KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK–ESAT
Kasteelpark Arenberg 10, 3001 Leuven-Heverlee

Combined Differential, Linear and Related-Key Attacks on Block Ciphers and MAC Algorithms

Promotoren:
Prof. Dr. ir. Bart Preneel
Prof. Dr. Sangjin Lee

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Jongsung Kim

November 2006

KATHOLIEKE UNIVERSITEIT LEUVEN
FACULTEIT INGENIEURSWETENSCHAPPEN
DEPARTEMENT ELEKTROTECHNIEK–ESAT
Kasteelpark Arenberg 10, 3001 Leuven-Heverlee

Combined Differential, Linear and Related-Key Attacks on Block Ciphers and MAC Algorithms

Jury:
Prof. Dr. ir. Etienne Aernoudt, voorzitter
Prof. Dr. ir. Bart Preneel, promotor
Prof. Dr. Sangjin Lee, promotor (Korea Univ.)
Prof. Dr. Seokhie Hong (Korea Univ.)
Prof. Dr. Lars R. Knudsen (DTU)
Prof. Dr. ir. Marc Van Barel
Prof. Dr. ir. Joos Vandewalle
Prof. Dr. ir. Patrick Wambacq

Proefschrift voorgedragen tot
het behalen van het doctoraat
in de ingenieurswetenschappen

door

Jongsung Kim

U.D.C. 681.3*D46 November 2006

D/2006/7515/90

ISBN 978-90-5682-756-4

Acknowledgements

It is my pleasure to thank all the people who have helped me to make this Ph.D. thesis.

First of all, I would like to express my gratitude to Prof. Sangjin Lee and Prof. Bart Preneel for being the promotors of this thesis. I thank them for their support and for helping me to develop my cryptography. I also want to thank Prof. Jongin Lim for giving me the opportunity to start studying cryptography.

I am very grateful to Prof. Seokhie Hong, Prof. Lars R. Knudsen, Prof. Marc Van Barel, Prof. Joos Vandewalle and Prof. Patrick Wambacq for serving on the jury and to Prof. Etienne Aernoudt for chairing the jury.

Special thanks go to Prof. Jaechul Sung and Prof. Soohak Sung, for introducing me to cryptanalysis of symmetric-key cryptography. I want to thank Elena Andreeva, Özgül Kücük, Joe Lano, Souradyuti Paul and Hongjun Wu for interesting discussions. I would also like to thank my coauthors: Alex Biryukov, Donghoon Chang, Jaemin Choi, Orr Dunkelman, Deukjo Hong, Seokhie Hong, Kitae Jeong, Nathan Keller, Guil Kim, Changhoon Lee, Jesang Lee, Sangjin Lee, Taekeon Lee, Wonil Lee, Jiqiang Lu, Dukjae Moon, Raphael C.-W. Phan, Bart Preneel and Jaechul Sung.

Thank you to the many CIST and COSIC members for the nice working atmospheres. Jooeun Lee and Péla Noë deserve a big thank for their support with administrative matters.

Last, but definitely not least, I want to thank my parents for their everlasting support and encouragement. I would also like to thank the God for giving me the ability to develop my cryptography.

Jongsung Kim
November 2006

Abstract

Differential and linear attacks are the most widely used cryptanalytic tools to evaluate the security of symmetric-key cryptography. Since the introduction of differential and linear attacks in the early 1990's, various variants of these attacks have been proposed such as the truncated differential attack, the impossible differential attack, the square attack, the boomerang attack, the rectangle attack, the differential-linear attack, the multiple linear attack, the nonlinear attack and the bilinear attack. One of the other widely used cryptanalytic tools is the related-key attack. Unlike the differential and linear attacks, this attack is based on the assumption that the cryptanalyst can obtain plaintext and ciphertext pairs by using different, but related keys.

This thesis provides several new combined differential, linear and related-key attacks, and shows their applications to block ciphers, hash functions in encryption mode and message authentication code (MAC) algorithms. The first part of this thesis introduces how to combine the differential-style, linear-style and related-key attacks: we combine them to devise the *differential-nonlinear attack*, the *square-(non)linear attack*, the *related-key differential-(non)linear attack*, the *related-key boomerang attack* and the *related-key rectangle attack*. The second part of this thesis presents some applications of the combined attacks to exiting symmetric-key cryptography. Firstly, we present their applications to the block ciphers SHACAL-1, SHACAL-2 and AES. In particular, we show that the differential-nonlinear attack is applicable to 32-round SHACAL-2, which leads to the best known attack on SHACAL-2 that uses a single key. We also show that the related-key rectangle attack is applicable to the full SHACAL-1, 42-round SHACAL-2 and 10-round AES-192, which lead to the first known attack on the full SHACAL-1 and the best known attacks on SHACAL-2 and AES-192 that use related keys. Secondly, we exploit the related-key boomerang attack to present practical distinguishing attacks on the cryptographic hash functions MD4, MD5 and HAVAL in encryption mode. Thirdly, we show that the related-key rectangle attack can be used to distinguish instantiated HMAC and NMAC from HMAC and NMAC with a random function.

Samenvatting

Differentiële en lineaire aanvallen zijn de meest gebruikte werktuigen van de cryptanalyse om de veiligheid van symmetrische-sleutel cryptografie te evalueren. Sinds de introductie van differentiële en lineaire aanvallen in de vroege jaren negentig, zijn verscheidene varianten van deze aanvallen geïntroduceerd, zoals de getrunceerde differentiële aanval, de onmogelijke differentiële aanval, de square-aanval, de boemerangaanval, de rechthoekaanval, de differentieel-lineaire aanval, de meervoudige lineaire aanval, de niet-lineaire aanval en de bilineaire aanval. Een van de andere vaak gebruikte werktuigen van de cryptanalyse is de verwante-sleutel aanval. Anders dan de differentiële en lineaire aanvallen, is deze aanval gebaseerd op de veronderstelling dat de cryptanalyst paren klaartekst en cijfertekst kan verkrijgen door het gebruiken van verschillende, maar gerelateerde sleutels.

Deze thesis brengt verscheidene nieuwe gecombineerde differentiële, lineaire en verwante-sleutel aanvallen aan, en toont hun toepassingen voor blokcijfers, hashfuncties in encryptiemode en algoritmes voor boodschapauthentiseringscodes (MAC). Het eerste deel van deze thesis introduceert hoe differentiële, lineaire en verwante-sleutel aanvallen gecombineerd kunnen worden: zo bekomen we de differentieel-niet-lineaire aanval, de square-(niet-)lineaire aanval, de verwante-sleutel differentieel-(niet-)lineaire aanval, de verwante-sleutel boemerangaanval en de verwante-sleutel rechthoekaanval. Het tweede deel van deze thesis presenteert enkele toepassingen van de gecombineerde aanvallen op bestaande symmetrische-sleutel cryptografie. Eerst stellen we hun toepassing voor op de blokcijfers SHACAL-1, SHACAL-2 en AES. In het bijzonder tonen we dat de differentieel-niet-lineaire aanval toepasbaar is op SHACAL-2 met 32 rondes, wat leidt tot de best gekende aanval op SHACAL-2 die een enkele sleutel gebruikt. We tonen ook aan dat de verwante-sleutel rechthoekaanval toepasbaar is op de volledige SHACAL-1, SHACAL-2 met 42 rondes en AES met 10 rondes, wat leidt tot de eerst gekende aanval op de volledige SHACAL-1 en de best gekende aanvallen op SHACAL-2 en AES-192 die gerelateerde sleutels gebruiken. Ten tweede buiten we de verwante-sleutel boemerangaanval uit om praktisch onderscheidende aan-

vallen voor te stellen op de cryptografische hashfuncties MD4, MD5 en HAVAL in encryptiemode. Ten derde tonen we dat de verwante-sleutel rechthoekaanval gebruikt kan worden om geïnstantieerde HMAC en NMAC te onderscheiden van HMAC en NMAC met een willekeurige functie.

Contents

List of Figures

List of Tables

List of Symbols

AES	Advanced Encryption Standard
ANSI	American National Standards Institute
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
FIPS	Federal Information Processing Standard
IEC	International Electrotechnical Commission
IETF	Internet Engineering Task Force
IPsec	Internet Protocol Security
ISO	International Organization for Standardization
MAC	Message Authentication Code
NESSIE	New European Schemes for Signatures, Integrity and Encryption
NIST	National Institute of Standards and Technology
RSA	Rivest-Shamir-Adleman
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
SSH	Secure Shell
TLS	Transport Layer Security
$\oplus$	the bitwise logical exclusive OR (XOR) operation
&	the bitwise logical AND operation
|	the bitwise logical OR operation
$\bullet$	the bitwise inner product
$+$	the addition modulo 2^{32} operation
$\neg$	the complement operation

Chapter 1

Introduction

1.1 Cryptography

We currently live in an information society. As information and communication technologies develop at an ever growing pace, the number of people who use these technologies on a day to day basis is also increasing. This development has brought substantial benefits but also introduced novel threats and vulnerabilities related to leakage of confidential information, the theft of identities and the unauthorized modification of data. This shows the need for the development of a reliable and trustworthy information infrastructure. This requires trustworthy systems and an essential building block for such systems is *cryptography*.

Cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication [97]. It makes extensive use of mathematics such as number theory, probability, statistics and combinatorics as well as of information theory, computational complexity and coding theory. There exist various cryptographic algorithms that provide confidentiality, data integrity and entity authentication services.

Confidentiality is a service for keeping the content of data from all but the authorized entities, so called Alice and Bob. This service makes it possible for Alice to send data to Bob such that no unauthorized entity can learn its content. Cryptographic primitives used to provide confidentiality are encryption and decryption algorithms. There are two types of algorithms: symmetric-key cryptography uses the same secret key for encryption and decryption, and asymmetric-key cryptography uses a public key for encryption and a private key for decryption. Asymmetric-key cryptography is also referred to as public-key cryptography (e.g., RSA [114] and ECC [79]). Symmetric-key cryptography for confidentiality en-

compasses block ciphers (e.g., DES [38] and AES [30]) and stream ciphers (e.g., RC4 [113]).

Data integrity is a service that detects the unauthorized alteration of data, which enables Alice to deliver information to Bob while Bob can check whether the information has been altered or not. Cryptographic tools or primitives that provide data integrity are hash functions (e.g., MD5 [112] and SHA-1 [41]), MAC algorithms (e.g., CBC-MAC [53] and HMAC [3]) and digital signature schemes (e.g., DSA [39] and RSA [114]). The former two belong to symmetric-key cryptography, while the latter belong to asymmetric-key cryptography (or public-key cryptography).

Entity authentication is a service that allows Bob to verify whether he is exchanging information with Alice or with someone else and whether Alice is taking actively part in this exchange. Entity authentication can be achieved through challenge-response protocols (e.g., Kerberos [100]), zero-knowledge protocols (e.g., Fiat-Shamir protocol [37]) and protocols for authenticated key agreement.

1.2 Cryptanalysis

Crytanalysis is the science of evaluating the security of ciphers. It is used by algorithm designers wishing to show that their ciphers are secure, or by attackers attempting to break them.

How can a designer show that a cipher is secure? One way is to show that the opponent cannot learn any new information on the plaintext from the ciphertext, even if the opponent has unlimited computational power. However, Shannon has shown in [118] that this kind of security (perfect information theoretic secrecy) requires that the secret key is at least as long as the plaintext, which is not practical for most applications. One could also try to show that finding information about the plaintext corresponds to solving a mathematical problem known to be hard. While this approach has had some success in asymmetric-key cryptography, no symmetric cryptographic scheme are known which are efficient and for which the security can be reduced to a well-established problem. In symmetric cryptography, the designer intends to demonstrate that the cipher resists all possible attacks. However, it is infeasible to check resistance to all kinds of possible attacks. As a countermeasure, the security of a cipher has been evaluated by checking its strength against the current state-of-the-art cryptanalysis. Hence, proposed ciphers have been able to win public confidence only if the designers have shown their resistance to known powerful and relevant attacks.

What does it mean to break a cipher (from the viewpoint of attackers)? A cipher can be considered broken if information is leaked on the plaintext from the ciphertext or if the secret key can be recovered more efficiently than by

exhaustive search. A cipher can even be considered to be broken if its outputs can be distinguished from those of a random cipher.

A strong cipher is secure even if the cryptanalyst knows the full description of the cipher except for the secret parameter, the key. This is a traditional assumption in cryptanalysis, called *Kerckhoffs' principle*. Under this principle, the cryptanalyst has access to the encryption box, the decryption box, or both of them, and only the key is unknown to him. It makes several following attack scenarios possible:

- ciphertext-only attack scenario – the cryptanalyst is assumed to learn only ciphertexts,
- known plaintext attack scenario – the cryptanalyst is assumed to learn plaintext and ciphertext pairs,
- chosen plaintext attack scenario – the cryptanalyst is assumed to choose plaintexts and get the corresponding ciphertexts,
- adaptive chosen plaintext and ciphertext attack scenario – the cryptanalyst is assumed to choose plaintexts, get the corresponding ciphertexts and then repeatedly choose additional ciphertexts (as a function of the output of the previous choices) and get the corresponding plaintexts.

Similarly, chosen ciphertext attack scenario and adaptive chosen ciphertext and plaintext attack scenario can be considered. Based on these attack scenarios, various attacks in symmetric-key cryptography have been proposed. In the following subsections, we briefly describe several important attacks. For this, we use an n-bit block cipher $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ divided into $E = E^f \circ E^0$, denoted $E_K = E^f_K(E^0_K(P))$, where P is an n-bit plaintext, K is a k-bit secret key, and E^0, E^f and E are all permutations on n bits for each k-bit secret key (see Fig. 1.1 and for the definitions of a block cipher including its key schedule and round, refer to Sect. 2.1).

1.2.1 The Differential Attack

Differential cryptanalysis [18], introduced by Biham and Shamir in 1990, is one of the most powerful chosen plaintext (or chosen ciphertext) attacks in symmetric-key cryptography (i.e., in block ciphers, stream ciphers, hash functions and MAC algorithms). After this attack was introduced, it has been applied effectively to many known ciphers and various variants of this attack have been proposed such as the truncated differential attack [74], the square attack [74, 29], the differential-linear attack [82], the impossible differential attack [8], the boomerang attack [123] and the rectangle attack [11].

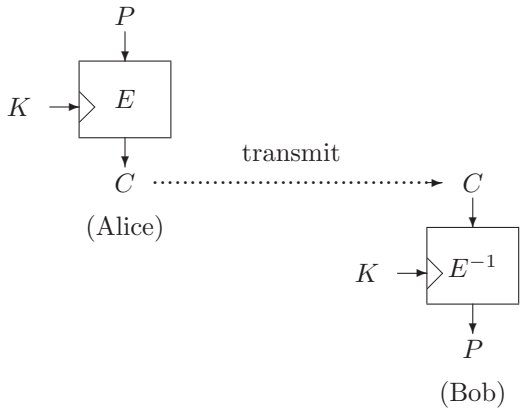


Figure 1.1: Block Cipher (P: Plaintext, C: Ciphertext, K: Key)

This attack first investigates a distribution of differences of output pairs for the first sub-cipher E^0 when their input pairs of E^0 have all the same difference. If this distribution is not uniform, then the differential attack can use this non-uniform distribution to retrieve the subkey for the second sub-cipher E^f by the following procedure (the subkey means the key in E^f that is generated by a key schedule and a k-bit secret key).

We assume that for E^0 there exists a differential[1] $\alpha \rightarrow \beta$ with probability p larger than 2^{-n}, i.e.,

$$Pr_{X,K}[E^0_K(X) \oplus E^0_K(X \oplus \alpha) = \beta] = p > 2^{-n},$$

where $Pr_{X,K}[\cdot]$ is an average probability over X and K. Then we can retrieve the subkey of E^f as follows:

1. Collect about $c \cdot p^{-1}$ plaintext pairs (P, P') whose differences are all α, where $c > 1$.

2. With a chosen plaintext attack scenario, obtain the corresponding ciphertext pairs (C, C').

3. For each subkey candidate k for E^f, calculate the number of (C, C') pairs

[1]In order for the distribution of output differences for E^0 to be uniform, the differential probability p should be 2^{-n} for any nonzero differences α and β.

satisfying

$$D_k^f(C) \oplus D_k^f(C') = \beta, \tag{1.1}$$

where $D_k^f = (E_k^f)^{-1}$. We denote this number by T_k. Output subkey k as the right key of E^f if all $T_k > T_{k'}$ for subkey candidates k' ($\neq k$).

If the subkey k is the right one, then about c ciphertext pairs are expected to pass the β test (see Eq. (1.1) for the β test) due to the differential for E^0. Otherwise, the expected number of ciphertext pairs passing the β test is about $c \cdot p^{-1} \cdot 2^{-n}$, for the β test has an n-bit filtering condition. Since $p > 2^{-n}$, the expected number of suggested ciphertext pairs for each wrong subkey is less than that for the right subkey. The success rate of this attack depends on the constant c and the number of subkey candidates for E^f. If we exploit an appropriate threshold to keep a portion of the subkey candidates for E^f instead of outputting the subkey with the maximal number T_k of hits (for instance, in Step 3, we can keep all the subkeys that suggest more than $c/2$ ciphertext pairs), then we can sieve many of wrong subkeys for E^f with a higher success rate.

1.2.2 The Linear Attack

Linear cryptanalysis [94], introduced by Matsui in 1993, is one of the most powerful known plaintext (or known ciphertext) attacks in symmetric-key cryptography (especially, in block ciphers and stream ciphers). It is known that this attack has similar properties to the differential attack when analyzing some block cipher structures: [1] shows that if an r-round Feistel structure is provably secure against the differential attack, then it is also provably secure against the linear attack, and vice versa.

This attack first investigates a correlation between the inputs and outputs for E^0. If the correlation leads to a linear approximation for E^0 with a relatively high bias, then the linear attack can use the linear approximation to retrieve the subkey for E^f. The attack can be described as follows.

If for E^0 there exists a linear approximation $\Gamma X \rightarrow \Gamma Y$ with bias ϵ such that $\epsilon > 2^{-\frac{n}{2}} \cdot c^{\frac{1}{2}}$ (or $c \cdot \epsilon^{-2} < 2^n$), where $c > 1$, i.e.,

$$|Pr_{X,K}[X \bullet \Gamma X \oplus E_K^0(X) \bullet \Gamma Y = 0] - \frac{1}{2}| = \epsilon,$$

where $X \bullet \Gamma X$ and $E_K^0(X) \bullet \Gamma Y$ are both bit-wise inner products, then the linear attack can retrieve the subkey of E^f as follows:

1. Collect about $c \cdot \epsilon^{-2}$ plaintexts P.

2. With a known plaintext attack scenario, obtain the corresponding ciphertexts C.

3. For each subkey candidate k for E^f, calculate the number of (P, C) pairs satisfying

$$P \bullet \Gamma X \oplus D_k^f(C) \bullet \Gamma Y = 0. \tag{1.2}$$

We denote this number by T_k. Output subkey k as the right key of E^f if all $|T_k - \frac{c \cdot \epsilon^{-2}}{2}| > |T_{k'} - \frac{c \cdot \epsilon^{-2}}{2}|$ for subkey candidates k' ($\neq k$).

This attack is based on the fact that for a wrong key Eq. (1.2) holds with a probability of approximately $\frac{1}{2}$ (i.e., bias 0), while for the right key it holds with bias ϵ. The success rate of this attack also depends on the constant c and the number of all possible subkey candidates for E^f. As in differential cryptanalysis, we can increase the success rate by keeping all the subkey candidates which satisfy an appropriate threshold.

After this attack was introduced, it has been extended and generalized into the multiple linear attack [57], the nonlinear attack [76], the chosen plaintext linear attack [75] and the bilinear attack [27].

1.2.3 The Related-Key Attack

In 1992 and 1993, Knudsen [72] and Biham [7] independently introduced a cryptanalytic method using related keys, called the related-key attack [7], which applies differential cryptanalysis to the cipher with different, but related unknown keys. This attack is based on the key scheduling algorithm and on the encryption/decryption algorithms, hence a cipher with a weak key scheduling algorithm may be vulnerable to this kind of attack (for the details of key scheduling and encryption/decryption algorithms of block ciphers, refer to Sect. 2.1).

This attack exploits a related-key differential $\alpha \rightarrow \beta$ for E^0 with probability p larger than 2^{-n}, i.e.,

$$Pr_{X,K}[E_K^0(X) \oplus E_{K \oplus \Delta K}^0(X \oplus \alpha) = \beta] = p > 2^{-n},$$

where ΔK is a non-zero known key difference chosen by the cryptanalyst. Then we can retrieve the subkey of E^f by applying this related-key differential to the differential attack algorithm.

The related-key attack is very difficult or even infeasible to conduct in many cryptographic applications, since it would certainly be unlikely that an attacker could persuade a sender to encrypt plaintexts under related keys unknown to the attacker. However, as demonstrated in [59, 104], the related-key attack is feasible

in some of the current real-world applications such as the IBM 4758 cryptoprocessor, key-exchange protocols that do not guarantee key integrity, and key-update protocols that updates session keys using a known function, for example, K, $K+1$, $K+2$, etc., where K is a session key.

