P_{i,j_1}, \widetilde{P}_{i,j_2})$ pass Step 3(a) for every guess of $(W_2, W_3, W_4, W_5, W_8, W_9, W_{12}, W_{16})$. Step 3(a) has a time complexity of $2 \times 2^{49} \times 2^{64} \times \frac{1}{4} \times \frac{1}{28} \approx 2^{107.2}$ 28-round HIGHT encryptions.

In Step 3(b) there is an 8-bit filtering condition over the candidate plaintext pairs, so it follows that about $2^{41} \times 2^{-8} = 2^{33}$ plaintext pairs $(P_{i,j_1}, \widetilde{P}_{i,j_2})$ pass Step 3(b) for every guess of $(W_2, W_3, W_4, W_5, W_7, W_8, W_9, W_{12}, W_{16})$. Step 3(b) has a time complexity of $2 \times 2^{41} \times 2^{72} \times \frac{1}{4} \times \frac{2}{28} \approx 2^{108.2}$ 28-round HIGHT encryptions.

In Step 3(c) there is an 8-bit filtering condition over the candidate plaintext pairs, so it follows that about $2^{33} \times 2^{-8} = 2^{25}$ plaintext pairs $(P_{i,j_1}, \widetilde{P}_{i,j_2})$ pass Step 3(c) for every guess of $(W_2, \cdots, W_9, W_{11}, W_{12}, W_{16})$. Step 3(c) has a time complexity of $2 \times 2^{33} \times 2^{88} \times \frac{1}{4} \times \frac{1}{2} \times \frac{3}{28} \approx 2^{115.78}$ 28-round HIGHT encryptions, where $\frac{1}{2}$ means the average fraction of the guessed keys that are tested.

In Step 4 there is an 8-bit filtering condition over the candidate ciphertext pairs, so it follows that about $2^{25} \times 2^{-8} = 2^{17}$ ciphertext pairs $(C_{i,j_1}, \widetilde{C}_{i,j_2})$ pass Step 4 for every guess of $(W_1, \cdots, W_9, W_{11}, W_{12}, W_{15}, W_{16})$. Step 4 has a time complexity of $2 \times 2^{25} \times 2^{104} \times \frac{1}{4} \times \frac{1}{2} \times \frac{4}{28} \approx 2^{124.2}$ 28-round HIGHT encryptions.

In Step 5(a) there is an 8-bit filtering condition over the candidate plaintext pairs, so it follows that about $2^{17} \times 2^{-8} = 2^9$ plaintext pairs $(P_{i,j_1}, \widetilde{P}_{i,j_2})$ pass Step 5(a) for every guess of $(W_1, \cdots, W_{12}, W_{15}, W_{16})$. Step 5(a) has a time complexity of $2 \times 2^{17} \times 2^{112} \times \frac{1}{4} \times \frac{1}{2} \times \frac{4}{28} \approx 2^{124.2}$ 28-round HIGHT encryptions.

In Step 5(b) there is a 7-bit filtering condition over the candidate plaintext pairs, so it is expected that about $2^{120} \times (1 - 2^{-7})^{2^9} \approx 2^{120} \times \mathtt{e}^{-2^2} \approx 2^{114.24}$ guesses for $(W_1, \cdots, W_{12}, W_{14}, W_{15}, W_{16})$ are recorded in Step 5(b). Thus, the expected number of suggested wrong keys in Step 6 is about $2^{114.24} \times 2^8 \times 2^{-192} = 2^{-69.76}$. Thus, it is very likely that we can find the correct key guess. Step 5(b) has a time complexity of about $2 \times 2^{120} \times [1 + (1 - 2^{-7}) + \cdots + (1 - 2^{-7})^{2^9}] \times \frac{1}{2} \times \frac{1}{4} \times \frac{4}{28} \approx 2^{122.19}$ 28-round HIGHT encryptions. Step 6 has about $2^{122.24}$ 28-round HIGHT encryptions.

Therefore, the attack has a total time complexity of about $2^{125.54}$ 28-round HIGHT encryptions.

10.8 Summary

In this chapter we have presented an impossible differential attack on 25-round HIGHT, a related-key rectangle attack on 26-round HIGHT, and a related-key impossible differential attack on 28-round HIGHT. These attacks are better than any previously published cryptanalytic results on HIGHT in terms of the number of attacked rounds. Table 10.5 summarises the published cryptanalytic results on HIGHT, where CP, KP and RK-CP refer to the required numbers of chosen plaintexts, known plaintexts and related-key chosen plaintexts, respectively; and Encryptions refers to the required number of encryption operations of the appropriate reduced version of HIGHT.

Table 10.5: Cryptanalytic results on HIGHT

Attack Type	Rounds	Data	Time	Source
Differential	13	2^{62}CP	not specified	[37]
Linear	13	2^{57}KP	not specified	
Boomerang	13	2^{62}CP	not specified	
Truncated differential	16	$2^{14.1}$CP	$2^{108.69}$Encryptions	
Saturation	16	2^{42}CP	2^{51}Encryptions	
Related-key boomerang	19	not specified	not specified	
Impossible differential	18	$2^{46.8}$CP	$2^{109.2}$Encryptions	
	25	2^{60}CP	$2^{126.78}$Encryptions	Section 10.5
Related-key rectangle	26	$2^{51.2}$RK-CP	$2^{120.41}$Encryptions	Section 10.6
Related-key impossible differential	28	2^{60}RK-CP	$2^{125.54}$Encryptions	Section 10.7

Chapter 11

Conclusions and Future Research

In this chapter we summarise the cryptanalytic results presented in the thesis and give some possible directions for future research.

Contents

11.1 Conclusions

In this thesis we propose a new extension of differential cryptanalysis, which we call the impossible boomerang attack. We describe the early abort technique for (related-key) impossible differential cryptanalysis and rectangle attacks. Finally, we analyse the security of a number of block ciphers that are currently being widely used or have been recently proposed for use in emerging cryptographic applications. The main cryptanalytic results are as follows.

- We give an impossible differential attack on 7-round AES when used with 128 or 192 key bits, and an impossible differential attack on 8-round AES when used with 256 key bits. We also present an impossible boomerang attack on 6-round AES when used with 128 key bits, and an impossible boomerang attack on 7-round AES when used with 192 or 256 key bits. Finally, we describe a related-key impossible boomerang attack on 8-round AES when used with 192 key bits, and a related-key impossible boomerang attack on 9-round AES when used with 256 key bits, both using two keys.

- We give an impossible differential attack on 11-round reduced Camellia when used with 128 key bits, an impossible differential attack on 12-round reduced Camellia when used with 192 key bits, and an impossible differential attack on 13-round reduced Camellia when used with 256 key bits.
- We give a related-key rectangle attack on the full Cobra-F64a, and a related-key differential attack on the full Cobra-F64b.
- We give a related-key rectangle attack on 44-round SHACAL-2.
- We give a related-key rectangle attack on 36-round XTEA.
- We give an impossible differential attack on 25-round reduced HIGHT, a related-key rectangle attack on 26-round reduced HIGHT, and a related-key impossible differential attack on 28-round reduced HIGHT.

In terms of either the attack complexity or the numbers of attacked rounds, the attacks presented in the thesis are better than any previously published cryptanalytic results for the block ciphers concerned, except in the case of AES. For AES, the impossible differential attacks on 7-round AES used with 128 key bits and 8-round AES used with 256 key bits are the best currently published results on AES in a single key attack scenario, and the presented related-key impossible boomerang attacks on 8-round AES used with 192 key bits and 9-round AES used with 256 key bits are the best currently published results on AES in a related-key attack scenario using two keys.

11.2 Possible Directions for Future Research

We give some possible directions for future research on block cipher cryptanalysis, and it would be interesting to investigate these directions.

- As mentioned in Chapter 3, the impossible boomerang attack can potentially be used to cryptanalyse other block ciphers, in particular analysing those with a simple key schedule in a related-key attack scenario. For example, IDEA [66] is a widely used block cipher with a linear key schedule, thus it is desirable to check whether some good cryptanalytic results can be obtained when we apply the impossible boomerang attack to IDEA. Actually, this is part of my ongoing work.

- Other than those described in Chapter 4, there exist a variety of other examples of the application of the early abort technique to (related-key) impossible differential and rectangle attacks, as well as other cryptanalytic methods, depending on the specific design of the round function of a block cipher; for instance, the way we exploit Properties 9.3 and 10.2 when conducting the attacks on XTEA and HIGHT. As a result, it may be possible to improve certain existing cryptanalytic results for block ciphers using the early abort technique.
- Cryptology is a very fast moving field. It is possible to improve the results presented in this thesis.

Bibliography

[1] Kazumaro Aoki, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, and Toshio Tokita. Camellia: a 128-bit block cipher suitable for multiple platforms — design and analysis. In D.R. Stinson and S.E. Tavares, editors, *Proceedings of SAC '00 — The 7th Annual Workshop on Selected Areas in Cryptography*, volume 2012 of *Lecture Notes in Computer Science*, pages 39–56. Springer-Verlag, 2001.

[2] Behnam Bahrak and Mohammad Reza Aref. A novel impossible differenital cryptanalysis of AES, In *Proceedings of WEWoRc '07 — Western European Workshop on Research in Cryptology.* 2007.

[3] Eli Biham. New types of cryptanalytic attacks using related keys. In T. Helleseth, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '93 — Workshop on the Theory and Application of Cryptographic Techniques*, volume 765 of *Lecture Notes in Computer Science*, pages 398–409. Springer-Verlag, 1993.

[4] Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials. In J. Stern, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '99 — International Conference on the Theory and Application of Cryptographic Techniques*, volume 1592 of *Lecture Notes in Computer Science*, pages 12–23. Springer-Verlag, 1999.

[5] Eli Biham, Alex Biryukov, and Adi Shamir. Miss in the middle attacks on IDEA and Khufu. In L.R. Knudsen, editor, *Proceedings of FSE '99 — The 6th International Workshop on Fast Software Encryption*, volume 1636 of *Lecture Notes in Computer Science*, pages 124–138. Springer-Verlag, 1999.

[6] Eli Biham, Orr Dunkelman, and Nathan Keller. The rectangle attack — rectangling the Serpent. In B. Pfitzmann, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '01 — International Conference on the Theory and Application of Cryptographic Techniques*, volume 2045 of *Lecture Notes in Computer Science*, pages 340–357. Springer-Verlag, 2001.

[7] Eli Biham, Orr Dunkelman, and Nathan Keller. Enhancing differential-linear cryptanalysis. In Y. Zheng, editor, *Advances in Cryptology - Proceedings of ASIACRYPT '02 — The 8th International Conference on the Theory and Application of Cryptology and Information Security*, volume 2501 of *Lecture Notes in Computer Science*, pages 254–266. Springer-Verlag, 2002.