1.3 Motivation of the Thesis

Evaluating the cryptanalytic strength of cryptographic algorithms builds confidence of users and encourages industry to develop innovative cryptographic technologies. However, it is difficult to prove that a given cryptographic algorithm is secure against all cryptographic attacks, so one has to prove its security against possible attacks which are meaningful and relevant. Taking into account the complexity of this work and the importance of the information security infrastructure, cryptanalytic tools should be developed. These kinds of tools allow to evaluate the security of cryptographic algorithms in a more accurate and reliable way.

There are mainly two approaches to develop cryptanalytic tools: one approach is to invent new cryptanalytic methods that are different from known ones (e.g., the differential attack [18], the linear attack [94] and the algebraic attack [28]), and the other approach is to generalize, extend or combine known cryptanalytic methods (e.g., the truncated differential attack [74], the non-linear attack [76] and the differential-linear attack [82]). In this thesis, we study how to combine existing attacks.

1.4 Contributions of the Thesis

We design several combinations of the differential-style, linear-style and related-key attacks. Combining the differential-style attacks with the linear-style attacks, we devise the *differential-nonlinear attack* and the *square-(non)linear attack*. We also combine the differential-(non)linear, boomerang and rectangle attacks with the related-key attack to devise the *related-key differential-(non)linear attack*, the *related-key boomerang attack* and the *related-key rectangle attack*. We analyze these combined attacks and show that they can be applied to block ciphers and MAC algorithms.

First, we present their applications to the block ciphers SHACAL-1, SHACAL-2 and AES: a square-nonlinear attack on 28-round SHACAL-2, a differential-nonlinear attack on 32-round SHACAL-2, a related-key differential-nonlinear attack on 35-round SHACAL-2, a related-key rectangle attack on 42-round SHACAL-2 and related-key rectangle attacks on the full 80-round SHACAL-1, 10-round AES-192 and 9-round AES-256. Our differential-nonlinear

attack is the best known attack on reduced SHACAL-2 that does not use related keys and our related-key rectangle attacks are the first known attack on the full SHACAL-1 and the best known attacks on reduced SHACAL-2 and reduced AES-192 that use related keys. Second, we present related-key boomerang attacks on encryption modes of MD4, MD5 and HAVAL. Our attacks are quite practical (in most of cases) and result in much faster attacks than previously known attacks in terms of the time and data complexities. Third, we show that the related-key rectangle attack can be used to distinguish the MAC algorithms HMAC and NMAC with the full 3-pass HAVAL and the full MD4 from HMAC and NMAC with a random function.

1.5 Outline of the Thesis

The thesis is organized as follows:

- Before moving to Chapter 2, we present notation and assumptions in the following section.
- Chapter 2 describes several cryptographic algorithms: the block ciphers SHACAL-1, SHACAL-2 and AES, the hash functions MD4, MD5, HAVAL, SHA-0 and SHA-1, and the MAC algorithms HMAC and NMAC.
- Chapter 3 introduces several new combined attacks: the differential-nonlinear attack, the square-(non)linear attack, the related-key differential-(non)linear attack, the related-key boomerang and rectangle attacks. These combined attacks have been introduced in [64, 67, 68, 119].
- Chapter 4 presents our combined attacks on the block ciphers SHACAL-1, SHACAL-2 and AES, which are based on the publications [35, 49, 68, 89, 119].
- Chapter 5 presents related-key boomerang attacks on MD4, MD5 and HAVAL in encryption mode, which have been published in [64].
- Chapter 6 presents distinguishing and forgery attacks on HMAC and NMAC based on HAVAL, MD4, MD5, SHA-0 and SHA-1 by using the differential and related-key rectangle techniques. The results presented in this chapter have been published in [63].
- Chapter 7 concludes the thesis.

1.6 Preliminaries

The following notation and assumptions are used throughout this thesis. The rightmost bit is referred to as the 0-th bit, i.e., the least significant bit, and round number starts with 0.

1.6.1 Notation

- P : A plaintext, for example, a 256-bit plaintext $P = (A, B, \cdots, H)$, where $A, B, \cdots, H$ are all 32-bit words.
- C : A ciphertext.
- P^r : The input of the $r\text{-}th$ round, for example, $P^r = (A^r, B^r \cdots, H^r)$.
- x_i^r : The $i\text{-}th$ bit of X^r, for example, $X^r \in \{A^r, B^r, \cdots, H^r\}$.
- ? : An unknown value or a set of unknown values.
- e_i : A 32-bit word that has $0's$ in all bit positions except for bit i.
- $e_{i_1,\cdots,i_k}$: $e_{i_1} \oplus \cdots \oplus e_{i_k}$, denoted also e_M, where $M = \{i_1, \cdots, i_k\}$.
- $e_{i_1,\cdots,i_k,\sim}$: A 32-bit word that has $1's$ in the position of bits $i_1, \cdots, i_k$, and arbitrary values in the position of bits $(i_k + 1)$-31, and $0's$ in the position of the other bits, where $i_1 < \cdots < i_k$. (The arbitrary value can be 0, 1 or an unknown value.)
- z_i : A 32-bit word that has 0 in the position of bit i, and arbitrary values in the positions of the other bits.
- CS (Constant Set) : A set containing a single value, repeated 2^{32} times.
- PS (Permutation Set): A set containing all 2^{32} possible values once, in an arbitrary order.
- $-PS$: A set containing all 2^{32} possible values once, in the order $-x$ if x occurs in PS at the same round.
- BS (Balanced Set) : A set containing 2^{32} elements with arbitrary values, but such that their sum (modulo 2^{32}) is zero. If this property only holds for the 0-th bit, we write BS_0.
- $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$: A block cipher that uses $\{0,1\}^k$ and $\{0,1\}^n$ as a key space and a plaintext/ciphertext space, respectively.

- $E = E^1 \circ E^0$ (i.e., $E_K(P) = E^1_K \circ E^0_K(P)$) : E is composed of E^0 and E^1 (E first performs E^0 and then E^1), where K is the k-bit secret key.
- $X \sim N(\mu, \sigma^2)$: a random variable X has the normal distribution, where μ is the expectation of X and σ is the standard deviation of X.
- $X \sim Bin(n, p)$: a random variable X has the binomial distribution, where n is the number of trials and p is the success rate for each trial.
- $X \sim \mathrm{Poi}(\lambda)$: a random variable X has the Poisson distribution, where λ is the expectation of X.

1.6.2 Assumptions

1. For the sub-cipher E^0 there exists a differential $\alpha \rightarrow \beta$ with a probability of p, i.e., $p = Pr_{X,K}[E^0_K(X) \oplus E^0_K(X \oplus \alpha) = \beta]$.
2. For the sub-cipher E^0 there exists a related-key differential $\alpha \rightarrow \beta$ with a probability of p^*, i.e., $p^* = Pr_{X,K}[E^0_K(X) \oplus E^0_{K \oplus \Delta K}(X \oplus \alpha) = \beta]$, where ΔK is a nonzero key difference chosen by the cryptanalyst.
3. For the sub-cipher E^1 there exists a differential $\gamma \rightarrow \delta$ with a probability of q, i.e., $q = Pr_{X,K}[E^1_K(X) \oplus E^1_K(X \oplus \gamma) = \delta]$.
4. For the sub-cipher E^1 there exists a related-key differential $\gamma \rightarrow \delta$ with a probability of q^*, i.e., $q^* = Pr_{X,K}[E^1_K(X) \oplus E^1_{K \oplus \Delta K'}(X \oplus \gamma) = \delta]$, where $\Delta K'$ is a nonzero key difference chosen by the cryptanalyst.
5. For the sub-cipher E^1 there exists a linear approximation $\lambda_I \rightarrow \lambda_C$ with a probability of $1/2+q'$, i.e., $1/2+q' = Pr_{X,K}[\lambda_I \bullet X \oplus \lambda_C \bullet E^1_K(X) \oplus \lambda_K \bullet K' = 0]$, where λ_K is a key mask, K' is the subkey used in E^1_K and $\bullet$ denotes the bit-wise inner product.
6. For the sub-cipher E^1 there exists a nonlinear approximation $\lambda_I \rightarrow f$ with a probability of $1/2+q''$, i.e., $1/2+q'' = Pr_{X,K}[\lambda_I \bullet X \oplus f(E^1_K(X), K') = 0]$ where f is a nonlinear function and K' is the subkey used in E^1_K.

Chapter 2

Selected Cryptographic Algorithms

2.1 Descriptions of Block Ciphers

Block ciphers consist of an encryption algorithm E, a decryption algorithm E^{-1} and a key scheduling algorithm. The encryption algorithm $E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ is a permutation family where E is a permutation on n bits for each k-bit key. If the length of inserted messages is larger than n bits, modes of operation are used to encrypt them such as ECB, CBC, CFB, OFB and CTR [120]. Block ciphers are one of the fundamental primitives to offer confidentiality of messages. Yet, they can also be used to build cryptographic hash functions that offer message authentication.

Modern block ciphers have significantly been affected by Shannon's pioneering work *Communication Theory of Secrecy Systems* [118], which shows that the iterative use of substitution (nonlinear layer) and permutation (linear layer) improves the security of a cipher. Based on this Shannon's work, most of modern block ciphers have been designed to use as many iterations of substitution and permutation as needed to give enough security (each iteration is referred to as *round*).

In this chapter, the block ciphers SHACAL-1, SHACAL-2 and AES are described.

2.1.1 SHACAL-1

In 2000 Handschuh and Naccache [43, 44] proposed a 160-bit block cipher SHACAL based on the standardized hash function SHA-1 [41] (cf. Sect. 2.2.5).

In 2001, they then proposed two versions, known as SHACAL-1 and SHACAL-2 [45], where SHACAL-1 is the same as the original SHACAL, while SHACAL-2 is a 256-bit block cipher based on the compression function of SHA-256 [42]. SHACAL-1 and SHACAL-2 were both submitted to the NESSIE project [99] and selected for the second phase of the evaluation: however, in 2003 SHACAL-1 was not recommended for a NESSIE portfolio because of concerns about its key schedule, while SHACAL-2 was selected to be in the NESSIE portfolio.

The SHACAL-1 cipher [43, 44] is a 160-bit block cipher based on the compression function of the hash standard SHA-1 [41]. It consists of 80 rounds and uses a variable key length up to 512 bits.

A 160-bit plaintext P of SHACAL-1 is composed of five 32-bit words A, B, C, D and E. According to our notation, the plaintext P is divided into A^0, B^0, C^0, D^0 and E^0, and the corresponding ciphertext C is divided into A^{80}, B^{80}, C^{80}, D^{80} and E^{80}. The *r-th* round of encryption is performed as follows:

$$\begin{aligned} A^{r+1} &= K^r + ROTL_5(A^r) + f^r(B^r, C^r, D^r) + E^r + Cst^r \\ B^{r+1} &= A^r \\ C^{r+1} &= ROTL_{30}(B^r) \\ D^{r+1} &= C^r \\ E^{r+1} &= D^r \end{aligned}$$

for $r = 0, \cdots, 79$, where $ROTL_j(X)$ represents rotation of the 32-bit word X to the left over j bits, K^r is the 32-bit round subkey, Cst^r is the 32-bit round constant, and

$$\begin{aligned} f^r(B^r, C^r, D^r) &= (B^r \& C^r) | (\neg B^r \& D^r), && (0 \le r \le 19) \\ f^r(B^r, C^r, D^r) &= B^r \oplus C^r \oplus D^r, && (20 \le r \le 39,\ 60 \le r \le 79) \\ f^r(B^r, C^r, D^r) &= (B^r \& C^r) | (B^r \& D^r) | (C^r \& D^r), && (40 \le r \le 59). \end{aligned}$$

We call these functions f_{if}, f_{xor} and f_{maj}, respectively.

Using the property $X - Y = X + (2^{32} - 1 - Y) + 1 = X + (\neg Y) + 1$ and the *r-th* round of encryption, we have the following *r-th* round of decryption:

$$\begin{aligned} A^r &= B^{r+1} \\ B^r &= ROTL_2(C^{r+1}) \\ C^r &= D^{r+1} \\ D^r &= E^{r+1} \\ E^r &= A^{r+1} + (\neg ROTL_5(B^{r+1})) + (\neg f^r(ROTL_2(C^{r+1}), D^{r+1}, E^{r+1})) + \\ &\quad (\neg Cst^r) + (\neg K^r) + 4. \end{aligned}$$

SHACAL-1 supports a variable key length up to 512 bits. However, the cipher is not intended to be used with a key shorter than 128 bits. If a shorter key than 512 bits is inserted in the cipher, the key is padded with zeroes to a 512-bit string. Let the 512-bit key string be denoted $K = [K^0||K^1||\cdots||K^{15}]$, where each K^i is a 32-bit word. The key expansion of 512 bits K to 2560 bits is defined by

$$K^i = ROTL_1(K^{i-3} \oplus K^{i-8} \oplus K^{i-14} \oplus K^{i-16}), \quad 16 \leq i \leq 79 \ .$$

2.1.2 SHACAL-2

The encryption of the SHACAL-2 cipher is performed as follows. The 256-bit plaintext is divided into eight 32-bit words – A, B, C, D, E, F, G and H. According to our notation, the plaintext P is divided into A^0, B^0, C^0, D^0, E^0, F^0, G^0 and H^0. Since this cipher is composed of 64 rounds, the ciphertext is divided into A^{64}, B^{64}, C^{64}, D^{64}, E^{64}, F^{64}, G^{64} and H^{64}. The r-th round of encryption can be described as follows:

$$\begin{aligned}
T_1^{r+1} &= H^r + \Sigma_1(E^r) + Ch(E^r, F^r, G^r) + Cst^r + K^r \\
T_2^{r+1} &= \Sigma_0(A^r) + Maj(A^r, B^r, C^r) \\
H^{r+1} &= G^r \\
G^{r+1} &= F^r \\
F^{r+1} &= E^r \\
E^{r+1} &= D^r + T_1^{r+1} \\
D^{r+1} &= C^r \\
C^{r+1} &= B^r \\
B^{r+1} &= A^r \\
A^{r+1} &= T_1^{r+1} + T_2^{r+1}
\end{aligned}$$

for $r = 0, ..., 63$, where K^r is the 32-bit round subkey, and Cst^r is the 32-bit round constant. The functions used in the above encryption process are defined as follows:

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

where $S_i(X)$ means a right rotation of X by i bits.

Using the property $X - Y = X + (\neg Y) + 1$ and the r-th round of encryption,

we have the following *r-th* round of decryption:

$$\begin{aligned}
T_1^{r+1} &= A^{r+1} - \Sigma_0(B^{r+1}) - Maj(B^{r+1}, C^{r+1}, D^{r+1}) \\
&= A^{r+1} + (\neg\Sigma_0(B^{r+1})) + (\neg Maj(B^{r+1}, C^{r+1}, D^{r+1})) + 2 \\
H^r &= T_1^{r+1} - \Sigma_1(F^{r+1}) - Ch(F^{r+1}, G^{r+1}, H^{r+1}) - K^r - W^r \\
&= T_1^{r+1} + (\neg\Sigma_1(F^{r+1})) + (\neg Ch(F^{r+1}, G^{r+1}, H^{r+1})) + (\neg K^r) + (\neg W^r) + 4 \\
G^r &= H^{r+1} \\
F^r &= G^{r+1} \\
E^r &= F^{r+1} \\
D^r &= E^{r+1} - T_1^{r+1} = E^{r+1} + (\neg T_1^{r+1}) + 1 \\
C^r &= D^{r+1} \\
B^r &= C^{r+1} \\
A^r &= B^{r+1}
\end{aligned}$$

The key schedule accepts a maximum 512-bit key and shorter keys than 512 bits are extended by padding the key with zeroes to a 512-bit string. In [45] it is strongly advised to use keys of at least 128 bits. Let the 512-bit key string be denoted $K = [K^0||K^1||\cdots||K^{15}]$. The key expansion of 512 bits K to 2048 bits is defined by

$$\begin{aligned}
K^i = \sigma_1(K^{i-2}) &+ K^{i-7} + \sigma_0(K^{i-15}) + K^{i-16}, \quad 16 \le i \le 63. \\
\sigma_0(x) &= S_7(x) \oplus S_{18}(x) \oplus R_3(x), \\
\sigma_1(x) &= S_{17}(x) \oplus S_{19}(x) \oplus R_{10}(x),
\end{aligned}$$

where $R_i(X)$ means the right shift of 32-bit word X by i bit positions.

2.1.3 AES

AES, the successor to DES, is a block cipher adapted as mandatory encryption standard by the US government. Since NIST announced that the block cipher Rijndael, designed by Daemen and Rijmen [30], was selected for the AES in 2000, it has gradually become one of the most widely used encryption algorithms in the world.

AES encrypts data blocks of 128 bits with 128, 192 or 256-bit keys. According to the length of the keys, AES uses a different number of rounds, i.e., it has 10, 12 and 14 rounds when used with 128, 192 and 256-bit keys, respectively. The round function of AES consists of the following four basic transformations:

- SubBytes (SB) is a nonlinear byte-wise substitution that applies the same 8×8 S-box to every byte.
- ShiftRows (SR) is a cyclic shift of the *i-th* row by i bytes to the left.

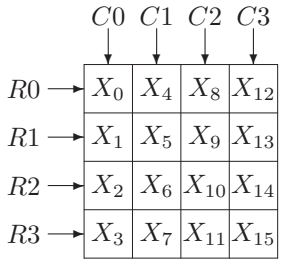


Figure 2.1: Byte Coordinate of a 128-bit Block of AES
(Ri: Row i, Ci: Column i, X_i: Byte i)

- MixColumns (MC) is a matrix multiplication applied to each column.
- AddRoundKey (ARK) is an exclusive-or with the round key.

Each round function of AES applies the BS, SR, MC and ARK steps in order, but MC is omitted in the last round. Before the first round, an extra ARK step is applied. We call the key used in this step a whitening key. For more details of the above four transformations, we refer to [30].

AES uses different key scheduling algorithms according to the length of the supplied keys. The key schedule of AES-128 accepts a 128-bit key (W_0, W_1, W_2, W_3) and generates subkeys $W_4, W_5, \cdots, W_{43}$, where each W_i is a 32-bit word composed of 4 bytes in column. The subkeys are generated by the following procedure:

- For $i = s$ till $i = t$ do the following (for AES-128, $s = 4$ and $t = 43$),
  - If $i \equiv 0 \bmod s$, then $W_i = W_{i-s} \oplus \mathrm{BS}(\mathrm{RotByte}(W_{i-1})) \oplus \mathrm{Rcon}(i/s)$,
  - else $W_i = W_{i-s} \oplus W_{i-1}$,

where RotByte represents one byte rotation and Rcon denotes fixed constants depending on its input. In AES-128, the whitening key is (W_0, W_1, W_2, W_3) and the subkey of round i is (W_{4i+4}, W_{4i+5},W_{4i+6},W_{4i+7}), where $0 \leq i \leq 9$.

Similarly, the key schedules of AES-192 and AES-256 accept 192- and 256-bit keys, and generate as many subkeys as required. The key schedule of AES-192 is exactly the same as that of AES-128 except for the use of $s = 6$ and $t = 51$. The subkeys of AES-256 are derived from the following procedure:

- For $i = 8$ till $i = 59$ do the following,

- If $i \equiv 0 \bmod 8$ then $W_i = W_{i-8} \oplus \mathrm{BS}(\mathrm{RotByte}(W_{i-1})) \oplus \mathrm{Rcon}(i/8)$,
- If $i \equiv 4 \bmod 8$ then $W_i = W_{i-8} \oplus \mathrm{BS}(W_{i-1})$,
- else $W_i = W_{i-8} \oplus W_{i-1}$.

In this thesis, a 128-bit block of AES is represented by a 4×4 byte matrix as in Fig. 2.1 or by $((X_0, X_1, X_2, X_3), (X_4, X_5, X_6, X_7), (X_8, X_9, X_{10}, X_{11}), (X_{12}, X_{13}, X_{14}, X_{15}))$.

2.2 Descriptions of Hash Functions

Hash functions are an important type of cryptographic algorithms; they are widely used in cryptographic applications such as digital signature, data authentication and e-cash. Hash functions are at work in the millions of transactions that take place on the internet every day. The purpose of the use of hash functions in many cryptographic protocols is to ensure their security as well as improve their efficiency. The most widely used hash functions are cryptographic hash functions such as MD5 [112] and SHA-1 [41], which follow the design principle of MD4.

Hash functions are message digest algorithms which compress any arbitrary-bit length message into a hash value with a small and fixed bit-length. The cryptographic hash functions such as MD4, MD5, HAVAL, SHA-0 and SHA-1 are performed based on the well-known Davies-Meyer construction, which is described as follows. Before the hash function is applied to a message M of arbitrary bit-length, it is padded to a multiple of t-bit and divided into n t-bit sub-messages $M^0||M^1||\cdots||M^{n-1}$, where t is specified. Then the l-bit hash value I^n for the message M is computed as follows:

$$I^0 = IV;\ I^{i+1} = \mathbf{com}(I^i, M^i) = E(I^i, M^i) + I^i \ \text{ for } \ 0 \le i < n, \tag{2.1}$$

where IV is a fixed l-bit initial value, **com** is a compression function and E is an iterative step function. In MD4, MD5, HAVAL, SHA-0 and SHA-1, the function E is composed of 3, 4 or 5 passes and in each pass there are 16, 20 or 32 rounds that use only simple basic operations and Boolean functions on 32-bit words. The l-bit input I^i is loaded into $l/32$ 32-bit registers denoted $(A^0, B^0, \cdots)$ and the t-bit message block is divided into $t/32$ 32-bit words denoted $(X^0, X^1, \cdots, X^{t/32})$. The $l/32$ registers are updated through a number of rounds. In each pass, a fixed Boolean function f and 32-bit constants Cst are used. Table 2.1 shows the parameters of HAVAL, MD4, MD5, SHA-0 and SHA-1.