[8] Eli Biham, Orr Dunkelman, and Nathan Keller. New results on boomerang and rectangle attacks. In J. Daemen and V. Rijmen, editors, *Proceedings of FSE '02 — The 9th International Workshop on Fast Software Encryption*, volume 2365 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, 2002.

[9] Eli Biham, Orr Dunkelman, and Nathan Keller. Related-key boomerang and rectangle attacks. In R. Cramer, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '05 — The 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, volume 3494 of *Lecture Notes in Computer Science*, pages 507–525. Springer-Verlag, 2005.

[10] Eli Biham, Orr Dunkelman, and Nathan Keller. Related-key impossible differential attacks on 8-round AES-192. In D. Pointcheval, editor, *Proceedings of CT-RSA '06 — Cryptographers' Track at the RSA Conference 2006*, volume 3860 of *Lecture Notes in Computer Science*, pages 21–33. Springer-Verlag, 2006.

[11] Eli Biham and Nathan Keller. Cryptanalysis of reduced variants of Rijndael. In *Proceedings of The Third Advanced Encryption Standard Candidate Conference*. NIST, 2000.

[12] Eli Biham and Adi Shamir. Differential cryptanalysis of DES-like cryptosystems. In A. Menezes and S.A. Vanstone, editors, *Advances in Cryptology - Proceedings of CRYPTO '90 — The 10th Annual International Cryptology Conference*, volume 537 of *Lecture Notes in Computer Science*, pages 2–21. Springer-Verlag, 1990.

[13] Eli Biham and Adi Shamir. Differential cryptanalysis of the Data Encryption Standard. Springer-Verlag, 1993.

[14] Eli Biham and Adi Shamir. Differential cryptanalysis of the full 16-round DES. In E.F. Brickell, editor, *Advances in Cryptology - Proceedings of CRYPTO '92 — The 12th Annual International Cryptology Conference*, volume 740 of *Lecture Notes in Computer Science*, pages 487–496. Springer-Verlag, 1993.

[15] Alex Biryukov. The boomerang attack on 5 and 6-round reduced AES. In H. Dobbertin, V. Rijmen, and A. Sowa, editors, *Proceedings of AES '04 — The 4th International Conference on Advanced Encryption Standard*, volume 3373 of *Lecture Notes in Computer Science*, pages 11–15. Springer-Verlag, 2005.

[16] Jung Hee Cheon, MunJu Kim, Kwangjo Kim, Jung-Yeun Lee, and SungWoo Kang. Improved impossible differential cryptanalysis of Rijndael and Crypton. In K. Kim, editor, *Proceedings of ICISC '01 — The 4th International Conference on Information Security and Cryptology*, volume 2288 of *Lecture Notes in Computer Science*, pages 39–49. Springer-Verlag, 2001.

[17] Nicolas Courtois. Feistel schemes and bi-linear cryptanalysis. In M.K. Franklin, editor, *Advances in Cryptology - Proceedings of CRYPTO '04 — The 24th Annual International Cryptology Conference*, volume 3152 of *Lecture Notes in Computer Science*, pages 23–40. Springer-Verlag, 2004.

[18] Nicolas Courtois and Josef Pieprzyk. Cryptanalysis of block ciphers with overdelned systems of equations. In Y. Zheng, editor, *Advances in Cryptology - Proceedings of ASIACRYPT '02 — The 8th International Conference on the Theory and Application of Cryptology and Information Security*, volume 2501 of *Lecture Notes in Computer Science*, pages 267–287. Springer-Verlag, 2002.

[19] CRYPTREC — Cryptography Research and Evaluatin Committees, report 2002. *http://www.ipa.go.jp/security/enc/CRYPTREC/index-e.html.*

[20] Joan Daemen, Lars R. Knudsen, and Vincent Rijmen. The block cipher Square. In E. Biham, editor, *Proceedings of FSE '97 — The 4th International Workshop on Fast Software Encryption*, volume 1267 of *Lecture Notes in Computer Science*, pages 149–165. Springer-Verlag, 1997.

[21] Joan Daemen and Vincent Rijmen. AES proposal: Rijndael. In *Proceedings of The First Advanced Encryption Standard Candidate Conference*. NIST, 1998.

[22] Huseyin Demirci and Ali Aydin Selcuk. A meet-in-the-middle attack on 8-round AES. In K. Nyberg, editor, *Proceedings of FSE '08 — The 15th International Workshop on Fast Software Encryption*, volume ? of *Lecture Notes in Computer Science*, pages ?–? Springer-Verlag, 2008.

[23] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654, 19767.

[24] Lei Duo, Chao Li, and Keqin Feng. New observation on Camellia. In B. Preneel and S.E. Tavares, editors, *Proceedings of SAC '05 — The 12th Annual Workshop on Selected Areas in Cryptography*, volume 3897 of *Lecture Notes in Computer Science*, pages 51–64. Springer-Verlag, 2006.

[25] Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner, and Doug Whiting. Improved cryptanalysis of Rijndael. In B. Schneier, editor, *Proceedings of FSE '00 — The 7th International Workshop on Fast Software Encryption*, volume 1978 of *Lecture Notes in Computer Science*, pages 213–230. Springer-Verlag, 2001.

[26] Henri Gilbert and Marine Minier. A collision attack on 7 rounds of Rijndael. In *Proceedings of The Third Advanced Encryption Standard Candidate Conference*. NIST, 2000.

[27] Nick D. Goots, Boris V. Izotov, Alexander A. Moldovyan, and Nick A. Moldovyan. Fast ciphers for cheap hardware: differential analysis of SPECTR-H64. In Vladimir Gorodetsky, Igor V. Kotenko, and Victor A. Skormin, editors, *Proceedings of MMM-ACNS '03 — The Second International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security*, volume 2776 of *Lecture Notes in Computer Science*, pages 449–452. Springer-Verlag, 2003.

[28] Nick D. Goots, Boris V. Izotov, Alexander A. Moldovyan, and Nick A. Moldovyan. Modern cryptography: protect your data with fast block ciphers. A-LIST Publishing, 2003.

[29] Nick D. Goots, Alexander A. Moldovyan, and Nick A. Moldovyan. Fast encryption algorithm SPECTR-H64. In Vladimir I. Gorodetski, Victor A. Skormin,

and Leonard J. Popyack, editors, *Proceedings of MMM-ACNS '01 — International Workshop on Information Assurance in Computer Networks: Methods, Models, and Architectures for Network Security*, volume 2052 of *Lecture Notes in Computer Science*, pages 275–286. Springer-Verlag, 2001.

[30] Nick D. Goots, Alexander A. Moldovyan, Nick A. Moldovyan, and D.H. Summerville. Fast DDP-based ciphers: from hardware to software. In *Proceedings of the 46th IEEE Midwest International Symposium on Circuits and Systems*, pages 770–773, 2003.

[31] Helena Handschuh and David Naccache. SHACAL. In *Proceedings of The First Open NESSIE Workshop*, 2000. Archive available at *https://www.cosic.esat.kuleuven.be/nessie/workshop/submissions.html.*

[32] Helena Handschuh and David Naccache. SHACAL. NESSIE, 2001. Archive available at *https://www.cosic.esat.kuleuven.be/nessie/tweaks.html.*

[33] Yasuo Hatano, Hiroki Sekine, and Toshinobu Kaneko. Higher order differential attack of Camellia(II). In K. Nyberg and H.M. Heys, editors, *Proceedings of SAC '02 — The 9th Annual Workshop on Selected Areas in Cryptography*, volume 2595 of *Lecture Notes in Computer Science*, pages 39–56. Springer-Verlag, 2003.

[34] Philip Hawkes. Differential-linear weak key classes of IDEA. In K. Nyberg, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '98 — International Conference on the Theory and Application of Cryptographic Techniques*, volume 1403 of *Lecture Notes in Computer Science*, pages 112–126. Springer-Verlag, 1998.

[35] Yeping He and Sihan Qing. Square attack on reduced Camellia cipher. In S. Qing, T. Okamoto, and J. Zhou, editors, *Proceedings of ICICS '01 — The Third International Conference on Information and Communications Security*, volume 2229 of *Lecture Notes in Computer Science*, pages 238–245. Springer-Verlag, 2001.

[36] Martin E. Hellman. A cryptanalytic time-memory trade-off. *IEEE Transactions on Information Theory*, IT-26(4):401–406, 1980. IEEE Press.

[37] Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bon-Seok Koo, Changhoon Lee, Donghoon Chang, Jesang Lee, Kitae Jeong, Hyun Kim,

Jongsung Kim, and Seongtaek Chee. HIGHT: a new block cipher suitable for low-resource device. In L. Goubin and M. Matsui, editors, *Proceedings of CHES '06 — The 8th International Workshop on Cryptographic Hardware and Embedded Systems*, volume 4249 of *Lecture Notes in Computer Science*, pages 46–59. Springer-Verlag, 2006.

[38] Seokhie Hong, Deukjo Hong, Youngdai Ko, Donghoon Chang, Wonil Lee, and Sangjin Lee. Differential cryptanalysis of TEA and XTEA. In J. Lim and D. Lee, editors, *Proceedings of ICISC '03 — The 6th International Conference on Information Security and Cryptology*, volume 2791 of *Lecture Notes in Computer Science*, pages 402–417. Springer-Verlag, 2003.

[39] Seokhie Hong, Jongsung Kim, Guil Kim, Jaechul Sung, Changhoon Lee, , and Sangjin Lee. Impossible differential attack on 30-round SHACAL-2. In T. Johansson and S. Maitra, editors, *Proceedings of INDOCRYPT '03 — The 4th International Conference on Cryptology in India*, volume 2904 of *Lecture Notes in Computer Science*, pages 97–106. Springer-Verlag, 2003.

[40] Seokhie Hong, Jongsung Kim, Sangjin Lee, , and Bart Preneel. Related-key rectangle attacks on reduced versions of SHACAL-1 and AES-192. In H. Gilbert and H. Handschuh, editors, *Proceedings of FSE '05 — The 12th International Workshop on Fast Software Encryption*, volume 3557 of *Lecture Notes in Computer Science*, pages 368–383. Springer-Verlag, 2005.

[41] The Institute of Electrical and Electronics Engineers (IEEE). *http://grouper.ieee.org/groups/802/11*.

[42] International Standardization of Organization (ISO), International Standard – ISO/IEC 18033-3, Information technology – Security techniques – Encryption algorithms – Part 3: Block ciphers, July, 2005.

[43] International Standardization of Organization (ISO), International Standard– ISO/IEC 8802-11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. *http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=39777*.