Table 2.1: Parameters of HAVAL, MD4, MD5, SHA-0 and SHA-1

Hash Function	Bit-Length of Message Block (t)	Bit-Length of Initial Value (l)	# of Passes	# of Rounds in a Pass	Total # of Rounds
HAVAL	1024	256	3,4 or 5	32	96, 128 or 160
MD4	512	128	3	16	48
MD5	512	128	4	16	64
SHA-0	512	160	4	20	80
SHA-1	512	160	4	20	80

2.2.1 MD4

MD4 [111] is a cryptographic hash function introduced in 1990 by Rivest. It uses basic arithmetic operations and several Boolean functions which are suitable for fast software implementations on 32-bit processors. After MD4 was published, several hash functions based on the design philosophy of MD4 have been proposed: MD5 [112], HAVAL [130], RIPEMD [110], RIPEMD-160 [33], SHA-1 [41], SHA-256 [42], etc.

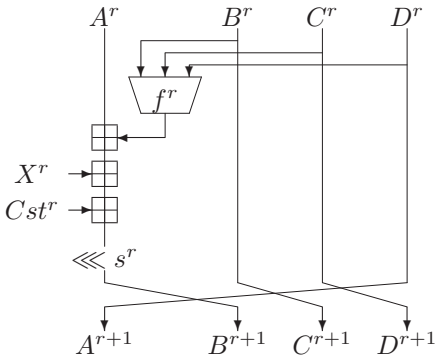


Figure 2.2: The r-th Round of the Compression Function of MD4

The r-th round of the compression function of MD4 is performed as in Fig. 2.2; the following three types of Boolean functions f and rotation amount s^r are used:

$$f^r(B^r, C^r, D^r) = \begin{cases} (B^r \& C^r) | (\neg B^r \& D^r) & \text{if } 0 \leq r \leq 15 \\ (B^r \& C^r) | (B^r \& D^r) | (C^r \& D^r) & \text{if } 16 \leq r \leq 31 \\ B^r \oplus C^r \oplus D^r & \text{if } 32 \leq r \leq 47 \end{cases}$$

$$s^r = \begin{cases} 3,7,11,19,3,7,11,19,3,7,11,19,3,7,11,19 & \text{if } 0 \le r \le 15 \\ 3,5,9,13,3,5,9,13,3,5,9,13,3,5,9,13 & \text{if } 16 \le r \le 31 \\ 3,9,11,15,3,9,11,15,3,9,11,15,3,9,11,15 & \text{if } 32 \le r \le 47 \end{cases}$$

In each pass, the supplied 512-bit message $M = X^0||X^1||\cdots||X^{15}$ is used exactly once by the following message expansion algorithm.

$$X^r = \begin{cases} X^0,X^1,X^2,X^3,X^4,X^5,X^6,X^7,X^8,X^9,X^{10},X^{11},X^{12},X^{13},X^{14},X^{15} & \text{if } 0 \le r \le 15 \\ X^0,X^4,X^8,X^{12},X^1,X^5,X^9,X^{13},X^2,X^6,X^{10},X^{14},X^3,X^7,X^{11},X^{15} & \text{if } 16 \le r \le 31 \\ X^0,X^8,X^4,X^{12},X^2,X^{10},X^6,X^{14},X^1,X^9,X^5,X^{13},X^3,X^{11},X^7,X^{15} & \text{if } 32 \le r \le 47 \end{cases}$$

2.2.2 MD5

MD5 [112] is a strengthened version of MD4, which increases the number of passes from 3 to 4 (i.e., it extends the number of rounds from 48 to 64) and uses the round function as in Fig. 2.3.

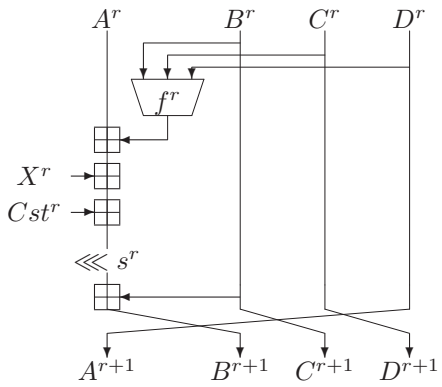


Figure 2.3: The r-th Round of the Compression Function of MD5

In MD5 four types of Boolean functions f are used; two of them are the same as the Boolean functions of MD4 used in rounds 1-15 and 32-47.

$$f^r(B^r,C^r,D^r) = \begin{cases} (B^r \& C^r)|(\neg B^r \& D^r) & \text{if } 0 \le r \le 15 \\ (B^r \& D^r)|(C^r \& \neg D^r) & \text{if } 16 \le r \le 31 \\ B^r \oplus C^r \oplus D^r & \text{if } 32 \le r \le 47 \\ C^r \oplus (B^r|\neg D^r) & \text{if } 48 \le r \le 63 \end{cases}$$

The rotation amount s^r is specified as follows:

$$s^r = \begin{cases} 7,12,17,22,7,12,17,22,7,12,17,22,7,12,17,22 & \text{if } 0 \le r \le 15 \\ 5,9,14,20,5,9,14,20,5,9,14,20,5,9,14,20 & \text{if } 16 \le r \le 31 \\ 4,11,16,23,4,11,16,23,4,11,16,23,4,11,16,23 & \text{if } 32 \le r \le 47 \\ 6,10,15,21,6,10,15,21,6,10,15,21,6,10,15,21 & \text{if } 48 \le r \le 63 \end{cases}$$

MD5 uses the following message expansion algorithm for a 512-bit message $M = X^0||X^1||\cdots||X^{15}$.

$$X^r = \begin{cases} X^0, X^1, X^2, X^3, X^4, X^5, X^6, X^7, X^8, X^9, X^{10}, X^{11}, X^{12}, X^{13}, X^{14}, X^{15} & \text{if } 0 \le r \le 15 \\ X^1, X^6, X^{11}, X^0, X^5, X^{10}, X^{15}, X^4, X^9, X^{14}, X^3, X^8, X^{13}, X^2, X^7, X^{12} & \text{if } 16 \le r \le 31 \\ X^5, X^8, X^{11}, X^{14}, X^1, X^4, X^7, X^{10}, X^{13}, X^0, X^3, X^6, X^9, X^{12}, X^{15}, X^2 & \text{if } 32 \le r \le 47 \\ X^0, X^7, X^{14}, X^5, X^{12}, X^3, X^{10}, X^1, X^8, X^{15}, X^6, X^{13}, X^4, X^{11}, X^2, X^9 & \text{if } 48 \le r \le 63 \end{cases}$$

2.2.3 HAVAL

In 1993 Zheng, Pieprzyk and Seberry proposed the one-way hashing algorithm HAVAL with 3, 4 and 5 passes [130]; the r-th round of the compression function of HAVAL is computed as in Fig. 2.4.

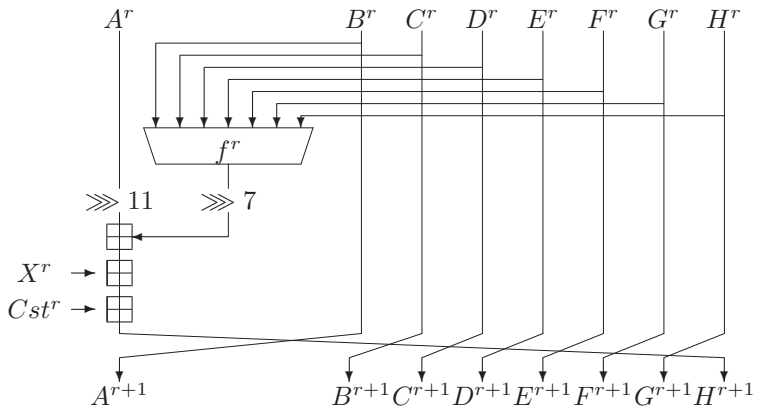


Figure 2.4: The r-th Round of the Compression Function of HAVAL

As stated above, HAVAL uses 3, 4 or 5 passes and each pass has 32 rounds. Like MD4 and MD5, HAVAL uses a 1024-bit message $M = X^0||X^1||\cdots||X^{31}$ exactly once in each pass by the following message expansion algorithm.

$$x^r = \begin{cases} x^0, x^1, x^2, x^3, x^4, x^5, x^6, x^7, x^8, x^9, x^{10}, x^{11}, x^{12}, x^{13}, x^{14}, x^{15} \\ x^{16}, x^{17}, x^{18}, x^{19}, x^{20}, x^{21}, x^{22}, x^{23}, x^{24}, x^{25}, x^{26}, x^{27}, x^{28}, x^{29}, x^{30}, x^{31} & \text{if } 0 \le r \le 31 \\ x^5, x^{14}, x^{26}, x^{18}, x^{11}, x^{28}, x^7, x^{16}, x^0, x^{23}, x^{20}, x^{22}, x^1, x^{10}, x^4, x^8 \\ x^{30}, x^3, x^{21}, x^9, x^{17}, x^{24}, x^{29}, x^6, x^{19}, x^{12}, x^{15}, x^{13}, x^2, x^{25}, x^{31}, x^{27} & \text{if } 32 \le r \le 63 \\ x^{19}, x^9, x^4, x^{20}, x^{28}, x^{17}, x^8, x^{22}, x^{29}, x^{14}, x^{25}, x^{12}, x^{24}, x^{30}, x^{16}, x^{26} \\ x^{31}, x^{15}, x^7, x^3, x^1, x^0, x^{18}, x^{27}, x^{13}, x^6, x^{21}, x^{10}, x^{23}, x^{11}, x^5, x^2 & \text{if } 64 \le r \le 95 \\ x^{24}, x^4, x^0, x^{14}, x^2, x^7, x^{28}, x^{23}, x^{26}, x^6, x^{30}, x^{20}, x^{18}, x^{25}, x^{19}, x^3 \\ x^{22}, x^{11}, x^{31}, x^{21}, x^8, x^{27}, x^{12}, x^9, x^1, x^{29}, x^5, x^{15}, x^{17}, x^{10}, x^{16}, x^{13} & \text{if } 96 \le r \le 127 \\ x^{27}, x^3, x^{21}, x^{26}, x^{17}, x^{11}, x^{20}, x^{29}, x^{19}, x^0, x^{12}, x^7, x^{13}, x^8, x^{31}, x^{10} \\ x^5, x^9, x^{14}, x^{30}, x^{18}, x^6, x^{28}, x^{24}, x^2, x^{23}, x^{16}, x^{22}, x^4, x^1, x^{25}, x^{15} & \text{if } 128 \le r \le 159 \end{cases}$$

In Fig. 2.4 the following five types of Boolean functions f are used (note that re-ordering functions ϕ are used to map from $(B^r, C^r, D^r, E^r, F^r, G^r, H^r)$ to $(x_0, x_1, x_2, x_3, x_4, x_5, x_6)$ which is the input to f^r).

$$f^r(x_0, x_1, x_2, x_3, x_4, x_5, x_6) = \begin{cases} (x_5 \& x_2) \oplus (x_4 \& x_1) \oplus (x_3 \& x_0) \oplus (x_6 \& x_5) \oplus x_6 & \text{if } 0 \le r \le 31 \\ (x_5 \& x_4 \& x_3) \oplus (x_4 \& x_2 \& x_1) \oplus (x_5 \& x_4) \oplus (x_5 \& x_2) \oplus (x_4 \& x_0) \\ \oplus (x_3 \& x_1) \oplus (x_2 \& x_1) \oplus (x_6 \& x_4) \oplus x_6 & \text{if } 32 \le r \le 63 \\ (x_5 \& x_4 \& x_3) \oplus (x_5 \& x_2) \oplus (x_4 \& x_1) \oplus (x_3 \& x_0) \oplus (x_6 \& x_3) \oplus x_6 & \text{if } 64 \le r \le 95 \\ (x_5 \& x_4 \& x_3) \oplus (x_4 \& x_2 \& x_1) \oplus (x_3 \& x_2 \& x_0) \oplus (x_5 \& x_2) \oplus (x_4 \& x_0) \\ \oplus (x_3 \& x_2) \oplus (x_3 \& x_1) \oplus (x_3 \& x_0) \oplus (x_2 \& x_1) \oplus (x_2 \& x_0) \oplus (x_6 \& x_2) \oplus x_6 & \text{if } 96 \le r \le 127 \\ (x_5 \& x_2) \oplus (x_4 \& x_1) \oplus (x_3 \& x_0) \oplus (x_6 \& x_5 \& x_4 \& x_3) \oplus (x_6 \& x_1) \oplus x_6 & \text{if } 128 \le r \le 159 \end{cases}$$

2.2.4 SHA-0

SHA-0 [40] is a standard hash function (FIPS 180), which is the initial version of SHA-1. It is exactly the same as SHA-1 except for the message expansion algorithm: SHA-0 uses the message expansion algorithm of SHA-1 without the one-bit rotation.

2.2.5 SHA-1

SHA-1 [41] is also a standard hash function (FIPS 180-1), from which the block cipher SHACAL-1 was designed. It is the hash mode of SHACAL-1: the message expansion algorithm is the same as the key scheduling algorithm of SHACAL-1 and the chaining values are the same as plaintexts of SHACAL-1.

Refer to [111, 112, 130, 40, 41] for IV and for the constants Cst^r of MD4, MD5, HAVAL, SHA-0 and SHA-1.

Encryption Modes of MD4, MD5, HAVAL, SHA-0 and SHA-1 Each of the rounds of MD4, MD5, HAVAL, SHA-0 and SHA-1 is an invertible function for each message word X^r. Hence, if we insert a secret key in the message part of M_i and a plaintext in the chaining value part of I_i, we get an invertible function from a compression function by removing the final addition with the previous chaining value. That is, $E(I_i, M_i)$ of Eq. (2.1) can be used in encryption mode $E(P, K)$, where P is a plaintext and K is a secret key. Therefore, according to Table 2.1, the encryption modes of MD4 and MD5 are 128-bit block ciphers with 512-bit keys and with 48 and 64 rounds, respectively, the encryption modes of SHA-0 and SHA-1 are 160-bit block ciphers with 512-bit keys and with 80 rounds, respectively, and the encryption mode of HAVAL is a 256-bit block cipher with 1024-bit keys and with 96, 128 or 160 round. Note that the encryption mode of SHA-1 is the block cipher SHACAL-1. In these encryption modes, we use the notation P and K for a plaintext and a key, respectively.

2.3 Descriptions of MAC Algorithms

MAC algorithms are widely used in Internet security protocols (SSL/TLS, SSH, IPsec) and in the financial sector for debit and credit transactions. MAC algorithms are keyed hash functions that allow to verify whether a transmitted message has been altered. In order to use a MAC algorithm in computer networks, a secret key should be first distributed to the authorized entities, Alice and Bob. When Alice sends a message to Bob, she computes the MAC value of the message with the shared secret key and appends it to the message. Once Bob receives the message and its MAC value, he recomputes the MAC value of the obtained message with the key and verifies the authenticity of the message by checking if the recomputed MAC value is the same as the received MAC value. The security of a MAC algorithm depends on the difficulty for an unauthorized entity to produce a forgery, that is, a new message with a valid MAC. Typically, the forger is allowed to query the MAC generation oracle with adaptively chosen queries (see for example [4, 107]).

In the literature there have been mainly two types of MAC algorithms: block cipher based MAC algorithms (e.g., CBC-MAC [53], TMAC [81], RMAC [56] and OMAC [54]) and hash function based MAC algorithms (e.g., NMAC [3], HMAC [3] and MDx-MAC [108]). Both types of MAC algorithms usually inherit the security and efficiency of its underlying primitives. Other MAC algorithms are based on the design principle of a stream cipher (e.g., SOBER [115]) and a universal hash function (e.g., UMAC [20] and Poly1305–AES MAC [6]). Furthermore, several authenticated encryption schemes have been proposed that offer both confidentiality and authenticity of a message (e.g., CWC [80], EAX [5] and GCM [96]). In this thesis, we consider the hash function based MAC algorithms HMAC

and NMAC.

2.3.1 HMAC

HMAC, designed by Bellare, Canetti and Krawczyk, is a widely used message authentication code and a pseudorandom function generator based on cryptographic hash functions such as MD5 and SHA-1. It has been standardized by ANSI, IETF, ISO and NIST. HMAC takes a message of an arbitrary bit-length and hashes it with one secret key. For the same length of the message it calls the compression function of the underlying hash function additionally three more times than the iterated hash construction, i.e., the Merkle-Damgård construction (shortly the MD construction [31, 98]; it is defined below). For long messages, its efficiency is thus almost the same as the MD construction. Furthermore, cryptographic hash functions such as MD5 and SHA-1 can be used in HMAC, which are more efficient in software than block ciphers, and thus HMAC is typically faster than block cipher based MAC algorithms. The general description of HMAC is as follows.

HMAC [3] applies in both its inner and outer parts the iterated MD construction of a hash function H given a compression function h; the MD construction is defined as

$$H(IV, M) = h(\cdots h(h(IV, M^1), M^2) \cdots, M^n),$$

where IV is an l-bit fixed initial value and M is an arbitrary-length message which is padded to a multiple of t bits and divided into n t-bit blocks $M^0||M^1||\cdots||M^{n-1}$ (note that the outputs of functions h and H are l-bit strings).

$$\begin{aligned} \mathbf{HMAC}(K, M) &= H(IV, (K \oplus opad)||H(IV, (K \oplus ipad)||M)) \\ &= h(h(IV, (K \oplus opad)), H(h(IV, (K \oplus ipad)), M)), \end{aligned} \tag{2.2}$$

where K is the secret key, $opad$, $ipad$ are constants and $|K \oplus opad| = |K \oplus ipad| = t$. If HMAC takes a one-block message M, it can be expressed as

$$\mathbf{HMAC}(K, M) = h(h(IV, (K \oplus opad)), h(h(IV, (K \oplus ipad)), M)). \tag{2.3}$$

In order to facilitate the description of our analysis of HMAC we denote the four compression functions h in (2.3) by h_1, h_2, h_3 and h_4, and the four functions in (2.2) by h_1, H_2, h_3 and h_4. See Fig. 2.5 for a schematic description of HMAC with this notation. Note that the outputs of H_2 and h_2 are padded to a t-bit string to be inserted into h_4.

In practice the function h can be replaced by the compression function of cryptographic hash functions such as HAVAL [130], MD4 [111], MD5 [112], SHA-0 [40] and SHA-1 [41].

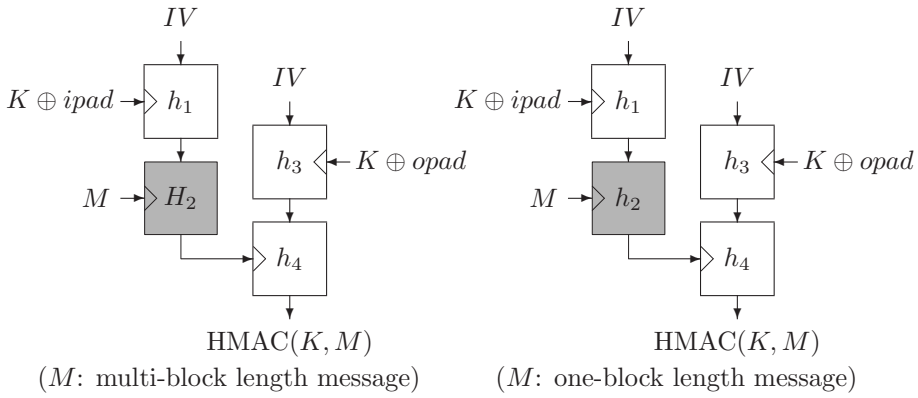


Figure 2.5: Schematic Description of HMAC

2.3.2 NMAC

NMAC is a generalized version of HMAC, which uses two l-bit secret keys (K_1, K_2). It is computed as follows:

$$\mathbf{NMAC}(K_1, K_2, M) \quad = \quad H(K_2, H(K_1, M))\,.$$

NMAC has exactly the same structure as HMAC except for the use of the keys, i.e., in NMAC the secret keys K_1 and K_2 are used instead of $h_1(IV, K \oplus ipad)$ and $h_3(IV, K \oplus opad)$.

Chapter 3

New Combined Attacks

This chapter introduces five new combined attacks: *the differential-nonlinear attack*, *the square-(non)linear attack*, *the related-key differential-(non)linear attack*, *the related-key rectangle attack* and *the related-key boomerang attack*. Each of these combined attacks treat a cipher as a cascade of two sub-ciphers, apply a known differential-style, linear-style or related-key distinguisher to each sub-cipher and then combines them to obtain a new distinguisher on the cipher.

3.1 The Differential-Nonlinear Attack

In [82] Langford and Hellman showed that differential cryptanalysis and linear cryptanalysis can be combined into a technique called the differential-linear attack. This attack is described as follows.