[44] The Internet Engineering Task Force (IETF), RFC 4301 – Security Architecture for the Internet Protocol, December, 2005.

[45] Goce Jakimoski and Yvo Desmedt. Related-key differential cryptanalysis of 192-bit key AES variants. In M. Matsui and R.J. Zuccherato, editors, *Proceedings of SAC '03 — The 10th Annual Workshop on Selected Areas in Cryptography*, volume 3006 of *Lecture Notes in Computer Science*, pages 208–221. Springer-Verlag, 2004.

[46] Burton S. Kaliski Jr. and Matthew J.B. Robshaw. Linear cryptanalysis using multiple approximations. In Y. Desmedt, editor, *Advances in Cryptology - Proceedings of CRYPTO '94 — The 14th Annual International Cryptology Conference*, volume 839 of *Lecture Notes in Computer Science*, pages 26–39. Springer-Verlag, 1994.

[47] Ari Juels. RFID security and privacy: a research survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, 2006.

[48] John Kelsey, Tadayoshi Kohno, and Bruce Schneier. Amplified boomerang attacks against reduced-round MARS and Serpent. In B. Schneier, editor, *Proceedings of FSE '00 — The 7th International Workshop on Fast Software Encryption*, volume 1978 of *Lecture Notes in Computer Science*, pages 75–93. Springer-Verlag, 2001.

[49] John Kelsey, Bruce Schneier, and David Wagner. Key-schedule cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES. In N. Koblitz, editor, *Advances in Cryptology - Proceedings of CRYPTO '96 — The 16th Annual International Cryptology Conference*, volume 1109 of *Lecture Notes in Computer Science*, pages 237–251. Springer-Verlag, 1996.

[50] John Kelsey, Bruce Schneier, and David Wagner. Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In Y. Han, T. Okamoto, and S. Qing, editors, *Proceedings of ICICS '97 — The First International Conference on Information and Communications Security*, volume 1334 of *Lecture Notes in Computer Science*, pages 233–246. Springer-Verlag, 1997.

[51] Auguste Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, IX:5–83, 1883.

[52] Jongsung Kim, Seokhie Hong, and Bart Preneel. Related-key rectangle attacks on reduced AES-192 and AES-256. In A. Biryukov, editor, *Proceedings of FSE*

'07 — The 14th International Workshop on Fast Software Encryption, volume 4593 of *Lecture Notes in Computer Science*, pages 225–241. Springer-Verlag, 2007.

[53] Jongsung Kim, Guil Kim, Seokhie Hong, Sangjin Lee, and Dowon Hong. The related-key rectangle attack — application to SHACAL-1. In H. Wang, J. Pieprzyk, and V. Varadharajan, editors, *Proceedings of ACISP '04 — The 9th Australasian Conference on Information Security and Privacy*, volume 3108 of *Lecture Notes in Computer Science*, pages 123–136. Springer-Verlag, 2004.

[54] Jongsung Kim, Guil Kim, Sangjin Lee, Jongin Lim, and Junghwan Song. Related-key attacks on reduced rounds of SHACAL-2. In A. Canteaut and K. Viswanathan, editors, *Proceedings of INDOCRYPT '04 — The 5th International Conference on Cryptology in India*, volume 3348 of *Lecture Notes in Computer Science*, pages 175–190. Springer-Verlag, 2004.

[55] Lars R. Knudsen. Cryptanalysis of LOKI91. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology - Proceedings of ASIACRYPT '92 — Workshop on the Theory and Application of Cryptographic Techniques*, volume 718 of *Lecture Notes in Computer Science*, pages 196–208. Springer-Verlag, 1993.

[56] Lars R. Knudsen. Trucated and higher order differentials. In B. Preneel, editor, *Proceedings of FSE '94 — The Second International Workshop on Fast Software Encryption*, volume 1008 of *Lecture Notes in Computer Science*, pages 196–211. Springer-Verlag, 1995.

[57] Lars R. Knudsen. *DEAL — a 128-bit block cipher.* Technical report, Department of Informatics, University of Bergen, Norway, 1998.

[58] Lars R. Knudsen and John E. Mathiassen. A chosen-plaintext linear attack on DES. In B. Schneier, editor, *Proceedings of FSE '00 — The 7th International Workshop on Fast Software Encryption*, volume 1978 of *Lecture Notes in Computer Science*, pages 262–272. Springer-Verlag, 2001.

[59] Lars R. Knudsen and Matthew J.B. Robshaw. Non-linear approximations in linear cryptoanalysis. In U.M. Maurer, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '96 — International Conference on the Theory*

and Application of Cryptographic Techniques, volume 1070 of *Lecture Notes in Computer Science*, pages 224–236. Springer-Verlag, 1996.

[60] Lars R. Knudsen and David Wagner. Integral cryptanalysis. In J. Daemen and V. Rijmen, editors, *Proceedings of FSE '02 — The 9th International Workshop on Fast Software Encryption*, volume 2365 of *Lecture Notes in Computer Science*, pages 112–127. Springer-Verlag, 2002.

[61] Youngdai Ko, Seokhie Hong, Wonil Lee, Sangjin Lee, and Ju-Sung Kang. Related key differential attacks on 27 rounds of XTEA and full-round GOST. In B. Roy and W. Meier, editors, *Proceedings of FSE '04 — The 11th International Workshop on Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 299–316. Springer-Verlag, 2004.