In order to make a distinguisher for $E = E^1 \circ E^0$ the differential-linear attack uses a differential $\alpha \rightarrow \beta$ for E^0 with probability p, and a linear approximation $\lambda_I \rightarrow \lambda_C$ for E^1 with probability $\frac{1}{2}+q'$. Let P and P^* be a pair of plaintexts that satisfy $P \oplus P^* = \alpha$. Langford and Hellman [82] suggested to use a truncated differential[1] $\alpha \rightarrow \beta$ for E^0 with probability 1. This allows us to get one bit equation

$$\lambda_I \bullet (E_K^0(P) \oplus E_K^0(P^*)) = \lambda_I \bullet (I \oplus I^*) = a \tag{3.1}$$

with probability 1, where $I = E_K^0(P)$, $I^* = E_K^0(P^*)$, $a = \lambda_I \bullet \beta$ and $\bullet$ denotes the

[1] A truncated differential is a set of differentials, hence it has more than one input difference or output difference [74].

bitwise inner product. According to Assumption 4 in Sect. 1.6.2, we also have

$$\lambda_I \bullet I \oplus \lambda_C \bullet C \oplus \lambda_K \bullet K' = 0, \tag{3.2}$$

$$\lambda_I \bullet I^* \oplus \lambda_C \bullet C^* \oplus \lambda_K \bullet K' = 0 \tag{3.3}$$

with probability $\frac{1}{2} + q'$ each, where $C = E^1_K(E^0_K(P))$, $C^* = E^1_K(E^0_K(P^*))$, λ_K is a key mask and K' is the subkey used in E^1_K. Hence, using the piling up lemma presented in [94] (i.e., summing over Eqs. (3.1), (3.2), (3.3)), we have the following equation

$$\lambda_C \bullet C \oplus \lambda_C \bullet C^* = \lambda_C \bullet E^1_K(E^0_K(P)) \oplus \lambda_C \bullet E^1_K(E^0_K(P^*)) = a \tag{3.4}$$

with probability $\frac{1}{2} + 2q'^2$ ($=\frac{1}{2} + 2^{3-1} \cdot \frac{1}{2} \cdot q' \cdot q'$). So the attack requires $O(q'^{-4})$ chosen plaintext pairs to work (cf. Sect. 1.2.2).

In [12] Biham, Dunkelman and Keller extended the above technique to an event where the probability of the differential part is smaller than 1. The description of the enhanced differential-linear attack is as follows.

If the plaintext pair P and P^* satisfies the differential $\alpha \rightarrow \beta$ (with probability p (≤ 1)), Eq. (3.1) holds with probability 1. If the plaintext pair P and P^* does not satisfy the differential (with probability $1-p$), we assume that $\lambda_I \bullet (E^0_K(P) \oplus E^0_K(P^*))$ follows a random behavior. From the above two cases, we get Eq. (3.1) with probability $\frac{1}{2} + \frac{p}{2}$ ($= p \cdot 1 + (1-p) \cdot \frac{1}{2}$). Recall that Eqs. (3.2), (3.3) hold with probability $\frac{1}{2} + q'$ each. Similarly, we sum over Eqs. (3.1), (3.2), (3.3) to obtain Eq. (3.4) with probability $\frac{1}{2} + 2p \cdot q'^2$ ($= \frac{1}{2} + 2^{3-1} \cdot \frac{p}{2} \cdot q'^2$). The attack requires $O(p^{-2}q'^{-4})$ chosen plaintext pairs to work.

We are now ready to introduce the differential-nonlinear attack. This attack uses a differential-nonlinear distinguisher that concatenates a nonlinear approximation for E^1 to a differential for E^0. However, the nonlinear approximation should be of a special form which can be attached to a differential, i.e., the input mask of the nonlinear approximation should be linear. Note that we cannot predict the specific two output values of E^0 even though we can learn the output difference β of E^0 with probability p.

This attack uses a nonlinear approximation $\lambda_I \rightarrow f$ for E^1 with probability $\frac{1}{2} + q''$ instead of a linear approximation (for the definition of the nonlinear approximation, refer to Assumption 6 in Sect. 1.6.2). In the same way, we get Eq. (3.1) with probability $\frac{1}{2} + \frac{p}{2}$. Our nonlinear approximation satisfies

$$\lambda_I \bullet I \oplus f(C, K') = 0, \tag{3.5}$$

$$\lambda_I \bullet I^* \oplus f(C^*, K') = 0 \tag{3.6}$$

with probability $\frac{1}{2} + q''$ each and thus we sum over Eqs. (3.1), (3.5), (3.6) to obtain the one bit equation

$$f(C, K') \oplus f(C^*, K') = f(E^1_K(E^0_K(P)), K') \oplus f(E^1_K(E^0_K(P^*)), K') = a$$

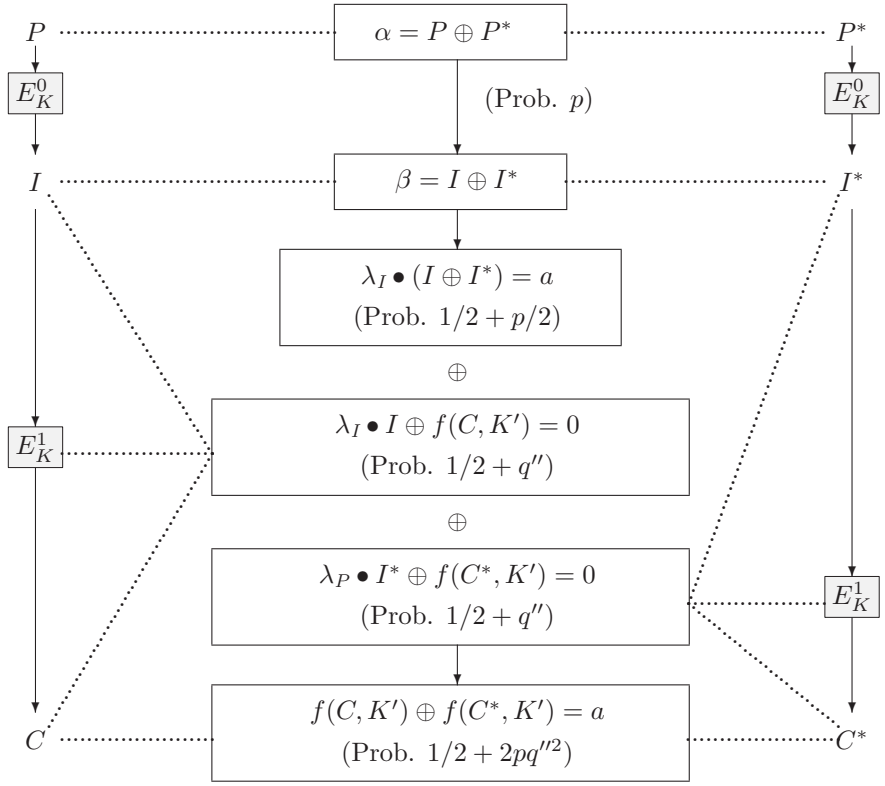


Figure 3.1: Differential-Nonlinear Distinguisher

with probability $\frac{1}{2} + 2p \cdot q''^2$ $(= \frac{1}{2} + 2^{3-1} \cdot \frac{p}{2} \cdot q''^2)$ (see Fig. 3.1). The attack requires $O(p^{-2}q''^{-4})$ chosen plaintext pairs to work.

3.2 The Square-(Non)linear Attack

In this section, we introduce the square-linear and square-nonlinear attacks that combine the square attack with the linear and nonlinear attacks, respectively. These attacks are similar to the differential-(non)linear techniques.

The square attack was introduced when the block cipher SQUARE was proposed [29]. After this attack was introduced, it has been extended and generalized to the multiset attack [22] and the integral attack [77]. The basic idea behind this attack is the same as that of the higher-order differential attack [74]. It exploits a square characteristic whose input data consist of a set of plaintexts in which some bits are formed of a saturated set, and whose output data have a property like balancedness in some bits. We call a set "a saturated set" if every value in $\{0,1\}^w$ is found exactly once in the set, where w is some bit length. Balancedness means that the sum of all the elements is zero.

The square-linear attack assumes that for the E^0 sub-cipher there exists a square characteristic whose input data consist of a set of 2^m plaintexts P_i $(i = 0, \ldots, 2^m - 1)$, and whose output data have a balanced property in some bits. If the balanced output bits of the square characteristic include the bits of λ_I that are 1, then we have one bit equation $\lambda_I \bullet (\bigoplus_{i=0}^{2^n-1} E_K^0(P_i)) = \lambda_I \bullet (\bigoplus_{i=0}^{2^n-1} I_i) = 0$ with probability 1, where $I_i = E_K^0(P_i)$. According to Assumption 4 in Sect. 1.6.2, we also have $\lambda_I \bullet I_i \oplus \lambda_C \bullet C_i \oplus \lambda_K \bullet K' = 0$ with probability $\frac{1}{2} + q'$ for each plaintext P_i, where $C_i = E_K^1(E_K^0(P_i))$. Hence, summing over all the $(2^m + 1)$ equations we have the following equation

$$\lambda_C \bullet (\bigoplus_{i=0}^{2^m-1} C_i) = \lambda_C \bullet (\bigoplus_{i=0}^{2^m-1} E_K^1(E_K^0(P_i))) = 0$$

with probability $\frac{1}{2} + 2^{2^m-1}q'^{2^m}$ (by the piling up lemma [94]). This attack requires $O((2^{2^m-1}q'^{2^m})^{-2})$ chosen plaintext sets to work. Thus, this attack can be efficiently applied to ciphers if $q' \approx 1/2$.

Furthermore, we can extend the above attack to the cases where a nonlinear approximation is used instead of a linear approximation for E^1. This attack uses a nonlinear approximation $\lambda_I \rightarrow f$ for E^1 with probability $\frac{1}{2} + q''$ to concatenate the square characteristic for E^0. In the same way, we get the one bit equation $\lambda_I \bullet (\bigoplus_{i=0}^{2^m-1} I_i) = 0$ with probability 1. Our nonlinear approximation satisfies $\lambda_I \bullet I_i \oplus f(C_i, K') = 0$ with probability $\frac{1}{2} + q''$ for each plaintext P_i and thus we

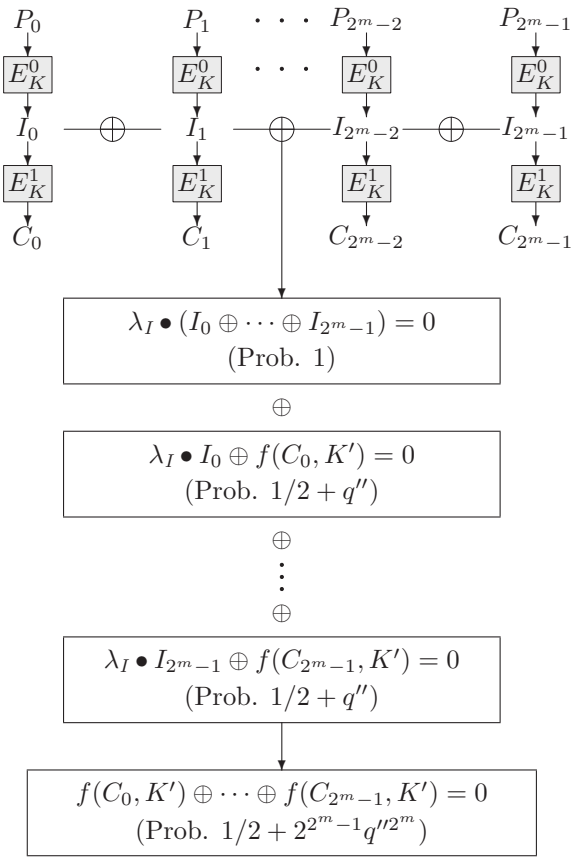


Figure 3.2: Square-Nonlinear Distinguisher

use the piling up lemma [94] to obtain the one bit equation

$$\bigoplus_{i=0}^{2^m-1} f(C_i, K') = \bigoplus_{i=0}^{2^m-1} f(E_K^1(E_K^0(P_i)), K') = 0$$

with probability $\frac{1}{2} + 2^{2^n-1}q''^{2^n}$. This attack requires $O((2^{2^n-1}q''^{2^n})^{-2})$ chosen plaintext sets to work. We call the extended attack *the square-nonlinear attack*. See Fig. 3.2 for a schematic description of the square-nonlinear attack.

3.3 The Related-Key Differential-(Non)linear Attack

In 1998 Hawkes [46] presented the related-key differential-linear attack which is a combination of the related-key and differential-linear attacks. The attack presented in [46] uses a related-key differential with probability 1 and a linear approximation with bias $\frac{1}{2}$. However, we can extend this technique to the general cases where the probability of the related-key differential is less than or equal to 1 and the bias of linear approximation is less than or equal to $\frac{1}{2}$. Furthermore, we can extend it into a technique called the related-key differential-nonlinear attack.

The related-key differential-linear attack requires the encryptions of plaintext pairs P and P^* under keys K and K^*, respectively, where K and K^* are different, but related keys. This attack uses a related-key differential $\alpha \rightarrow \beta$ for E^0 with a probability of p^* (i.e., $p^* = Pr_{X,K}[E_K^0(X) \oplus E_{K^*}^0(X^*) = \beta | X \oplus X^* = \alpha, K \oplus K^* = \Delta K]$, where ΔK is a specific key difference) and a linear approximation $\lambda_I \rightarrow \lambda_C$ for E^1 with a probability of $\frac{1}{2} + q'$ (i.e., $\frac{1}{2} + q' = Pr_{X,K}[\lambda_I \bullet X \oplus \lambda_C \bullet E_K^1(X) \oplus \lambda_K \bullet K' = 0]$).

With a similar argument of the enhanced differential-linear attack, we get one bit equation

$$\lambda_I \bullet (E_K^0(P) \oplus E_{K^*}^0(P^*)) = a \tag{3.7}$$

with probability $\frac{1}{2} + \frac{p^*}{2} (= p^* \cdot 1 + (1 - p^*) \cdot \frac{1}{2})$, where $P \oplus P^* = \alpha$. According to Assumption 4 in Sect. 1.6.2, we also have the following two linear approximations

$$\lambda_I \bullet E_K^0(P) \oplus \lambda_C \bullet E_K^1(E_K^0(P)) \oplus \lambda_K \bullet K' = 0 \tag{3.8}$$

$$\lambda_I \bullet E_{K^*}^0(P^*) \oplus \lambda_C \bullet E_{K^*}^1(E_{K^*}^0(P^*)) \oplus \lambda_K \bullet K^{*'} = 0 \tag{3.9}$$

with probability $\frac{1}{2} + q'$, respectively. Hence, applying Eqs. (3.7), (3.8), (3.9) to the piling up lemma presented in [94] (i.e, summing over Eqs. (3.7), (3.8), (3.9)), we have the following equation

$$\lambda_C \bullet E_K^1(E_K^0(P)) \oplus \lambda_C \bullet E_{K^*}^1(E_{K^*}^0(P^*)) \oplus \lambda_K \bullet K' \oplus \lambda_K \bullet K^{*'} = a$$

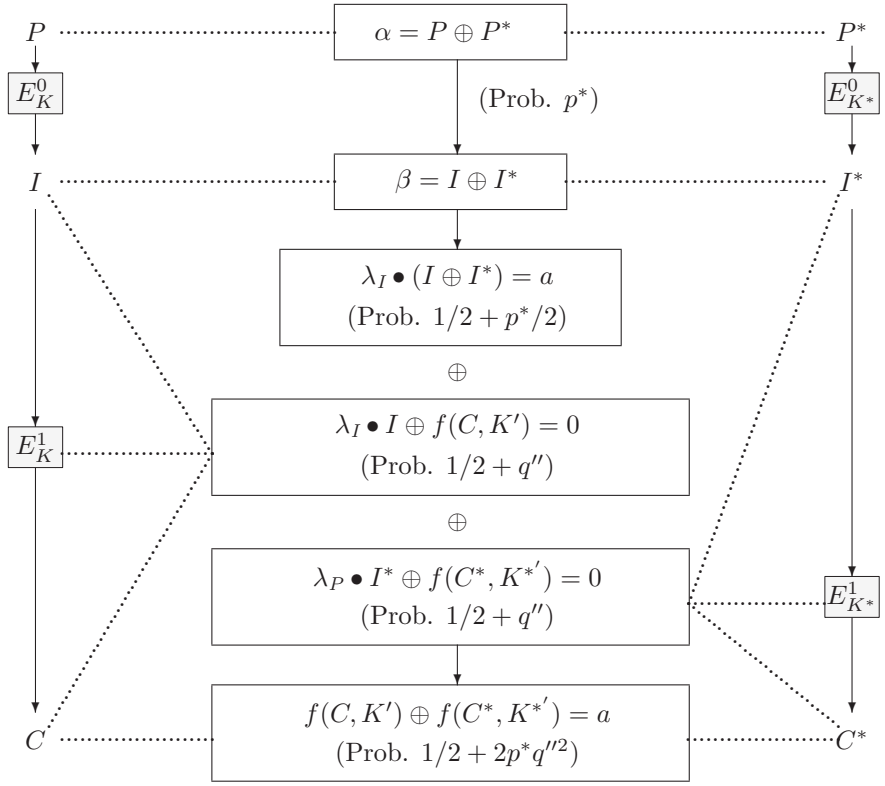


Figure 3.3: Related-Key Differential-Nonlinear Distinguisher

with probability $\frac{1}{2} + 2p^*q'^2$ $(= \frac{1}{2} + 2^{3-1} \cdot \frac{p^*}{2} \cdot q'^2)$. That is, we obtain

$$\lambda_C \bullet E_K(P) \oplus \lambda_C \bullet E_{K^*}(P^*) = 0$$

with bias $2p^*q'^2$. Hence the attack using the related-key differential-linear distinguisher requires $O(p^{*-2}q'^{-4})$ related-key chosen plaintexts to succeed.

As stated above, this attack can be extended to the cases where a nonlinear approximation is used instead of a linear approximation. A nonlinear approximation $\lambda_I \to f$ with a probability of q'' is used for E^1. With a similar argument of the differential-nonlinear attack, we can get

$$f(E_K(P), K') \oplus f(E_{K^*}(P^*), K^{*'}) = 0$$

with bias $2p^*q''^2$ (see Fig. 3.3). The attack using the related-key differential-nonlinear distinguisher requires $O(p^{*-2}q''^{-4})$ related-key chosen plaintexts to succeed.

3.4 The Related-Key Rectangle and Boomerang Attacks

In this section, we introduce the related-key rectangle and boomerang attacks. In these attacks, there exist three types of related-key rectangle and boomerang distinguishers according to the usage of related-key differentials and the number of related keys. The first type of distinguisher is applicable when related-key differentials are used in the first sub-cipher, and regular differentials (or related-key differentials with the same key difference as those used in the first sub-cipher) in the second sub-cipher. The second type uses related-key differentials in the second sub-cipher and regular differentials for the first sub-cipher. The third type uses related-key differentials in both sub-ciphers. The first and second types of distinguishers use two related keys, but they use different methods for selecting plaintexts to work with. On the other hand, the third type of distinguisher uses four related keys. We call these three types of distinguishers *related-key rectangle and boomerang distinguishers of TYPE 1, TYPE 2 and TYPE 3*, respectively.

We first introduce the three types of related-key rectangle distinguishers and then of related-key boomerang distinguishers. The related-key rectangle distinguishers of TYPE 1, TYPE 2 and TYPE 3 work as follows:

- Choose two random n-bit plaintexts P and P' and compute two other plaintexts $P^* = P \oplus \alpha$ and $P'^* = P' \oplus \alpha$ for a constant α.
- With a chosen plaintext attack scenario, obtain the corresponding ciphertexts $C = E_K(P)$, $C^* = E_{K^*}(P^*)$, $C' = E_{K'}(P')$ and $C'^* = E_{K'^*}(P'^*)$,

where $K^* = K \oplus \Delta K$, $K' = K \oplus \Delta K'$, $K'^* = K \oplus \Delta K \oplus \Delta K'$ (i.e., $K \oplus K^* = K' \oplus K'^* = \Delta K$ and $K \oplus K' = K^* \oplus K'^* = \Delta K'$) and ΔK, $\Delta K'$ are key differences chosen by the cryptanalyst.

- Check if $C \oplus C' = C^* \oplus C'^* = \delta$ or $C \oplus C'^* = C^* \oplus C' = \delta$.

As stated, the related-key rectangle distinguisher checks if the two pairs chosen from the ciphertext quartet have the same difference δ. If this difference δ holds with a higher probability than for a random cipher, then the related-key rectangle distinguisher can be applied effectively to the underlying cipher.

In the above process the difference among the three types of distinguishers is on the condition of the key differences ΔK and $\Delta K'$. Namely, in TYPE 1 $\Delta K \neq 0$ and $\Delta K' = 0$ (or $\Delta K = \Delta K' \neq 0$), in TYPE 2 $\Delta K = 0$ and $\Delta K' \neq 0$ and in TYPE 3 $\Delta K \neq 0$, $\Delta K' \neq 0$ and $\Delta K \neq \Delta K'$. If the plaintext quartet (P, P^*, P', P'^*) satisfies the last δ test, we call such a quartet a *right quartet*.