[62] Youngdai Ko, Changhoon Lee, Seokhie Hong, and Sangjin Lee. Related key differential cryptanalysis of full-round SPECTR-H64 and CIKS-1. In H. Wang, J. Pieprzyk, and V. Varadharajan, editors, *Proceedings of ACISP '04 — The 9th Australasian Conference on Information Security and Privacy*, volume 3108 of *Lecture Notes in Computer Science*, pages 137–148. Springer-Verlag, 2004.

[63] Youngdai Ko, Changhoon Lee, Seokhie Hong, Jaechul Sung, and Sangjin Lee. Related-key attacks on ddp based ciphers: CIKS-128 and CIKS-128H. In A. Canteaut and K. Viswanathan, editors, *Proceedings of INDOCRYPT '04 — The 5th International Conference on Cryptology in India*, volume 3348 of *Lecture Notes in Computer Science*, pages 191–205. Springer-Verlag, 2004.

[64] Ulrich Kühn. Cryptanalysis of reduced-round MISTY. In B. Pfitzmann, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '01 — International Conference on the Theory and Application of Cryptographic Techniques*, volume 2045 of *Lecture Notes in Computer Science*, pages 325–339. Springer-Verlag, 2001.

[65] Xuejia Lai. Higher order derivatives and differential cryptanalysis. *Communications and Cryptography*, pages 227–233, 1994. Kluwer Academic Publishers.

[66] Xuejia Lai and James L. Massey. A proposal for a new block encryption standard. In I. Damgard, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '90 — Workshop on the Theory and Application of Cryptographic*

Techniques, volume 473 of *Lecture Notes in Computer Science*, pages 389–404. Springer-Verlag, 1991.

[67] Suzan K. Langford and Martin E. Hellman. Differential-linear cryptanalysis. In Y. Desmedt, editor, *Advances in Cryptology - Proceedings of CRYPTO '94 — The 14th Annual International Cryptology Conference*, volume 839 of *Lecture Notes in Computer Science*, pages 17–25. Springer-Verlag, 1994.

[68] Changhoon Lee, Jongsung Kim, Seokhie Hong, Jaechul Sung, and Sangjin Lee. Related-key differential attacks on Cobra-S128, Cobra-F64a and Cobra-F64b. In E. Dawson and S. Vaudenay, editors, *Proceedings of Mycrypt '05 — The First International Conference on Cryptology in Malaysia*, volume 3715 of *Lecture Notes in Computer Science*, pages 244–262. Springer-Verlag, 2005.

[69] Changhoon Lee, Jongsung Kim, Jaechul Sung, Seokhie Hong, Sangjin Lee, and Dukjae Moon. Related-key differential attacks on Cobra-H64 and Cobra-H128. In N.P. Smart, editor, *Proceedings of IMA Cryptography and Coding '05 — The 10th IMA International Conference on Cryptography and Coding*, volume 3796 of *Lecture Notes in Computer Science*, pages 201–219. Springer-Verlag, 2005.

[70] Eunjin Lee, Deukjo Hong, Donghoon Chang, Seokhie Hong, and Jongin Lim. A weak key class of XTEA for a related-key rectangle attack. In P.Q. Nguyen, editor, *Proceedings of Vietcrypt '06 — The First International Conferenceon Cryptology in Vietnam*, volume 4341 of *Lecture Notes in Computer Science*, pages 286–297. Springer-Verlag, 2006.

[71] Seonhee Lee, Seokhie Hong, Sangjin Lee, Jongin Lim, and Seonhee Yoon. Truncated differential cryptanalysis of Camellia. In K. Kim, editor, *Proceedings of ICISC '01 — The 4th International Conference on Information Security and Cryptology*, volume 2288 of *Lecture Notes in Computer Science*, pages 32–38. Springer-Verlag, 2002.

[72] B.W. Lindgren and G.W. Mcelrath. *Introduction to PROBABILITY and STATISTICS — third edition.* The Macmillan Company, 1969.

[73] Helger Lipmaa and Shiho Moriai. Efficient algorithms for computing differential properties of addition. In M. Matsui, editor, *Proceedings of FSE '01 —*

The 8th International Workshop on Fast Software Encryption, volume 2355 of *Lecture Notes in Computer Science*, pages 336–350. Springer-Verlag, 2001.

[74] Jiqiang Lu. Cryptanalysis of reduced versions of the HIGHT block cipher from CHES 2006. In K. Nam and G. Rhee, editors, *Proceedings of ICISC '07 — The 10th International Conference on Information Security and Cryptology*, volume 4817 of *Lecture Notes in Computer Science*, pages 11–26. Springer-Verlag, 2007.

[75] Jiqiang Lu. Related-key rectangle attack on 36 rounds of the XTEA block cipher. *International Journal of Information Security*, ?:?–?, 2008.

[76] Jiqiang Lu and Jongsung Kim. Attacking 44 rounds of the SHACAL-2 block cipher using related-key rectangle cryptanalysis. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 91-A:?–?, 2008.

[77] Jiqiang Lu, Jongsung Kim, Nathan Keller, and Orr Dunkelman. Differential and rectangle attacks on reduced-round SHACAL-1. In R. Barua and T. Lange, editors, *Progress in Cryptology - INDOCRYPT '06 — The 7th International Conference on Cryptology in India*, volume 4329 of *Lecture Notes in Computer Science*, pages 17–31. Springer-Verlag, 2006.