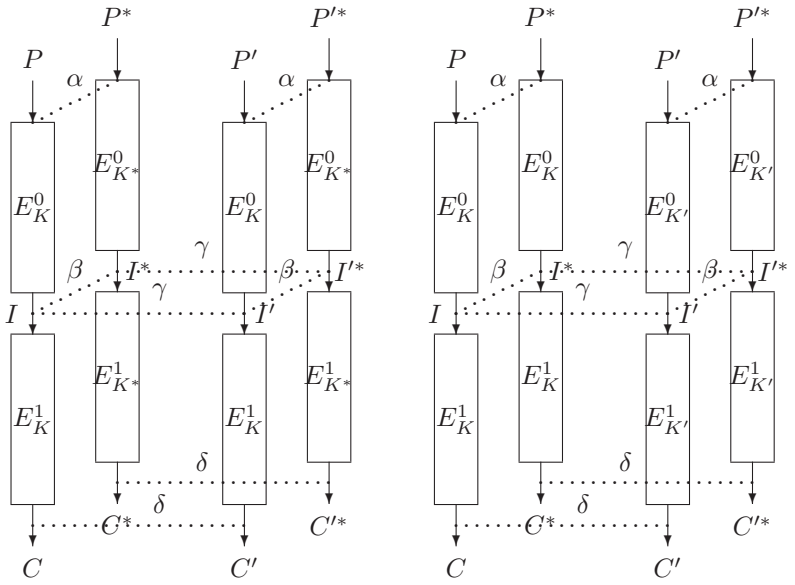
The related-key rectangle distinguishers can be formed by building quartets of plaintexts (P, P^*, P', P'^*) that satisfy the following four differential conditions.

- Differential Condition 1 : $P \oplus P^* = P' \oplus P'^* = \alpha$
- Differential Condition 2 : $I \oplus I^* = I' \oplus I'^* = \beta$ (for some β)
- Differential Condition 3 : $I \oplus I' = \gamma$ (or $I \oplus I'^* = \gamma$) (for some γ)
- Differential Condition 4 : $C \oplus C' = C^* \oplus C'^* = \delta$ (or $C \oplus C'^* = C^* \oplus C' = \delta$)

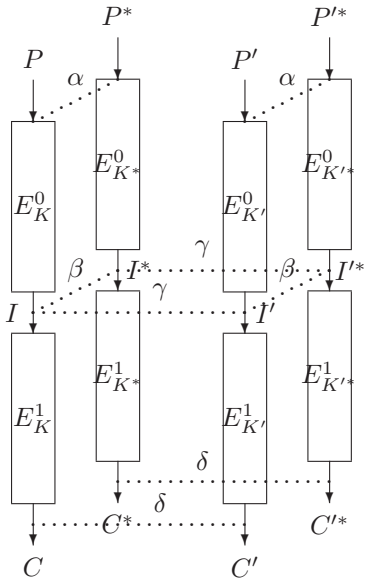
where $I = E^0_K(P)$, $I^* = E^0_{K^*}(P^*)$, $I' = E^0_{K'}(P')$ and $I'^* = E^0_{K'^*}(P'^*)$. In these four differential conditions, α and δ represent specific differences, and β and γ represent arbitrary differences. Note that the differential conditions 2 and 3 imply $I^* \oplus I'^* = \gamma$ (or $I^* \oplus I' = \gamma$) with probability 1. If these four differential conditions are satisfied, such a quartet (P, P^*, P', P'^*) is a right quartet. See Fig. 3.4 for schematic descriptions of these kinds of right quartets. We now analyze the three types of distinguishers in terms of the right quartets described in Fig. 3.4.

3.4.1 Related-Key Rectangle Distinguisher of TYPE 1

Assume that we have m plaintext pairs with difference α, where one plaintext of each pair is encrypted with the key K and the other plaintext with the key K^*, then we have about mp^* pairs satisfying the related-key differential $\alpha \rightarrow \beta$ for E^0 under the key difference ΔK. The mp^* pairs generate about $\frac{(mp^*)^2}{2}$ quartets satisfying conditions 1 and 2. Assuming that the intermediate encryption values are uniformly distributed over all possible values, we get $I \oplus I' = \gamma$ with a probability of 2^{-n} and $I \oplus I'^* = \gamma$ with a probability of 2^{-n}. If we take into account the difference of the I, I' pair, the regular differential $\gamma \rightarrow \delta$ with



(TYPE 1) (TYPE 2)



(TYPE 3)

Figure 3.4: Related-Key Rectangle Distinguishers (Right Quartets)

probability q for E^1 is used twice in this distinguisher. On the other hand, if we take into account the difference of the I, I'^* pair, the related-key differential $\gamma \rightarrow \delta$ with probability q^* for E^1 is used twice in this distinguisher (here, $q^* = Pr_{X,K}[E^1_K(X) \oplus E^1_{K \oplus \Delta K}(X \oplus \gamma) = \delta]$). Therefore, the expected number of right quartets is about

$$\sum_{\text{any } \beta,\gamma} \frac{(m \cdot p^*)^2}{2} \cdot 2^{-n} \cdot (q^2 + q^{*2}) = \; m^2 \cdot 2^{-n-1} \cdot \widehat{p^*}^2 \cdot (\widehat{q}^2 + \widehat{q^*}^2)\,,$$

where $\widehat{p^*} = \sqrt{\sum_\beta p^{*2}}$, $\widehat{q} = \sqrt{\sum_\gamma q^2}$ and $\widehat{q^*} = \sqrt{\sum_\gamma q^{*2}}$.

On the other hand, the expected number of right quartets for a random cipher is about $m^2 \cdot 2^{-2n}$ ($\approx \binom{m}{2} \cdot 2 \cdot 2^{-2n}$), since there are $\binom{m}{2} \cdot 2$ possible quartets and each of the pairs (C, C') and (C^*, C'^*) (or the pairs (C, C'^*) and (C^*, C')) satisfies the δ difference with probability 2^{-n}. Therefore, if $\widehat{p^*} \cdot (\frac{1}{2} \cdot (\widehat{q}^2 + \widehat{q^*}^2))^{1/2} > 2^{-n/2}$ and m is sufficiently large, we can distinguish between E and a random cipher.

In this distinguisher, we can use either regular differentials $\gamma \rightarrow \delta$ (related to the probability $\widehat{q}$) or related-key differentials $\gamma \rightarrow \delta$ (related to the probability $\widehat{q^*}$). By using both of them, we increase the probability for a random cipher to succeed. However, if we take only the maximum of $\widehat{q}$ and $\widehat{q^*}$, then the ratio of the expected number of right quartets between E and a random cipher is optimal. In this case, the expected number of right quartets for the E cipher is about $m^2 \cdot 2^{-1} \cdot 2^{-n} \cdot (\widehat{p^*} \cdot \widehat{q})^2$ or $m^2 \cdot 2^{-1} \cdot 2^{-n} \cdot (\widehat{p^*} \cdot \widehat{q^*})^2$. On the other hand, the expected number of right quartets for a random cipher is about $m^2 \cdot 2^{-1} \cdot 2^{-2n}$. Thus, $\widehat{p^*} \cdot \widehat{q} > 2^{-n/2}$ or $\widehat{p^*} \cdot \widehat{q^*} > 2^{-n/2}$ must hold for the related-key rectangle distinguisher to work.

Note that our estimated expectations are approximate values since the actual values of the expectations depend on the values of the chosen plaintexts and the used differential probabilities are average ones over the text and key.

3.4.2 Related-Key Rectangle Distinguisher of TYPE 2

If we have m_1 pairs (P, P^*) and m_2 pairs (P', P'^*) with difference α, where P and P^* are all encrypted under the key K and P' and P'^* are all encrypted under the key K', then we have about $m_1 \cdot p$ pairs together with $m_2 \cdot p$ pairs satisfying the regular differential $\alpha \rightarrow \beta$ for E^0. Similarly, we get $I \oplus I' = \gamma$ with a probability of 2^{-n} and $I \oplus I'^* = \gamma$ with a probability of 2^{-n}. Since the probability that both pairs (I, I') and (I^*, I'^*) (or both pairs (I, I'^*) and (I^*, I')) are right pairs with respect to the related key differential $\gamma \rightarrow \delta$ for E^1 is q^{*2} (here, $q^* = Pr_{X,K}[E^1_K(X) \oplus E^1_{K \oplus \Delta K'}(X \oplus \gamma) = \delta]$), the expected number of right

quartets is about

$$\sum_{\beta,\gamma}(m_1 \cdot p) \cdot (m_2 \cdot p) \cdot 2^{-n} \cdot 2 \cdot q^{*2} = m_1 \cdot m_2 \cdot 2^{-n+1} \cdot (\widehat{p} \cdot \widehat{q^*})^2 \,.$$

Since the expected number of right quartets for a random cipher is about $m_1 \cdot m_2 \cdot 2^{-2n+1}$, we can distinguish between E and a random cipher if $\widehat{p} \cdot \widehat{q^*} > 2^{-n/2}$ and $m_1,\ m_2$ are sufficiently large.

3.4.3 Related-Key Rectangle Distinguisher of TYPE 3

In order to optimize the ratio of the expected number of right quartets between E and a random cipher, we should only consider the maximum of $\widehat{q^*}$ and $\widehat{q'^*}$ in the related-key rectangle distinguisher of TYPE 3, where

$$\widehat{q^*} = (\sum_{\gamma}(Pr_{X,K}[E^1_K(X) \oplus E^1_{K \oplus \Delta K'}(X \oplus \gamma) = \delta])^2)^{1/2}, \text{and}$$
$$\widehat{q'^*} = (\sum_{\gamma}(Pr_{X,K}[E^1_K(X) \oplus E^1_{K \oplus \Delta K \oplus \Delta K'}(X \oplus \gamma) = \delta])^2)^{1/2}.$$

In our analysis, we assume $\widehat{q^*} > \widehat{q'^*}$.

To begin with, we also assume that we have m_1 pairs of (P, P^*) and m_2 pairs of (P', P'^*) with difference α, where $P,\ P^*,\ P'$ and P'^* are encrypted with the keys K, K^*, K' and K'^*, respectively. Then about $m_1 \cdot p^*$ and $m_2 \cdot p^*$ pairs will satisfy the related-key differential $\alpha \to \beta$ for E^0 under the key difference ΔK. Thus, we have about $m_1 \cdot m_2 \cdot p^{*2}$ quartets satisfying the differential conditions 1 and 2. Moreover, we get $I \oplus I' = \gamma$ with probability 2^{-n}. These assumptions enable us to obtain about $m_1 \cdot m_2 \cdot 2^{-n} \cdot p^{*2}$ quartets satisfying the differential conditions 1, 2 and 3. As stated above, the differential conditions 2 and 3 allow us to get $I^* \oplus I'^* = \gamma$ with probability 1, and each of the pairs (I, I') and (I^*, I'^*) satisfies the related-key differential $\gamma \to \delta$ for E^1 with probability q^*. Therefore, the expected number of right quartets is about

$$\sum_{\beta,\gamma} m_1 \cdot m_2 \cdot 2^{-n} \cdot p^{*2} \cdot q^{*2} = m_1 \cdot m_2 \cdot 2^{-n} \cdot (\widehat{p^*})^2 \cdot (\widehat{q^*})^2 \,.$$

For a random cipher the expected number of right quartets is about $m_1 \cdot m_2 \cdot 2^{-2n}$. Thus, $\widehat{p^*} \cdot \widehat{q^*} > 2^{-n/2}$ must hold for the related-key rectangle distinguisher to work.

3.4.4 Related-Key Boomerang Distinguishers

In order to get at least one right quartet in the related-key rectangle distinguishers, we need at least $2^{n/2}$ plaintext queries. However, under an adaptive chosen

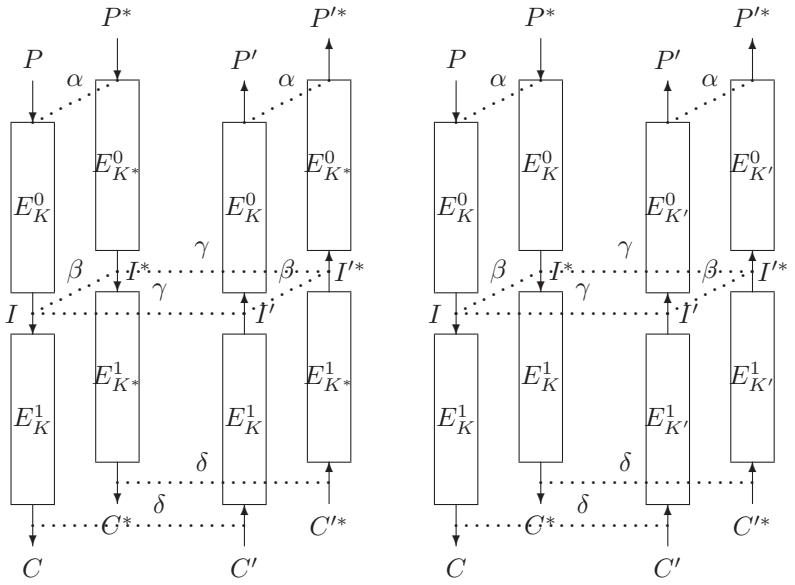
plaintext and ciphertext attack scenario we can make a related-key boomerang distinguisher which can remove the factor $2^{n/2}$ in the data requirement. As a compensation of a smaller data requirement, this attack works only in a stronger attack model; it requires access to both the encryption box and the decryption box. The related-key boomerang distinguishers based on two or four related keys work as follows.

- Choose two n-bit plaintexts P and P^* such that $P \oplus P^* = \alpha$, and obtain the corresponding ciphertexts $C = E_K(P)$ and $C^* = E_{K^*}(P^*)$, where $K \oplus K^* = \Delta K$.
- Compute other two ciphertexts $C' = C \oplus \delta$ and $C'^* = C^* \oplus \delta$, and obtain the corresponding plaintexts $P' = E_{K'}^{-1}(C')$ and $P'^* = E_{K'^*}^{-1}(C'^*)$, where $K' \oplus K'^* = \Delta K$ and $K \oplus K' = K^* \oplus K'^* = \Delta K'$.
- Check if $P' \oplus P'^* = \alpha$.

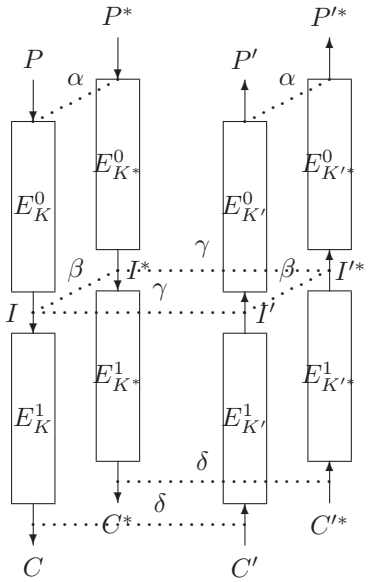
Similarly, we can classify the three types of distinguishers by the condition of the key differences ΔK and $\Delta K'$. See Fig. 3.5 for their schematic descriptions. Note that the difference between the related-key rectangle and boomerang distinguishers of the same TYPE is on the encryption and decryption process for the plaintexts (P', P'^*) and the ciphertexts (C', C'^*) in Figs. 3.4 and 3.5. In a similar way, we can analyze the three types of the related-key boomerang distinguishers. Let us consider the related-key boomerang distinguisher of TYPE 3.

The probability that $I \oplus I^* = \beta$ is p^* (in the encryption direction) and the probability that $I \oplus I' = I^* \oplus I'^* = \gamma$ is q^{*2} (in the decryption direction). Therefore, for any β and γ, $I \oplus I^* = \beta$ and $I \oplus I' = I^* \oplus I'^* = \gamma$ (as in these cases $I' \oplus I'^* = \beta$) hold with probability $p^* \cdot q^{*2}$. Since the probability of the related-key differential $\beta \rightarrow \alpha$ for $(E^0)^{-1}$ under the related-key difference ΔK is p^*, the probability that $P' \oplus P'^* = \alpha$ is $\sum_{\beta,\gamma} p^{*2} \cdot q^{*2} = \widehat{p^{*2}} \cdot \widehat{q^{*2}}$. Therefore, if we have m chosen plaintext pairs (P, P^*) with difference α and we have another m adaptively chosen ciphertext pairs (C', C'^*) such that $C' = C^* \oplus \delta$ and $C'^* = C^* \oplus \delta$, then about $m \cdot \widehat{p^*}^2 \cdot \widehat{q^*}^2$ quartets satisfy the α test. Since for a random cipher the α test holds with probability 2^{-n}, $\widehat{p^*} \cdot \widehat{q^*} > 2^{-n/2}$ must hold for the related-key boomerang distinguisher to work.

Note 1: The actual probabilities of the related-key rectangle and boomerang distinguishers are larger than those computed above, for they also encompass the case of different β and γ differences in the middle, i.e., differences β, β' and γ, γ' can be used in the middle, where $\beta \neq \beta'$, $\gamma' = \gamma \oplus \beta \oplus \beta'$. However, it is difficult to compute these actual probabilities.



(TYPE 1) (TYPE 2)



(TYPE 3)

Figure 3.5: Related-Key Boomerang Distinguishers (Right Quartets)

Note 2: In a similar way, (related-key) truncated differentials for E^0 and E^1 can be used to form related-key rectangle and boomerang distinguishers and more than four related keys can also be used in the related-key rectangle and boomerang distinguisher in which the basic idea is the same as that of the distinguishers with two or four related keys.

Chapter 4

Applications to Block Ciphers

4.1 Introduction

Cryptanalysis of SHACAL-1 and SHACAL-2. In 2004 and 2005 several important cryptanalytic articles [9, 10, 124, 125, 126, 127] have been published that demonstrate collisions for the hash functions SHA-0 and SHA-1. Especially, a new message modification technique proposed by Wang et al. greatly improves previously known cryptanalytic results on SHA-0 and SHA-1 [125, 127]. Due to the structure of SHACAL-1, differentials of SHA-1 correspond to related-key differentials of SHACAL-1. Hence, it seems natural that some of the techniques used in the new attacks on SHA-1 can be converted into a related-key attack on SHACAL-1. We show that this is indeed the case. The differentials devised in [125] can be converted into high probability related-key differentials of SHACAL-1.

After transforming the collision producing differentials into related-key differentials, we use them in a related-key rectangle attack. The resulting attack succeeds to attack the full 80-round SHACAL-1 using 4 related-keys faster than exhaustive key search. The related-key rectangle technique was used in previously published attacks on SHACAL-1 [67, 49]. The best previously known related-key rectangle attack on the cipher was applicable up to 70 rounds of SHACAL-1. Our results extend these previously known results by using improved differentials and improved attack techniques. We note that the best known attack on SHACAL-1 that does not use related keys is a differential attack on 55-round SHACAL-1 [90] and the best known attack on SHACAL-1 that uses related keys is a related-key slide attack on the full SHACAL-1 [17]. The latter attack is superior to our

Table 4.1: Key Recovery Attacks on SHACAL-1, SHACAL-2 and AES

Block Cipher	Type of Attack	#R/#K	Complexity Data / Time
SHACAL-1 (80 rounds)	Differential	49/1	2^{142}CP / $2^{496.5}$[90]
		55/1	2^{154}CC / $2^{507.3}$[90]
	Amplified Boomerang	47/1	$2^{158.5}$CP / $2^{508.4}$[69]
	Rectangle	49/1	$2^{151.9}$CP / $2^{508.5}$[13]
		51/1	$2^{153.7}$CC / $2^{503.7}$[90]
		52/1	$2^{159.7}$KP / $2^{492.7}$MA[90]
	RK Rectangle	57/2	$2^{154.8}$RK-CP / $2^{503.4}$ [67]
		59/2	$2^{149.7}$RK-CP / $2^{498.3}$ [67]
		70/4	$2^{151.8}$RK-CP / $2^{500.1}$ [49]
		80/4	**$2^{159.8}$RK-CP / $2^{419.0}$ (Sect. 4.2)**
		80/4	**$2^{153.8}$RK-CP / $2^{500.2}$ (Sect. 4.2)**
	RK Slide	80/2	2^{97}RK-CP / 2^{447} [17]
		80/4	$2^{99.6}$RK-CP / 2^{321} [17]
		80/8	$2^{101.3}$RK-CP / $2^{101.3}$ [17]
SHACAL-2 (64 rounds)	**Square-Nonlinear**	**28/1**	**2^{37}CP / $2^{494.1}$ (Sect. 4.3)**
	Impossible Differential	30/1	744CP / $2^{495.1}$ / $2^{14.5}$ [48]
	Differential-Nonlinear	**32/1**	**$2^{43.4}$CP / $2^{504.2}$ (Sect. 4.4)**
	RK Differential-Nonlinear	**35/2**	**$2^{42.3}$RK-CP / $2^{451.1}$ (Sect. 4.5)**
	RK Rectangle	37/2	$2^{233.2}$RK-CP / 2^{485} [68]
		40/2	**$2^{243.4}$RK-CP / $2^{447.4}$ (Sect. 4.6)**
		42/2	**$2^{243.4}$RK-CP / $2^{487.4}$ (Sect. 4.6)**
AES-192 (12 rounds)	Impossible Differential	7/1	2^{92}CP / 2^{186} [103]
	Square	7/1	2^{32}CP / 2^{184} [92]
	Partial Sums	7/1	$19 \cdot 2^{32}$CP / 2^{155} [36]
		7/1	$2^{128} - 2^{119}$CP / 2^{120} [36]
		8/1	$2^{128} - 2^{119}$CP / 2^{188} [36]
	RK Impossible Differential	7/2	2^{111}RK-CP / 2^{116} [55]
		8/2	2^{88}RK-CP / 2^{183} [55]
		7/32	2^{56}CP / 2^{94} [16]
		8/32	2^{116}RK-CP / 2^{134} [16]
		8/32	2^{92}RK-CP / 2^{159} [16]
		8/32	$2^{68.5}$RK-CP / 2^{184} [16]
	RK Rectangle	**8/2**	**2^{94}RK-CP / 2^{120} (Sect. 4.7)**
		8/4	**$2^{86.5}$RK-CP / $2^{86.5}$ (Sect. 4.7)**
		9/256	2^{86}RK-CP / 2^{125}[14]
		10/256	**2^{125}RK-CP / $2^{146.7}$ (Sect. 4.7)**
AES-256 (14 rounds)	Partial Sums	8/1	$2^{128} - 2^{119}$CP / 2^{204} [36]
		9/1	2^{85}CP / $5 \cdot 2^{224}$[36]
	RK Rectangle	**9/4**	**2^{99}RK-CP / 2^{120} (Sect. 4.7)**
		10/256	$2^{114.9}$RK-CP / $2^{171.8}$[14]

#R: Number of attacked rounds, #K: Number of keys, RK: Related-Key,
KP: Known Plaintext, CP: Chosen Plaintexts, CC: Chosen Ciphertexts,
MA: Memory Access, Time: Encryption units.

related-key rectangle attack which is the first known shortcut attack on the full SHACAL-1 in terms of the data/time complexity.