[78] Jiqiang Lu, Jongsung Kim, Nathan Keller, and Orr Dunkelman. Related-key rectangle attack on 42-round SHACAL-2. In S.K. Katsikas, J. Lopez, M. Backes, and B. Preneel, editors, *Proceedings of ISC '06 — The 9th International Conference on Information Security*, volume 4176 of *Lecture Notes in Computer Science*, pages 85–100. Springer-Verlag, 2006.

[79] Jiqiang Lu, Jongsung Kim, Nathan Keller, and Orr Dunkelman. Improving the efficiency of impossible differential cryptanalysis of reduced Camellia and MISTY1. In T. Malkin, editor, *Proceedings of CT-RSA '08 — Cryptographers' Track at the RSA Conference 2008*, volume 4964 of *Lecture Notes in Computer Science*, pages 370–386. Springer-Verlag, 2008.

[80] Jiqiang Lu, Changhoon Lee, and Jongsung Kim. Related-key attacks on the full-round Cobra-F64a and Cobra-F64b. In R.D. Prisco and M. Yung, editors, *Proceedings of SCN '06 — The Fifth International Conference on Security*

and Cryptography for Networks, volume 4116 of *Lecture Notes in Computer Science*, pages 95–110. Springer-Verlag, 2006.

[81] Stefan Lucks. Attacking seven rounds of Rijndael under 192-bit and 256-bit keys. In *Proceedings of The Third Advanced Encryption Standard Candidate Conference*. NIST, 2000.

[82] Stefan Lucks. The saturation attack — a bait for Twofish. In M. Matsui, editor, *Proceedings of FSE '01 — The 8th International Workshop on Fast Software Encryption*, volume 2355 of *Lecture Notes in Computer Science*, pages 1–15. Springer-Verlag, 2002.

[83] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In T. Helleseth, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '93 — Workshop on the Theory and Application of Cryptographic Techniques*, volume 765 of *Lecture Notes in Computer Science*, pages 386–397. Springer-Verlag, 1994.

[84] Mitsuru Matsui and Atsuhiro Yamagishi. A new method for known plaintext attack of FEAL cipher. In R.A. Rueppel, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '92 — Workshop on the Theory and Application of Cryptographic Techniques*, volume 658 of *Lecture Notes in Computer Science*, pages 81–91. Springer-Verlag, 1993.

[85] Alexander A. Moldovyan and Nick A. Moldovyan. A cipher based on data-dependent permutations. *Journal of Cryptology*, 15(1):61–72, 2002. Springer.

[86] Dukjae Moon, Kyungdeok Hwang, Wonil Lee, Sangjin Lee, and Jongin Lim. Impossible differential cryptanalysis of reduced round XTEA and TEA. In J. Daemen and V. Rijmen, editors, *Proceedings of FSE '02 — The 9th International Workshop on Fast Software Encryption*, volume 2365 of *Lecture Notes in Computer Science*, pages 49–60. Springer-Verlag, 2002.

[87] Sean Murphy. The cryptanalysis of FEAL-4 with 20 chosen plaintexts. *Journal of Cryptology*, 2(3):145–154, 1990. Springer.

[88] Roger M. Needham and David J. Wheeler. *TEA extensions*. Technical report, the Computer Laboratory, University of Cambridge, 1997. Archive available at *http://www.cl.cam.ac.uk/ftp/users/djw3/xtea.ps*.

[89] NESSIE — New European Schemes for Signatures, Integrity, and Encryption, final report. *https://www.cosic.esat.kuleuven.be/nessie/Bookv015.pdf.*

[90] NIST — National Institute of Standards and Technology, Advanced Encryption Standard (AES), FIPS-197, 2001.

[91] NIST — National Institute of Standards and Technology, Data Encryption Standard (DES), FIPS-46, 1977.

[92] NIST — National Institute of Standards and Technology, Secure Hash Standard, FIPS 180-1, 1995.

[93] NIST — National Institute of Standards and Technology, Secure Hash Standard, FIPS 180-2, 2002.

[94] Kaisa Nyberg. Linear approximation of block ciphers. In A.D. Santis, editor, *Advances in Cryptology - Proceedings of EUROCRYPT '94 — Workshop on the Theory and Application of Cryptographic Techniques*, volume 950 of *Lecture Notes in Computer Science*, pages 439–444. Springer-Verlag, 1994.

[95] Kaisa Nyberg and Lars R. Knudsen. Provable security against differential cryptanalysis. In E.F. Brickell, editor, *Advances in Cryptology - Proceedings of CRYPTO '92 — the 12th Annual International Cryptology Conference*, volume 740 of *Lecture Notes in Computer Science*, pages 566–574. Springer-Verlag, 1993.

[96] Raphael C.-W. Phan. Impossible differential cryptanalysis of 7-round Advanced Encryption Standard (AES). *Information Processing Letters*, 91:33–38, 2004. Elsevier Science.

[97] Akihiro Shimizu and Shoji Miyaguchi. Fast data encipherment algorithm FEAL. In D. Chaum and W.L. Price, editors, *Advances in Cryptology - Proceedings of EUROCRYPT '87 — Workshop on the Theory and Application of Cryptographic Techniques*, volume 304 of *Lecture Notes in Computer Science*, pages 267–278. Springer-Verlag, 1988.