As for SHACAL-2, we examine its security against the square-nonlinear, differential-nonlinear, related-key differential-nonlinear and related-key rectangle attacks. We show that the square-nonlinear and differential-nonlinear attacks can be applied to 28 and 32-round SHACAL-2. The latter attack is more powerful than the previously best known attack on SHACAL-2 which does not use related keys (an impossible differential attack on 30-round SHACAL-2 [48]). We also show that the related-key differential-nonlinear and related-key rectangle attacks can be applied to 35 and 42-round SHACAL-2 with 2 related keys. The latter attack, which extends the previously known related-key rectangle attack on 37-round SHACAL-2, leads to the best known attack on SHACAL-2 which uses related keys.

Cryptanalysis of AES. This thesis examines the security of AES-192 and AES-256 against the related-key rectangle attack. We find the following new attacks: 8-round (out of 12) AES-192 with 2 or 4 related keys, 10-round (out of 12) AES-192 with 256 related keys and 9-round (out of 14) AES-256 with 4 related keys. Our attacks reduce the complexity of earlier related-key rectangle attacks presented at EUROCRYPT 2005 [15]: we present the first shortcut attack on AES-192 reduced to 10 rounds; for reduced AES-256 with 9 rounds, we decrease the required number of related keys from 256 to 4 and both the data and time complexity at the cost of a smaller number of rounds.

A comparison of the known attacks along with our new results on SHACAL-1, SHACAL-2 and AES is presented in Table 4.1. Note that our attacks presented in this chapter are not practical due to their high complexities; however, they show certificational weaknesses on the reduced or full SHACAL-1, SHACAL-2 and AES.

4.2 Related-Key Rectangle Attack on the Full 80-Round SHACAL-1

Our attack on SHACAL-1 is based on a 69-round related-key rectangle distinguisher of TYPE 3. In the attack on the full SHACAL-1, we try all the possible subkeys of the remaining 11 rounds, and decrypt all the ciphertexts. Then, the 69-round distinguisher is applied. We improve the time complexity of the attack by partially decrypting only 8 rounds, and then use the early abort approach to reduce the number of values that are decrypted through the remaining three more rounds, before the attack is applied. It is expected that for the right guess of the subkey of the last 11 rounds, the distinguisher would be more successful

than for a wrong guess. Thus, we can use this distinguisher to identify (to some extent) the right subkey.

Before describing our 69-round related-key rectangle distinguisher of SHACAL-1, we present two differential properties of SHACAL-1 that are used for computing probabilities of differentials.

4.2.1 Differential Properties of SHACAL-1

The first differential property of SHACAL-1 is derived from the combination of XOR and modular additions. Assume that $Z = X + Y$, $Z^* = X^* + Y^*$ where X, Y and X^*, Y^* are all 32-bit independent random variables with uniform distribution. Denote $Pr_{X,Y}[(X+Y)\oplus(X^*+Y^*) = \Delta Z | X \oplus X^* = \Delta X, Y \oplus Y^* = \Delta Y, \Delta Z]$ by $Pr[(\Delta X, \Delta Y) \xrightarrow{+} \Delta Z]$. Then we have the following property.

Property 1 *(Lipmaa [88]) Given three 32-bit differences ΔX, ΔY and ΔZ. If the probability $Pr[(\Delta X, \Delta Y) \xrightarrow{+} \Delta Z] > 0$, then*

$$Pr[(\Delta X, \Delta Y) \xrightarrow{+} \Delta Z] = 2^{-s},$$

where the integer s is given by $s = \#\{i | 0 \leq i \leq 30, not((\Delta X)_i = (\Delta Y)_i = (\Delta Z)_i)\}$.

The second element that affects the differential behavior of SHACAL-1 is the functions f_{xor}, f_{if} and f_{maj}. The three functions f_{xor}, f_{if} and f_{maj} operate in a bit-by-bit manner, therefore, each of them can be regarded as a Boolean function from a 3-bit input to a 1-bit output.

Property 2 *(Handschuh et al. [43]) Table 4.2 shows the distribution probability of XOR differences through the f_{xor}, f_{if} and f_{maj} functions. The first column of Table 4.2 represents the eight possible differences in x, y, z. The second column indicates the differences in the outputs of each of the three functions. In the second column, a '0' (resp. '1') means that the difference is always 0 (resp. 1), and a '0/1' means that in half of the cases, the difference is 0 and in the other half of the cases, the difference is 1.*

4.2.2 69-Round Related-Key Rectangle Distinguisher of TYPE 3

We decompose 69-round SHACAL-1 into two sub-ciphers: E^0 contains the first 34 rounds of SHACAL-1 (rounds 0-33), and E^1 contains the remaining 35 rounds (rounds 34-68).

Table 4.2: The XOR Differential Distribution Table of the f-Functions of SHACAL-1

Δx	Δy	Δz	Δf_{xor}	Δf_{if}	Δf_{maj}
0	0	0	0	0	0
0	0	1	1	0/1	0/1
0	1	0	1	0/1	0/1
1	0	0	1	0/1	0/1
0	1	1	0	1	0/1
1	0	1	0	0/1	0/1
1	1	0	0	0/1	0/1
1	1	1	1	0/1	1

We have transformed the collision producing differentials of SHA-1 presented in [125] into related-key differentials for each of the two sub-ciphers. The first related-key differential for E^0 is presented in Table 4.3. The probability of the differential is 2^{-39}. This differential includes two bits fixed in each of the plaintexts of the pair: the most significant bit of A is zero and bit 3 of A differs from bit 3 of B (by fixing these bits we can ignore probability $\frac{1}{2}$ derived from the f_{if} function for rounds 1 and 2 each). Due to the nature of the related-key rectangle attack, we can improve the probability by counting over several differentials. We have counted over differentials which have the same first 33 rounds as the differential presented in Table 4.3. The resulting probability is $\widehat{p^*} = 2^{-38.5}$ (when fixing the respective bits of the plaintext).

The second related-key differential for rounds 34-68 (E^1) is presented in Table 4.4. The probability of this differential is 2^{-39}. Again, due to the nature of the rectangle attack, we can improve the probability by counting over several differentials. We count over various similar characteristics, by changing the first round of this differential. The resulting probability is $\widehat{q^*} = 2^{-38.3}$.

Combining these two differentials leads to a 69-round related-key rectangle distinguisher with probability $2^{-160} \cdot \widehat{p^*}^2 \cdot \widehat{q^*}^2 = 2^{-313.6}$, i.e., given m related-key chosen plaintext pairs (P, P^*) and (P', P'^*) each, we expect $m^2 \cdot 2^{-160} \cdot (\widehat{p^*} \cdot \widehat{q^*})^2$ right quartets. Hence, given two sets of $2^{157.8}$ related-key chosen plaintext pairs, we expect four right rectangle quartets, while for a random cipher only $2^{-4.4}$ are expected.

Table 4.3: Related-Key Differential for Rounds 0-33 of SHACAL-1 (E^0)

Round (r)	ΔA^r	ΔB^r	ΔC^r	ΔD^r	ΔE^r	ΔK^r	Prob.
0	0	0	e_{31}	e_{31}	e_{31}	e_1	2^{-1}
1	e_1	0	0	e_{31}	e_{31}	e_6	2^{-1}
2	0	e_1	0	0	e_{31}	$e_{1,31}$	2^{-1}
3	0	0	e_{31}	0	0	e_{31}	2^{-1}
4	0	0	0	e_{31}	0	$e_{1,31}$	2^{-2}
5	e_1	0	0	0	e_{31}	$e_{6,31}$	2^{-1}
6	0	e_1	0	0	0	0	2^{-2}
7	e_1	0	e_{31}	0	0	$e_{6,31}$	2^{-2}
8	0	e_1	0	e_{31}	0	e_{31}	2^{-3}
9	e_1	0	e_{31}	0	e_{31}	e_6	2^{-2}
10	0	e_1	0	e_{31}	0	e_{31}	2^{-3}
11	e_1	0	e_{31}	0	e_{31}	e_6	2^{-2}
12	0	e_1	0	e_{31}	0	$e_{1,31}$	2^{-3}
13	0	0	e_{31}	0	e_{31}	0	2^{-1}
14	0	0	0	e_{31}	0	e_{31}	2^{-1}
15	0	0	0	0	e_{31}	e_{31}	1
16	0	0	0	0	0	0	1
17	0	0	0	0	0	0	1
18	0	0	0	0	0	0	1
19	0	0	0	0	0	0	1
20	0	0	0	0	0	0	1
21	0	0	0	0	0	0	1
22	0	0	0	0	0	0	1
23	0	0	0	0	0	0	1
24	0	0	0	0	0	0	1
25	0	0	0	0	0	0	1
26	0	0	0	0	0	e_2	2^{-1}
27	e_2	0	0	0	0	e_7	2^{-1}
28	0	e_2	0	0	0	e_2	2^{-1}
29	0	0	e_0	0	0	$e_{0,3}$	2^{-2}
30	e_3	0	0	e_0	0	$e_{0,8}$	2^{-2}
31	0	e_3	0	0	e_0	$e_{0,3}$	2^{-2}
32	0	0	e_1	0	0	$e_{1,4}$	2^{-2}
33	e_4	0	0	e_1	0	$e_{1,9}$	2^{-2}
34	0	e_4	0	0	e_1		

Table 4.4: Related-Key Differential for Rounds 34-68 of SHACAL-1 (E^1)

Round (r)	ΔA^r	ΔB^r	ΔC^r	ΔD^r	ΔE^r	$\Delta K'^r$	Prob.
34	0	e_1	e_{31}	0	$e_{30,31}$	$e_{1,30}$	2^{-2}
35	0	0	e_{31}	e_{31}	0	e_1	2^{-1}
36	e_1	0	0	e_{31}	e_{31}	e_6	2^{-1}
37	0	e_1	0	0	e_{31}	$e_{1,31}$	2^{-1}
38	0	0	e_{31}	0	0	e_{31}	1
39	0	0	0	e_{31}	0	$e_{1,31}$	2^{-1}
40	e_1	0	0	0	e_{31}	$e_{6,31}$	2^{-1}
41	0	e_1	0	0	0	0	2^{-2}
42	e_1	0	e_{31}	0	0	$e_{6,31}$	2^{-2}
43	0	e_1	0	e_{31}	0	e_{31}	2^{-3}
44	e_1	0	e_{31}	0	e_{31}	e_6	2^{-2}
45	0	e_1	0	e_{31}	0	e_{31}	2^{-3}
46	e_1	0	e_{31}	0	e_{31}	e_6	2^{-2}
47	0	e_1	0	e_{31}	0	$e_{1,31}$	2^{-3}
48	0	0	e_{31}	0	e_{31}	0	2^{-1}
49	0	0	0	e_{31}	0	e_{31}	2^{-1}
50	0	0	0	0	e_{31}	e_{31}	1
51	0	0	0	0	0	0	1
52	0	0	0	0	0	0	1
53	0	0	0	0	0	0	1
54	0	0	0	0	0	0	1
55	0	0	0	0	0	0	1
56	0	0	0	0	0	0	1
57	0	0	0	0	0	0	1
58	0	0	0	0	0	0	1
59	0	0	0	0	0	0	1
60	0	0	0	0	0	0	1
61	0	0	0	0	0	e_2	2^{-1}
62	e_2	0	0	0	0	e_7	2^{-1}
63	0	e_2	0	0	0	e_2	2^{-1}
64	0	0	e_0	0	0	$e_{0,3}$	2^{-2}
65	e_3	0	0	e_0	0	$e_{0,8}$	2^{-2}
66	0	e_3	0	0	e_0	$e_{0,3}$	2^{-2}
67	0	0	e_1	0	0	$e_{1,4}$	2^{-2}
68	e_4	0	0	e_1	0	$e_{1,9}$	2^{-2}
69	0	e_4	0	0	e_1		

4.2.3 Key Recovery Attack

The basic approach for a key recovery attack is to guess the subkey of the last 11 rounds, partially decrypt all ciphertexts, and apply the distinguisher for the remaining 69 rounds. This approach can be improved using the fact that in every round, only a small part of the intermediate value is substantially changed, while most of the value is only shifted. The attack is based on the early abort technique of [24, 36]. In this technique, once a pair/quartet does not satisfy the required differences/properties, it is excluded from further analysis.

In the description of the attack algorithm we use the following notation: Y_A denotes the value of word A in Y. Similarly, $Z_{D,E}$ denotes words D and E of Z, etc. We also denote by S' the set of possible values of ΔA^{70} given from the output difference of the second differential.

We observe that even if we partially decrypt only 8 rounds, we still have a filtering condition on the quartets: since $\Delta D^{72} = ROTL_{30}(\Delta A^{69})$ and $\Delta E^{72} = ROTL_{30}(\Delta B^{69})$, we can check whether the difference in these words D^{72} and E^{72} corresponds to the output difference in words A^{69} and B^{69} of the second differential. In addition, we observe that we can extend the second differential by a truncated differential of one additional round. There are only $324 = 2^{8.3}$ possible ΔA^{70} values in S', hence, there are only 324 possible values for ΔC^{72} in case the second differential holds.

Using these observations, we can get a filtering of $64 + 23.7 = 87.7$ bits for every pair at the end of round 71, or a filtering of 175.4 bits in total (for every quartet). Since the attack starts with $2^{315.6}$ quartets, we expect that $2^{140.3}$ quartets pass the filtering for any given subkey guess of rounds 72-79. We then guess the subkey of round 71 and compute ΔE^{71} that is equal to ΔC^{69}. From the second differential, we obtain an additional 64-bit filtering on the remaining quartets. After this filtering only $2^{76.3}$ quartets remain for each subkey guess. Then we continue by guessing the subkeys of rounds 70 and 69. As a result, the time complexity of the attack drops substantially, while the data complexity remains unchanged.

The algorithm of the attack is as follows:

1. Data Collection:

- **(a)** Ask for the encryption of $2^{157.8}$ pairs of plaintexts (P, P^*), where $P^* = P \oplus \alpha$, where P and P^* satisfy the 2-bit restrictions, and where P is encrypted under K and P^* is encrypted under K^*.
- **(b)** Ask for the encryption of $2^{157.8}$ pairs of plaintexts (P', P'^*), where $P' = P'^* \oplus \alpha$, where P' and P'^* satisfy the 2-bit restrictions, and where P' is encrypted under K' and P'^* is encrypted under K'^*.

2. Partial Decryption:

(a) For each guess of the subkey of rounds 72-79:

i. Partially decrypt all ciphertexts C, C^*, C', C'^* (under the corresponding keys).

ii. Find all pairs of (T, T'), such that $T_{C,D,E} \oplus T'_{C,D,E} \in S$, where T is the partially decrypted value of C, T' is the partially decrypted value of C' and $S = \{(x, y, z) : ROTL_{30}(x) \in S',\ ROTL_{30}\ (y) = \delta_A = 0,\ ROTL_{30}(z) = \delta_B = e_2\}$.

iii. For each such pair (T, T'), let P and P' be the corresponding plaintexts. Let $P^* = P \oplus \alpha$ and $P'^* = P' \oplus \alpha$, and let T^* and T'^* be the partially decrypted values of C^* and C'^*, respectively.

iv. If $T^*_{C,D,E} \oplus T'^*_{C,D,E} \in S$ pass the quartet (T, T^*, T', T'^*) for a further analysis.

(b) Partial Decryption of Round 71: For each guess of the subkey of round 71:

i. Partially decrypt all the remaining quartets (under the corresponding keys) and denote the resulting intermediate values by (U, U^*, U', U'^*).

ii. For each of the remaining quartets, check whether $U_E \oplus U'_E = \delta_C = 0$ and discard all the quartets that do not satisfy the equation.

iii. For each of the remaining quartets, check whether $U^*_E \oplus U'^*_E = \delta_C = 0$ and discard all the quartets that do not satisfy the equation.

(c) Partial Decryption of Round 70: For each guess of the subkey of round 70:

i. Partially decrypt all the remaining quartets (under the corresponding keys) and denote the resulting intermediate values by (V, V^*, V', V'^*).

ii. For each of the remaining quartets, check whether $V_E \oplus V'_E = \delta_D = 0$ and discard all the quartets that do not satisfy the equation.

iii. For each of the remaining quartets, check whether $V^*_E \oplus V'^*_E = \delta_D = 0$ and discard all the quartets that do not satisfy the equation.

(d) Partial Decryption of Round 69: For each guess of the subkey of round 69:

i. Partially decrypt all the remaining quartets (under the corresponding keys) and denote the resulting intermediate values by (W, W^*, W', W'^*).

ii. For each of the remaining quartets, check whether $W_E \oplus W'_E = \delta_E = e_1$ and discard all the quartets that do not satisfy the equation.

iii. For each of the remaining quartets, check whether $W^*_E \oplus W'^*_E = \delta_E = e_1$ and discard all the quartets that do not satisfy the equation.

iv. Pass all the remaining quartets to further analysis.

(e) Further Analysis: If for this subkey guess only one quartet is suggested (or no quartets are suggested) discard the subkey guess. If the subkey is not discarded, exhaustively search all possible values for the remaining 160 subkey bits for the correct key.

The time complexity of Step 1 is $2^{159.8}$ encryptions. The average time complexity of Step 2(a) is $\frac{8}{80} \cdot 2^{256} \cdot 2^{158.8} \cdot \frac{1}{2} = 2^{410.5}$ SHACAL-1 encryptions. Steps 2(b)-2(e) are repeated for each subkey guess, i.e., 2^{255} times on average. For a given subkey guess, Step 2(b) consists of $2^{141.3} \cdot 2^{32}$ partial decryptions of one SHACAL-1 round. This is equivalent to $2^{141.3} \cdot 2^{32} \cdot \frac{1}{80} = 2^{167.0}$ full SHACAL-1 encryptions. Thus, the total expected time complexity of Step 2(b) is about $2^{255} \cdot 2^{167.0} = 2^{422.0}$ SHACAL-1 encryptions. The time complexities of the other steps are relatively smaller.

Using a more delicate analysis we can reduce the time complexity of Step 2(b) by a factor of 2^3: in Steps 2(a) and 2(b), we can check the S and δ_C test with the actual values of the subkey of rounds 71-79 except for the most significant bits (MSBs) of the subkey of rounds 71-73. However, we need to guess their MSBs after Step 2(b) to check the δ_D and δ_E tests. Hence, the total time complexity of the attack is decreased to $2^{419.0}$ SHACAL-1 encryptions. The data complexity of this attack is $2^{159.8}$ related-key chosen plaintexts encrypted under four keys. The memory requirement of the attack is about $2^{159.8}$ memory blocks of 160 bits, required for storing the large amount of data.

We note that a different approach may be used in our attack. We can remove the last three rounds of the second differential to increase its probability by a factor of 2^6, resulting in a 66-round related-key rectangle distinguisher with probability $2^{-160} \cdot \widehat{p^*}^2 \cdot \widehat{q^*}^2 = 2^{-301.6}$. The resulting distinguisher requires $2^{151.8}$ related-key chosen plaintext pairs (P, P^*) and (P', P'^*) each to produce four right plaintext quartets (while for a random cipher about $2^{-16.4}$ quartets that satisfy the rectangle conditions are expected). Then, we apply partial decryptions of rounds 69-79, 68, 67 and 66 in Steps 2(a), 2(b), 2(c) and 2(d), respectively, and then run the final exhaustive search for the remaining 64-bit keys in Step 2(e).