[98] Yongsup Shin, Jongsung Kim, Guil Kim, Seokhie Hong, and Sangjin Lee. Differential-linear type attacks on reduced rounds of SHACAL-2. In H. Wang, J. Pieprzyk, and V. Varadharajan, editors, *Proceedings of ACISP '04 — The 9th Australasian Conference on Information Security and Privacy*, volume

3108 of *Lecture Notes in Computer Science*, pages 110–122. Springer-Verlag, 2004.

[99] Taizo Shirai. Differential, linear, boomerang and rectangle cryptanalysis of reduced-round Camellia. In *Proceedings of The Third NESSIE Workshop*, 2002.

[100] Nicolas Sklavos, Nick A. Moldovyan, and Odysseas G. Koufopavlou. A new DDP-based cipher CIKS-128H: architecture, design and VLSI implementation optimization of CBC-encryption and hashing over 1 GBPS. In *Proceedings of The 46th IEEE Midwest International Symposium on Circuits and Systems*, pages 463–466, 2003.

[101] Nicolas Sklavos, Nick A. Moldovyan, and Odysseas G. Koufopavlou. High speed networking security: design and implementation of two new DDP-based ciphers. *Mobile Networks and Applications*, 10(1–2):219–231, 2005. Kluwer Academic Publishers.

[102] Makoto Sugita, Kazukuni Kobara, and Hideki Imai. Security of reduced version of the block cipher Camellia against truncated and impossible differential cryptanalysis. In C. Boyd, editor, *Advances in Cryptology - Proceedings of ASIACRYPT '01 — The 7th International Conference on the Theory and Application of Cryptology and Information Security*, volume 2248 of *Lecture Notes in Computer Science*, pages 193–207. Springer-Verlag, 2001.

[103] David Wagner. The boomerang attack. In L.R. Knudsen, editor, *Proceedings of FSE '99 — The 6th International Workshop on Fast Software Encryption*, volume 1636 of *Lecture Notes in Computer Science*, pages 156–170. Springer-Verlag, 1999.

[104] Gaoli Wang. Related-key rectangle attack on 43-round SHACAL-2. In E. Dawson and D.S. Wong, editors, *Proceedings of ISPEC '07 — The Third International Conference on Information Security Practice and Experience*, volume 4464 of *Lecture Notes in Computer Science*, pages 33–42. Springer-Verlag, 2007.

[105] Gaoli Wang, Nathan Keller, and Orr Dunkelman. The delicate issues of addition with respect to XOR differences. In C. Adams, A. Miri, and M. Wiener, editors, *Proceedings of SAC '07 — The 14th Annual Workshop on Selected*

Areas in Cryptography, volume 4876 of *Lecture Notes in Computer Science*, pages 212–231. Springer-Verlag, 2008.

[106] David J. Wheeler and Roger M. Needham. TEA, a tiny encryption algorithm. In B. Preneel, editor, *Proceedings of FSE '94 — The Second International Workshop on Fast Software Encryption*, volume 1008 of *Lecture Notes in Computer Science*, pages 363–366. Springer-Verlag, 1995.

[107] Wenling Wu, Dengguo Feng, and Hua Chen. Collision attack and pseudorandomness of reduced-round Camellia. In H. Handschuh and M.A. Hasan, editors, *Proceedings of SAC '04 — The 11th Annual Workshop on Selected Areas in Cryptography*, volume 3357 of *Lecture Notes in Computer Science*, pages 256–270. Springer-Verlag, 2005.

[108] Wenling Wu, Wentao Zhang, and Dengguo Feng. Impossible differential cryptanalysis of reduced-round ARIA and Camellia. *Journal of Computer Science and Technology*, 22(3):449–456, 2007. Springer.

[109] Yongjin Yeom, Sangwoo Park, and Iljun Kim. On the security of Camellia against the square attack. In J. Daemen and V. Rijmen, editors, *Proceedings of FSE '02 — The 9th International Workshop on Fast Software Encryption*, volume 2356 of *Lecture Notes in Computer Science*, pages 89–99. Springer-Verlag, 2002.

[110] Yongjin Yeom, Sangwoo Park, and Iljun Kim. A study of integral type cryptanalysis on Camellia. In *Proceedings of The 2003 Symposium on Cryptography and Information Security*, pages 453–456, 2003.

[111] Wentao Zhang, Wenling Wu, and Dengguo Feng. New results on impossible differential cryptanalysis of reduced AES. In K.-H. Nam and G. Rhee, editors, *Proceedings of ICISC '07 — The 10th International Conference on Information Security and Cryptology*, volume 4817 of *Lecture Notes in Computer Science*, pages 239–250. Springer-Verlag, 2007.

[112] Wentao Zhang, Lei Zhang, Wenling Wu, and Dengguo Feng. Improved related-key impossible differential attacks on reduced-round AES-192. In E. Biham and A.M. Youssef, editors, *Proceedings of SAC '06 — The 13th Annual Workshop on Selected Areas in Cryptography*, volume 4356 of *Lecture Notes in Computer Science*, pages 15–27. Springer-Verlag, 2007.

[113] Wentao Zhang, Lei Zhang, Wenling Wu, and Dengguo Feng. Related-key differential-linear attacks on reduced AES-192. In K. Srinathan, C. Pandu Rangan, and M. Yung, editors, *Proceedings of INDOCRYPT '07 — The 8th International Conference on Cryptology in India*, volume 4859 of *Lecture Notes in Computer Science*, pages 73–85. Springer-Verlag, 2007.