The time complexity of Step 2(a) in this case is $2^{152.8+352} \cdot \frac{11}{80} \cdot \frac{1}{2} = 2^{500.9}$ SHACAL-1 encryptions on average. In this attack we can derive the set S in Step 2-(a) for the filtering of quartets, which has $2^{70.8}$ elements, and thus the

number of remaining quartets after this step is about $(2^{151.9}.2^{-160+70.8})^2 = 2^{125.2}$. It follows that Step 2(b) takes about $2^{126.2} \cdot 2^{352+32} \cdot \frac{1}{80} \cdot \frac{1}{2} = 2^{502.9}$ SHACAL-1 encryptions on average. Compared to Steps 2(a) and 2(b), the following steps have quite small time complexities. With a similar delicate analysis, we can also reduce the time complexity of Steps 2(a) and 2(b) by factors of 2^2 and 2^3, respectively. Hence, this full-round attack on SHACAL-1 works with a data complexity of $2^{153.8}$ related-key chosen plaintexts encrypted under four related keys and with a time complexity of $2^{498.9}+2^{499.9} = 2^{500.2}$ SHACAL-1 encryptions.

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

In the next four sections, we apply our combined attacks to reduced versions of SHACAL-2. The attacks described in the first three sections exploit a 3-round nonlinear relation of the attacks in [48]. The details of the 3-round nonlinear relation are as follows.

The value h_0^r can be represented as the output of a nonlinear function $NF(A^{r+3}, B^{r+3}, \cdots, H^{r+3}, Cst^r, Cst^{r+1}, Cst^{r+2}, K^r, K^{r+1}, K^{r+2})$, denoted NF^{r+3}, where $0 \le r \le 61$. Note that x_i^r denotes the *i-th* bit of the input word X^r to round r.

$$\begin{aligned} h_0^r \;=\; & c_0^{r+3} \oplus d_2^{r+3} \oplus d_{13}^{r+3} \oplus d_{22}^{r+3} \oplus (d_0^{r+3} \& (e_0^{r+3} \oplus t_{1,0}^{r+3})) \oplus (d_0^{r+3} \& (f_0^{r+3} \oplus t_{1,0}^{r+2})) \\ \oplus \quad & ((e_0^{r+3} \oplus t_{1,0}^{r+3}) \& (f_0^{r+3} \oplus t_{1,0}^{r+2})) \oplus h_6^{r+3} \oplus h_{11}^{r+3} \oplus h_{25}^{r+3} \\ \oplus \quad & (h_0^{r+3} \& h_0^{r+2}) \oplus ((\neg h_0^{r+3}) \& h_0^{r+1}) \oplus cst_0^r \oplus k_0^r \,. \end{aligned}$$

The values h_0^{r+1}, $t_{1,0}^{r+2}$, h_0^{r+2} and $t_{1,0}^{r+3}$ in the above equation are represented as follows.

$$\begin{cases} h_0^{r+1} = t_{1,0}^{r+2} \oplus g_6^{r+3} \oplus g_{11}^{r+3} \oplus g_{25}^{r+3} \oplus (g_0^{r+3} \& h_0^{r+3}) \oplus ((\neg g_0^{r+3}) \& h_0^{r+2}) \oplus cst_0^{r+1} \\ \qquad\qquad \oplus\ k_0^{r+1} \\ t_{1,0}^{r+2} = b_0^{r+3} \oplus c_2^{r+3} \oplus c_{13}^{r+3} \oplus c_{22}^{r+3} \oplus (c_0^{r+3} \& d_0^{r+3}) \oplus (c_0^{r+3} \& (e_0^{r+3} \oplus t_{1,0}^{r+3})) \\ \qquad\qquad \oplus\ (d_0^{r+3} \& (e_0^{r+3} \oplus t_{1,0}^{r+3})) \\ h_0^{r+2} = t_{1,0}^{r+3} \oplus f_6^{r+3} \oplus f_{11}^{r+3} \oplus f_{25}^{r+3} \oplus (f_0^{r+3} \& g_0^{r+3}) \oplus ((\neg f_0^{r+3}) \& h_0^{r+3}) \oplus cst_0^{r+2} \\ \qquad\qquad \oplus\ k_0^{r+2} \\ t_{1,0}^{r+3} = a_0^{r+3} \oplus b_2^{r+3} \oplus b_{13}^{r+3} \oplus b_{22}^{r+3} \oplus (b_0^{r+3} \& c_0^{r+3}) \oplus (b_0^{r+3} \& d_0^{r+3}) \oplus (c_0^{r+3} \& d_0^{r+3}) \end{cases}$$

In this section, we first describe a 13-round square-nonlinear distinguisher and then exploit it to attack 28-round SHACAL-2.

4.3.1 13-Round Square-Nonlinear Distinguisher

Our 13-round square-nonlinear distinguisher of SHACAL-2 first applies a 10-round square characteristic for rounds 0-9 and then concatenates to this square characteristic the foregoing 3-round nonlinear relation for rounds 10-12. Our 10-round square characteristic is built based on the structual property of SHACAL-2 that most of words (six out of all the eight words) are just shifted through each round. It starts from collecting a well-chosen set of plaintexts. If a set of 2^{32} plaintexts $P_i \in (\mathbf{0},\mathbf{0},PS,CS,\mathbf{1},CS,-PS,CS)$ is inserted to SHACAL-2, where $0 \leq i \leq 2^{32}-1$, $\mathbf{0}$ and $\mathbf{1}$ represent constant sets composed of the 32-bit words `0x00000000` and `0xffffffff`, respectively, then the least significant bits of the eighth words after 10 rounds are balanced, i.e., $\bigoplus_{i=0}^{2^{32}-1} h_{i,0}^{10} = 0$. See Table 4.5 for more details of our 10-round square characteristic (the notation used in Table 4.5 listed in Sect. 1.6.1). To this square characteristic we concatenate the 3-round nonlinear relation to obtain the following 13-round square-nonlinear distinguisher

$$\bigoplus_{i=0}^{2^{32}-1} NF_i^{13} = 0 \tag{4.1}$$

with probability 1, where its input is a set of 2^{32} plaintexts $P_i \in (\mathbf{0},\mathbf{0},PS,CS,\mathbf{1},CS,-PS,CS)$.

Table 4.5: Square Characteristic for Rounds 0-9 of SHACAL-2 (E^0)

Round (r)	A^r	B^r	C^r	D^r	E^r	F^r	G^r	H^r
0	$\mathbf{0}$	$\mathbf{0}$	PS	CS	$\mathbf{1}$	CS	$-PS$	CS
1	CS	$\mathbf{0}$	$\mathbf{0}$	PS	CS	$\mathbf{1}$	CS	$-PS$
2	PS	CS	$\mathbf{0}$	$\mathbf{0}$	CS	CS	$\mathbf{1}$	CS
3	BS_0	PS	CS	$\mathbf{0}$	CS	CS	CS	$\mathbf{1}$
4	?	BS_0	PS	CS	CS	CS	CS	CS
5	?	?	BS_0	PS	CS	CS	CS	CS
6	?	?	?	BS_0	PS	CS	CS	CS
7	?	?	?	?	BS_0	PS	CS	CS
8	?	?	?	?	?	BS_0	PS	CS
9	?	?	?	?	?	?	BS_0	PS
10	?	?	?	?	?	?	?	BS_0

4.3.2 Key Recovery Attack

The 13-round square-nonlinear distinguisher is used to attack 28-round SHACAL-2. We first encrypt plaintext sets of the form ($\mathbf{0}$,$\mathbf{0}$,PS,CS,$\mathbf{1}$,CS,$-PS$, CS) to obtain the corresponding ciphertext sets and then we partially decrypt the ciphertext sets from round 11 to round 27 with a guessed key. If the guessed key is the right one, all the decrypted ciphertext sets will meet Eq. (4.1);[1] otherwise, each of the decrypted ciphertext sets will meet Eq. (4.1) with probability 1/2. It follows that each of the plaintext sets enables to reduce a half of the subkey space for rounds 11-27. The attack procedure is as follows.

1. Choose 32 plaintext sets of the form ($\mathbf{0}$,$\mathbf{0}$,PS,CS,$\mathbf{1}$,CS,$-PS$,CS). Request the corresponding ciphertext sets.
2. Guess a 463-bit key K^{27},K^{26},$\cdots$, K^{16}, $k_0^{15}, k_1^{15}, \cdots, k_{25}^{15}$, $k_0^{14}, k_1^{14}, \cdots, k_{25}^{14}$, $k_0^{13}, k_1^{13}, \cdots, k_{24}^{13}$, k_0^{12} and k_0^{11} (note that it is sufficient to guess this 463-bit subkey pair for computing the value ΔNF^{13} from a given ciphertext pair of 28-round SHACAL-2).
3. For each of the ciphertext sets, do a partial decryption using the guessed key, and if the ciphertext set does not satisfy Eq. (4.1), then go to Step 2. If all the ciphertext sets satisfy Eq. (4.1), then keep the guessed key.
4. For the suggested key, do an exhaustive search for the 49-bit remaining keys using trial encryption. If a 512-bit key is suggested, output the key as a master key of 28-round SHACAL-2. Otherwise, go to Step 2.

The data complexity of this attack is 2^{37} chosen plaintexts, and the memory requirements are 2^{42} bytes. The time complexity of Step 1 (the data collecting step) is 2^{37} 28-round SHACAL-2 encryptions, and the time complexity of Step 3 is $\frac{1}{2} \cdot \frac{15}{28} \cdot (2^{463} \cdot 2^{32} + 2^{462} \cdot 2^{32} + \cdots + 2^{432} \cdot 2^{32}) = 2^{494.1}$ 28-round SHACAL-2 encryptions on average. Since the expected number of the keys suggested in Step 3 is about $2^{463} \cdot 2^{-32} = 2^{431}$, the time complexity of Step 4 is about 2^{480} 28-round SHACAL-2 encryptions. Thus, the total time complexity of this attack is about $2^{494.1}$ 28-round SHACAL-2 encryptions.

4.4 Differential-Nonlinear Attack on 32-Round SHACAL-2

In this section, we describe a longer distinguisher than the 13-round square-nonlinear distinguisher by using a probability less than one. Firstly, we construct

[1] In order to check Eq. (4.1) for each ciphertext set, we need to guess some bits of the key for rounds 11 to 27.

a 14-round truncated differential with probability $2^{-18.7}$ and then concatenate the 3-round nonlinear relation to this differential to make a 17-round differential-nonlinear distinguisher with probability $\frac{1}{2} + 2^{-19.7}$. Secondly, we show how to exploit this 17-round differential-nonlinear distinguisher to devise a key recovery attack on 32-round SHACAL-2.

Before describing our 14-round truncated differential, we present differential properties of SHACAL-2 that are used for computing the probability of the differential.

4.4.1 Differential Properties of SHACAL-2

As in SHACAL-1, the differential properties in SHACAL-2 are derived from the use of both XOR and modular additions (Property 1), and of the functions Ch and Maj (Property 3). In addition to these properties, Property 4 is also used for computing the probability of our SHACAL-2 differential. In Property 4, $e_{i,\sim}$ represents a 32-bit word that has 1 in the position of bit i, arbitrary values in the position of bits $(i+1)$-31, and $0's$ in the position of the other bits and z_k represents a 32-bit word that has 0 in the position of bit k, and arbitrary values in the positions of the other bits (cf. Sect. 1.6.1).

Property 3 *The Ch and Maj functions have the same XOR differential distributions as the f_{if} and f_{maj} functions in Table 4.2, respectively.*

Property 4 *If $X \oplus X^* = e_{i,\sim}$, $Y \oplus Y^* = e_{j,\sim}$ and $i > j$, then $Z \oplus Z^* = e_{j,\sim}$. Note that if $Z \oplus Z^* = e_{j,\sim}$, then $Z \oplus Z^* = z_k$ where $0 \leq k < j$.*

4.4.2 17-Round Differential-Nonlinear Distinguisher

We first construct a 14-round truncated differential from rounds r to $r+13$. For the sake of clarity, we consider the case $r = 0$, which will be used in our attack. As in the 10-round square characteristic, our 14-round truncated differential is also built based on the structual property that six out of all the eight words are just shifted through each round. Let a plaintext pair $P = (A, B, C, D, E, F, G, H)$, $P^* = (A^*, B^*, C^*, D^*, E^*, F^*, G^*, H^*)$ have a difference $(0, 0, e_{M_1}, 0, 0, e_{31}, e_{M_2}, 0)$, where $M_1 = \{9, 18, 29\}$ and $M_2 = \{6, 9, 18, 20, 25, 29\}$, and assume that some bits of the P, P^* pair are fixed as in Eq. (4.2) (by fixing these bits we can ignore differential probabilities derived from Ch and Maj for the first few rounds). Then the least significant bit of the output difference in the eighth word after 14 rounds, Δh_0^{14} is 0 with probability 2^{-22}. See Table 4.6 for more details of the differential. It is easy to check the probabilities depicted

in Table 4.6 by using the differential properties of SHACAL-2.

$$\begin{array}{llll} a_9 = b_9, & a_{18} = b_{18}, & a_{29} = b_{29}, & a_{31} = b_{31}, \\ e_6 = 1, & e_9 = 1, & e_{18} = 1, & e_{20} = 1, \\ e_{25} = 1, & e_{29} = 1, & e_{31} = 1\,. & \end{array} \tag{4.2}$$

Table 4.6: Truncated Differential for Rounds 0-13 of SHACAL-2 (E^0)

r	ΔA^r	ΔB^r	ΔC^r	ΔD^r	ΔE^r	ΔF^r	ΔG^r	ΔH^r	Prob.
0	0	0	e_{M_1}	0	0	e_{31}	e_{M_2}	0	1
1	e_{31}	0	0	e_{M_1}	e_{31}	0	e_{31}	e_{M_2}	2^{-10}
2	0	e_{31}	0	0	0	e_{31}	0	e_{31}	2^{-2}
3	0	0	e_{31}	0	0	0	e_{31}	0	2^{-2}
4	0	0	0	e_{31}	0	0	0	e_{31}	1
5	e_{31}	0	0	0	0	0	0	0	2^{-4}
6	e_{M_1}	e_{31}	0	0	0	0	0	0	1
7	z_0	e_{M_1}	e_{31}	0	0	0	0	0	1
8	?	z_0	e_{M_1}	e_{31}	0	0	0	0	1
9	?	?	z_0	e_{M_1}	e_{31}	0	0	0	2^{-4}
10	?	?	?	z_0	$e_{M_3,\sim}$	e_{31}	0	0	1
11	?	?	?	?	z_0	$e_{M_3,\sim}$	e_{31}	0	1
12	?	?	?	?	?	z_0	$e_{M_3,\sim}$	e_{31}	1
13	?	?	?	?	?	?	z_0	$e_{M_3,\sim}$	1
14	?	?	?	?	?	?	?	z_0	

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

Improvement on the Probability. To combine the 14-round differential with the 3-round nonlinear relation, we need only the value Δh_0^{14} of this differential. Thus, we can increase the above differential probability 2^{-22} by taking into account a variety of truncated differentials of which the values ΔH^{14} are of the form z_0. In order to improve the differential probability, we count over a variety of truncated differentials which have the same first 9 rounds in the 14-round truncated differential described in Table 4.6. Table 4.7 presents some of these differentials. Based on these results we can increase the differential probability up to $2^{-18.7}$ ($\approx 1{\cdot}2^{-22}{+}4{\cdot}2^{-23}{+}9{\cdot}2^{-24}{+}16{\cdot}2^{-25}{+}16{\cdot}2^{-26}{+}42{\cdot}2^{-27}{+}51{\cdot}2^{-28}$). Thus, we have a 14-round truncated differential (which includes a small portion of truncated differentials) with a probability of approximately $2^{-18.7}$.

We use our 14-round truncated differential to build a distinguisher with a probability of $\frac{1}{2} + 2^{-19.7}$ ($= 2^{-18.7} + \frac{1}{2} \cdot (1 - 2^{-18.7})$). That is, if the plaintext

Table 4.7: Possible ΔE^{10} Values for the 14-Round Truncated Differential with the Respective Probabilities in SHACAL-2

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

pairs P, P^* have difference $(0, 0, e_{M_1}, 0, 0, e_{31}, e_{M_2}, 0)$ and meet Eq. (4.2), then $h_0^{14} = h_0^{*14}$ with a probability of $\frac{1}{2} + 2^{-19.7}$. This approximation assumes that the behavior of the remaining fraction of $1 - 2^{-18.7}$ of the pairs follows the uniform distribution. In order to verify the probability $\frac{1}{2} + 2^{-19.7}$ we performed 10 simulations using 2^{34} plaintext pairs each (we used different random keys and different plaintext pairs in each of the simulations). While our estimation in the 2^{34} plaintext pairs is $2^{33} + 20170$ $(= 2^{34} \cdot (\frac{1}{2} + 2^{-19.7}))$, we obtained the values $2^{33} + 153189, 2^{33} + 159168, 2^{33} + 161745, 2^{33} + 168761, 2^{33} + 173142, 2^{33} + 175476, 2^{33} + 177866, 2^{33} + 196441, 2^{33} + 197654, 2^{33} + 217151$ from our simulations. Our simulations show that the probability of the 14-round distinguisher is higher than our estimation $\frac{1}{2} + 2^{-19.7}$. This difference is due to the fact that our estimation only considers a small portion of truncated differentials with high probabilities for which the values ΔH^{14} are of the form z_0. Thus, we conclude that the actual probability of our 14-round distinguisher is at least $\frac{1}{2} + 2^{-19.7}$.

To this distinguisher we concatenate the 3-round nonlinear relation with probability 1. Since given the P, P^* pairs it holds that $h_0^{14} = h_0^{*14}$ with a probability of approximately $\frac{1}{2} + 2^{-19.7}$, we have the following 17-round differential-nonlinear distinguisher

$$NF^{17} = NF^{*17} \tag{4.3}$$

with a probability of approximately $\frac{1}{2} + 2^{-19.7}$, where its input is a plaintext pair that satisfies difference $(0, 0, e_{M_1}, 0, 0, e_{31}, e_{M_2}, 0)$ and Eq. (4.2).

4.4.3 Key Recovery Attack

We present a method to use the 17-round distinguisher to find a master key of 32-round SHACAL-2. First of all, we collect $O(|p - 1/2|^{-2})$ plaintext pairs with difference $(0, 0, e_{M_1}, 0, 0, e_{31}, e_{M_2}, 0)$, where p is the probability of the 17-round

differential-nonlinear distinguisher. Second, we encrypt them to obtain the corresponding ciphertext pairs and partially decrypt each ciphertext pair with a guessed key (for rounds 15 to 31) to check Eq. (4.3). If the number of the decrypted ciphertext pairs that satisfy Eq. (4.3) is larger than an appropriate threshold, then we bring the guessed key for further analysis (exhaustive search step). The details of the attack are as follows.

1. Choose $2^{42.4}$ ($= 2^3 \cdot (2^{-19.7})^{-2}$) plaintext pairs with the difference $(0, 0, e_{M_1}, 0, 0, e_{31}, e_{M_2}, 0)$ and the conditions of Eq. (4.2). Request the corresponding ciphertext pairs.

2. Guess a 463-bit key K^{31},K^{30},$\cdots$, K^{20}, $k_0^{19}, k_1^{19}, \cdots, k_{25}^{19}$, $k_0^{18}, k_1^{18}, \cdots, k_{25}^{18}$, $k_0^{17}, k_1^{17}, \cdots, k_{24}^{17}$, k_0^{16}, and k_0^{15}.

3. For each of the ciphertext pairs, do a partial decryption using the guessed key, and check Eq. (4.3). If the number of ciphertext pairs satisfying Eq. (4.3) is greater than or equal to $2^{41.4} + 2^{22}$, then keep the guessed key. Otherwise, go to Step 2.

4. For the suggested key, do an exhaustive search for the 49 remaining key bits using trial encryption (for the suggested key, we use one or two known plaintext/ciphertext pairs for the trial encryption). If a 512-bit key is suggested, output the key as a master key of 32-round SHACAL-2. Otherwise, go to Step 2.

The data complexity of this attack is $2^{43.4}$ chosen plaintexts. The memory requirements of this attack are dominated by the memory for ciphertext pairs, so this attack requires about $2^{43.4} \cdot 32 = 2^{48.4}$ memory bytes.

The time complexity of Step 1 (the data collecting step) is $2^{43.4}$ 32-round SHACAL-2 encryptions, and the average time complexity of Step 3 is $\frac{1}{2} \cdot \frac{15}{32} \cdot 2^{43.4} \cdot 2^{463} = 2^{504.2}$ 32-round SHACAL-2 encryptions (the factor $\frac{1}{2}$ corresponds to the average fraction of 463-bit keys which are tested in Step 3). In order to estimate the number of 463-bit keys which pass the test of Step 3, we use the following statistical method. For a wrong key the value NF^{17} according to each ciphertext behaves randomly. It implies that on average half of the ciphertext pairs satisfy $NF^{17} = NF^{*17}$ for a wrong key. Hence, in the case of a wrong key, the number of ciphertext pairs satisfying $NF^{17} = NF^{*17}$ is a binomial random variable $X \sim Bin(2^{42.4}, \frac{1}{2})$. Since this distribution can be approximated by the normal distribution, i.e., $X \sim N(\mu, \sigma^2)$ where $\mu = 2^{41.4}$ and $\sigma^2 = 2^{40.4}$, equivalently Z ($= \frac{X-\mu}{\sigma}$) $\sim N(0, 1)$ (due to the fact that the number of trials ($= 2^{42.4}$) is large and the success rate for each trial is $\frac{1}{2}$), it is easy to see that $Pr[X \geq 2^{41.4} + 2^{22}] = Pr[Z \geq 3.5813] \approx 2^{-12.7}$. It follows that the average number of 463-bit wrong keys that pass the test of Step 3 is about

$1/2 \cdot 2^{463} \cdot 2^{-12.7} = 2^{449.7}$. (Note that the number of the suggested keys in Step 3 is more than our estimation, because some special wrong keys proposed in [48] produce a non-random value of ΔNF^{17}. However the number of such keys is less than 2^{83}. Thus, the expected number of wrong keys which are suggested in Step 3 is less than $2^{449.7} + 2^{83}$.) So the time complexity of Step 4 is about $2^{449.7} \cdot 2^{49} = 2^{498.7}$ 32-round SHACAL-2 encryptions, and thus the total time complexity of this attack is about $2^{504.2}$ 32-round SHACAL-2 encryptions.

Using the above analysis for the right key with $X \sim Bin(2^{42.4}, \frac{1}{2} + 2^{-19.7})$ we can check the probability that the right key passes the test of Step 3 is about 0.98. Therefore, the success rate of this attack is about 0.98.

Note: Our attack algorithm can be converted into the key ranking algorithm presented in [117]. That is, instead of keeping the keys whose counters are greater than or equal to $2^{41.4} + 2^{22}$ in Step 3, we can keep the $2^{450.3}$ $(= 2^{463} \cdot 2^{-12.7})$ keys with counters greater than those of the other $(2^{462} - 2^{450.3})$ keys. By the order statistics presented in [117] the success rate of the key ranking algorithm is 0.98, which is the same as that of our attack algorithm. However, the key ranking algorithm requires a number of memory bytes for all possible 2^{463} keys.

4.5 Related-Key Differential-Nonlinear Attack on 35-Round SHACAL-2

In this attack, we show how to extend the previous 14-round differential to a 25-round related-key differential, which we combine with the 3-round nonlinear relation to devise a 28-round related-key differential-nonlinear distinguisher of SHACAL-2. Finally, we use it to present a key recovery attack on 35-round SHACAL-2.

4.5.1 28-Round Related-Key Differential-Nonlinear Distinguisher

As described in Sect. 2.1.2, the key scheduling algorithm of SHACAL-2 is based on a linear feedback shift register. However, this key scheduling algorithm has a slow difference propagation for the first few round keys. That is, in case the related keys are identical except for the sixth round key K^6, the expanded round keys $K^{16}, K^{17}, \cdots, K^{20}$ have all zero differences and K^{21}, K^{22} have the $e_{13,\sim}$ and the e_{31} differences, respectively. This difference propagation of related keys enables us to find a 25-round related-key truncated differential with a high probability. Namely, we can construct a 25-round related-key truncated differential $\alpha \rightarrow \beta$ for rounds 0 to 24 (E^0) with probability 2^{-16}, where

$\alpha = (0, e_{31}, 0, 0, e_{6,20,25}, 0, 0, e_{9,13,19})$ and $\beta = (?, ?, ?, e_{13,\sim}, ?, ?, ?, e_{13,\sim})$. See Table 4.8 and Eq. (4.4) for the details of this differential. Note that this related-key truncated differential requires plaintext pairs (P, P^*) with 4-bit fixed values in Eq. (4.4).

$$a_{31} = c_{31}, \quad f_6 = g_6, \quad f_{20} = g_{20}, \quad f_{25} = g_{25}\,. \tag{4.4}$$

Table 4.8: Related-Key Differential for Rounds 0-24 of SHACAL-2 (E^0)

r	ΔA^r	ΔB^r	ΔC^r	ΔD^r	ΔE^r	ΔF^r	ΔG^r	ΔH^r	ΔK^r	Prob.
0	0	e_{31}	0	0	e_{M_1}	0	0	e_{M_2}	0	2^{-3}
1	0	0	e_{31}	0	0	e_{M_1}	0	0	0	2^{-4}
2	0	0	0	e_{31}	0	0	e_{M_1}	0	0	2^{-3}
3	0	0	0	0	e_{31}	0	0	e_{M_1}	0	2^{-4}
4	0	0	0	0	0	e_{31}	0	0	0	2^{-1}
5	0	0	0	0	0	0	e_{31}	0	0	2^{-1}
6	0	0	0	0	0	0	0	e_{31}	e_{31}	1
7	0	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
20	0	0	0	0	0	0	0	0	0	1
21	0	0	0	0	0	0	0	0	$e_{13,\sim}$	1
22	$e_{13,\sim}$	0	0	0	$e_{13,\sim}$	0	0	0	e_{31}	1
23	?	$e_{13,\sim}$	0	0	?	$e_{13,\sim}$	0	0	?	1
24	?	?	$e_{13,\sim}$	0	?	?	$e_{13,\sim}$	0	?	1
25	?	?	?	$e_{13,\sim}$	?	?	?	$e_{13,\sim}$		

$M_1 = \{6, 20, 25\}$, $M_2 = \{9, 13, 19\}$

As in the previous attack, this 25-round related-key truncated differential works as a distinguisher with a probability of $\frac{1}{2} + 2^{-17}$ $(= 2^{-16} + \frac{1}{2} \cdot (1 - 2^{-16}))$. The converted distinguisher follows: if the plaintext pairs P, P^* have difference $(0, e_{31}, 0, 0, e_{6,20,25}, 0, 0, e_{9,13,19})$ and meet Eq. (4.4), then $h_0^{25} = h_0^{*25}$ with a probability of $\frac{1}{2} + 2^{-17}$. In order to check this probability, we also performed a series of 5 simulations using 2^{34} plaintext pairs each (for any two simulations we used different random related keys and different plaintext pairs) and we found that each of our simulations follows the estimated probability (since the 25-round distinguisher has a probability of $\frac{1}{2} + 2^{-17}$, we expect about $2^{33} + 131072$ $(= 2^{34} \cdot (\frac{1}{2} + 2^{-17}))$ pairs out of 2^{34} plaintext pairs which satisfy $h_0^{25} = h_0^{*25}$; in our 5 simulations, we obtained the following number: $2^{33} + 128629, 2^{33} + 130921, 2^{33} + 138897, 2^{33} + 143916, 2^{33} + 145975$).

As mentioned before, we concatenate the 3-round nonlinear relation to the above distinguisher in order to obtain a stronger distinguisher. Since given the P, P^* pairs, $h_0^{25} = h_0^{*25}$ with a probability of approximately $\frac{1}{2} + 2^{-17}$, we have the equation $NF^{28} = NF^{*28}$ with the same probability. Equivalently, we obtain

the following equation

$$MNF^{28} = MNF^{*28} \tag{4.5}$$

with a bias of approximately 2^{-17}, where $MNF^{28} = NF^{28} \oplus cst_0^{25} \oplus k_0^{25}$. Thus, we have a 28-round related-key differential-nonlinear distinguisher with a bias of approximately 2^{-17}.

4.5.2 Key Recovery Attack

We assume that 35-round SHACAL-2 uses related keys with difference $(0,0,0,0,0,0,e_{31},0,0,0,0,0,0,0,0,0)$. In our attack, we apply our 28-round distinguisher to retrieve the master key pair of 35-round SHACAL-2. The basic idea of this attack is the same as that of the differential-nonlinear attack on 32-round SHACAL-2. The attack procedure is as follows.

1. Prepare 5 pools of 2^{39} plaintext pairs $(P_{i,j}, P^*_{i,j})$, $i = 0, 1, \cdots, 4$, $j = 0, 1, \cdots, 2^{39}-1$, that have difference α and meet Eq. (4.4). Note that each $P_{i,j}$ is encrypted using a key K, and each $P^*_{i,j}$ is encrypted using a key K^* where K and K^* have difference $(0,0,0,0,0,0,e_{31},0,0,0,0,0,0,0,0,0)$. Encrypt all these plaintext pairs to get the 5 pools of 2^{39} ciphertext pairs $(C_{i,j}, C^*_{i,j})$.

2. Guess a 207-bit subkey pair (sk, sk^*). A subkey sk represents $K^{34}, K^{33}, K^{32}, K^{31}, k_0^{30}, k_1^{30}, \cdots, k_{25}^{30}, k_0^{29}, k_1^{29}, \cdots, k_{25}^{29}, k_0^{28}, k_1^{28}, \cdots, k_{24}^{28}, k_0^{27}, k_0^{26}$ and the other subkey sk^* represents $K^{*34}, K^{*33}, K^{*32}, K^{*31}, k_0^{*30}, k_1^{*30}, \cdots, k_{25}^{*30}, k_0^{*29}, k_1^{*29}, \cdots, k_{25}^{*29}, k_0^{*28}, k_1^{*28}, \cdots, k_{24}^{*28}, k_0^{*27}, k_0^{*26}$.

3. For $i = 0$ to 4 do the following :

   (a) Partially decrypt all 2^{39} ciphertexts $C_{i,j}$ (resp. $C^*_{i,j}$) using the sk subkey (resp. the sk^* subkey), and check Eq. (4.5). If the number of ciphertext pairs satisfying Eq. (4.5) is greater than $2^{38} - 2^{21.6}$ and less than $2^{38} + 2^{21.6}$ (for any of i), then go to Step 2.

4. For the suggested subkey sk, do an exhaustive search for the 305-bit remaining keys using trial encryption (for the suggested subkey sk, we use two known plaintext and ciphertext pairs for the trial encryption). If a 512-bit key k' is suggested, output the k' key as a master key of 35-round SHACAL-2. In this case, we also output the key $k' \oplus (0,0,0,0,0,0,e_{31},0,0,0,0,0,0,0,0,0)$ as the related master key of 35-round SHACAL-2. Otherwise, go to Step 2. In case 207-bit subkey pairs (sk, sk^*) are all tested and there does not exist a suggested key k', we stop this algorithm without output.

The data complexity of this attack is about $2^{42.3}$ ($\approx 5 \cdot 2 \cdot 2^{39}$) related-key chosen plaintexts and the memory requirement of this attack is about $5 \cdot 2 \cdot 2^{39} \cdot \frac{256}{8} = 2^{47.3}$ memory bytes.

To compute the time complexity of Step 3 we should estimate the survived fraction of subkey pairs (sk, sk^*) with respect to each loop i. As in the previous differential-nonlinear attack on 32-round SHACAL-2, the number of ciphertext pairs satisfying Eq. (4.5) behaves like a binomial random variable $X \sim Bin(2^{39}, \frac{1}{2})$ for a wrong key. Thus, X is approximated according to the normal distribution $N(\mu, \sigma^2)$ where $\mu = 2^{38}$ and $\sigma^2 = 2^{37}$, equivalently Z $(= \frac{X-\mu}{\sigma}) \sim N(0, 1)$. Since $Pr[X \geq 2^{38} + 2^{21.6}$ or $X \leq 2^{38} - 2^{21.6}] = Pr[Z \geq 8.5742$ or $Z \leq -8.5742] \approx 2^{-53.3}$, the surviving fraction of subkey pairs with respect to each loop i is about $2^{-53.3}$. It follows that after the i^{th} loop the number of surviving subkey pairs is about $(2^{207})^2 \cdot 2^{-53.3 \cdot (i+1)}$. Hence the average time complexity of Step 3 is about $\sum_{i=0}^{4} 2^{39} \cdot 2 \cdot (2^{207})^2 \cdot 2^{-53.3 \cdot i} \cdot \frac{7}{35} \cdot \frac{1}{2} = 2^{450.6}$ 35-round SHACAL-2 encryptions. Since the number of surviving subkey pairs in Step 3 is about $(2^{207})^2 \cdot 2^{-53.3 \cdot 5} \cdot 1/2 = 2^{146.7}$, the average time complexity of Step 4 is about $2^{146.7} \cdot 2^{305} \cdot \frac{7}{35} = 2^{449.2}$ 35-round SHACAL-2 encryptions. Thus, the total average time complexity of this attack is about $2^{450.6} + 2^{449.2} = 2^{451.1}$ 35-round SHACAL-2 encryptions.

In order to compute the success rate of this attack we check the probability that the right subkey pair survives in Step 3. For the right subkey pair Eq. (4.5) holds with a probability of approximately $\frac{1}{2} + 2^{-17}$ or $\frac{1}{2} - 2^{-17}$. If the probability is approximately $\frac{1}{2} + 2^{-17}$, we have $X \sim Bin(2^{39}, \frac{1}{2} + 2^{-17})$ (i.e., $X \sim N(\mu, \sigma^2)$ where $\mu = 2^{38} + 2^{22}$ and $\sigma^2 = \mu \cdot (\frac{1}{2} - 2^{-17})$) where X is the number of ciphertext pairs satisfying Eq. (4.5) for the right subkey pair. Using the above analysis we can check the probability that the right subkey pair survives in each loop of Step 3 is about $1 - 2^{-8.34}$ ($\approx Pr[X \geq 2^{38} + 2^{21.6}] = Pr[Z \geq -2.7395]$). It follows that the probability that the right subkey pair survives in Step 3 is about 0.98 ($\approx (1 - 2^{-8.34})^5$). If Eq. (4.5) holds with a probability of approximately $\frac{1}{2} - 2^{-17}$, we have the same result. Therefore, the success rate of this attack is about 0.98.

4.6 Related-Key Rectangle Attack on 42-Round SHACAL-2

In this section, we use the following two properties in SHACAL-2 along with Properties 1 and 3.

Property 5 *Consider the difference propagation between a pair of data for any four consecutive rounds i to $i + 3$. If the difference $(\Delta A^i, \Delta B^i, \cdots, \Delta H^i)$ just before the i-th round is known, then we have the following properties:*

1. *The differences* ΔB^{i+1}, ΔC^{i+1}, ΔD^{i+1}, ΔF^{i+1}, ΔG^{i+1} *and* ΔH^{i+1} *just before the* $(i+1)$*-th round can be determined; they are equal to* ΔA^i, ΔB^i, ΔC^i, ΔE^i, ΔF^i *and* ΔG^i, *respectively.*

2. *The differences* ΔC^{i+2}, ΔD^{i+2}, ΔG^{i+2} *and* ΔH^{i+2} *just before the* $(i+2)$*-th round can be determined; they are equal to* ΔA^i, ΔB^i, ΔE^i *and* ΔF^i, *respectively.*

3. *The differences* ΔD^{i+3} *and* ΔH^{i+3} *just before the* $(i+3)$*-th round can be determined; they are equal to* ΔA^i *and* ΔE^i, *respectively.*

Property 6 *Let the two related keys* K *and* K^* *have difference* e_{31} *in both the 0-th and 9-th round keys and have all zero difference in the other round keys of the first 16 rounds, then we can conclude by the key schedule that the round keys from 16 to 23 (*i.e.*,* $K^{16}, K^{17}, \cdots, K^{23}$*) have all zero differences, for the following equation holds with probability 1,*

$$\begin{aligned} K^{*16} &= \sigma_1(K^{*14}) + K^{*9} + \sigma_0(K^{*1}) + K^{*0} \\ &= \sigma_1(K^{14}) + (K^9 \oplus e_{31}) + \sigma_0(K^1) + (K^0 \oplus e_{31}) \\ &= \sigma_1(K^{14}) + K^9 + \sigma_0(K^1) + K^0 \\ &= K^{16}. \end{aligned}$$

Based on Properties 1, 3 and 6, we explore a 34-round related-key rectangle distinguisher, which can be directly used to mount a related-key rectangle attack on 38-round SHACAL-2. By Property 5, we can partially determine whether a candidate quartet is a valid one earlier than usual: if not, we can discard it immediately, which results in fewer computations in the remaining steps and may allow us to proceed by guessing one or more round subkeys, depending on how many candidate quartets are remaining. In the case of SHACAL-2, we find that this early abort technique can allow us to break two more rounds, that is, 40-round SHACAL-2 can be broken faster than exhaustive key search. Finally, based on several other delicate observations, we mount a related-key rectangle attack on 42-round SHACAL-2. The details are as follows.

4.6.1 34-Round Related-Key Rectangle Distinguisher of TYPE 1

The key schedule of SHACAL-2 has the property that if the two related keys K and K^* have non-zero difference in only one word, then the 22-*nd* round key is the furthest round key such that all the round keys from rounds 16 to 22 have all zero differences, while if they have non-zero difference in two, three or more words, then the 23-*rd* round key is the furthest round key that has such a property. As

Table 4.9: Related-Key Differential for Rounds 1-24 (E^0) and the Preceding Differential for Round 0 (E^b) of SHACAL-2

r	ΔA^r	ΔB^r	ΔC^r	ΔD^r	ΔE^r	ΔF^r	ΔG^r	ΔH^r	ΔK^r	Prob.
0	0	e_M	e_{31}	·	$e_{9,13,19}$	$e_{18,29}$	e_{31}	·	e_{31}	·
1	0	0	e_M	e_{31}	0	$e_{9,13,19}$	$e_{18,29}$	e_{31}	0	1
2	e_{31}	0	0	e_M	0	0	$e_{9,13,19}$	$e_{18,29}$	0	2^{-12}
3	0	e_{31}	0	0	$e_{6,20,25}$	0	0	$e_{9,13,19}$	0	2^{-7}
4	0	0	e_{31}	0	0	$e_{6,20,25}$	0	0	0	2^{-4}
5	0	0	0	e_{31}	0	0	$e_{6,20,25}$	0	0	2^{-3}
6	0	0	0	0	e_{31}	0	0	$e_{6,20,25}$	0	2^{-4}
7	0	0	0	0	0	e_{31}	0	0	0	2^{-1}
8	0	0	0	0	0	0	e_{31}	0	0	2^{-1}
9	0	0	0	0	0	0	0	e_{31}	e_{31}	1
10	0	0	0	0	0	0	0	0	0	1
⋮				⋮					⋮	⋮
23	0	0	0	0	0	0	0	0	0	1
24	0	0	0	0	0	0	0	0	·	2^{-6}
25	$e_{13,24,28}$	0	0	0	$e_{13,24,28}$	0	0	0	·	·

$M = \{6, 9, 18, 20, 25, 29\}$

stated in Property 6, if the two related keys K and K^* have the difference e_{31} in both the 0-th and 9-th round keys and have all zero difference in the other first 16 round keys, then the round keys from 16 until 23 have all zero differences. Moreover, we observe that these related keys K and K^* produce $K^{24} = L_0 + L_1$ and $K^{*24} = L_0 + (L_1 \oplus e_{13,24,28})$, respectively, where $L_0 = \sigma_1(K^{22}) + K^{17} + K^8$ and $L_1 = \sigma_0(K^9)$. This property is used in building the 24-*th* round of our first related-key differential, which is described in Table 4.9.

To make maximal use of Property 5, we use the first round of the differential (denoted E^b) for key recovery in our following attacks on 40 and 42-round SHACAL-2. As the first part of our related-key rectangle distinguisher we use a 24-round ($1 \sim 24$, denoted E^0) related-key differential $\alpha \rightarrow \beta$ with probability 2^{-38}: $(0, 0, e_{6,9,18,20,25,29}, e_{31}, 0, e_{9,13,19}, e_{18,29}, e_{31}) \rightarrow (e_{13,24,28}, 0, 0, 0, e_{13,24,28}, 0, 0, 0)$. Note that our 24-round related-key differential described in Table 4.9 requires the following 12-bit conditions on the two inputs to round 1, $(A^1, B^1, C^1, D^1, E^1, F^1, G^1, H^1)$ and $(A^{*1}, B^{*1}, C^{*1}, D^{*1}, E^{*1}, F^{*1}, G^{*1}, H^{*1})$ with difference α:

$$\begin{array}{cccc} a^1_6 = b^1_6, & a^1_9 = b^1_9, & a^1_{18} = b^1_{18}, & a^1_{20} = b^1_{20}, \\ a^1_{25} = b^1_{25}, & a^1_{29} = b^1_{29}, & a^1_{31} = b^1_{31}, & e^1_9 = 0, \\ e^1_{13} = 0, & e^1_{18} = 1, & e^1_{19} = 0, & e^1_{29} = 1, \end{array} \tag{4.6}$$

where a^1_i, b^1_i and e^1_i are the i-th bits of A^1, B^1 and E^1, respectively. If the two input values to round 1 meet the α difference and Eq. (4.6), we can remove the

Table 4.10: Differential for Rounds 25-34 of SHACAL-2 (E^1)

r	ΔA^r	ΔB^r	ΔC^r	ΔD^r	ΔE^r	ΔF^r	ΔG^r	ΔH^r	Prob.
25	e_{31}	e_{31}	$e_{M'}$	0	0	$e_{9,13,19}$	$e_{18,29,31}$	0	2^{-15}
26	e_{31}	e_{31}	e_{31}	$e_{M'}$	0	0	$e_{9,13,19}$	$e_{18,29,31}$	2^{-12}
27	0	e_{31}	e_{31}	e_{31}	$e_{6,20,25}$	0	0	$e_{9,13,19}$	2^{-7}
28	0	0	e_{31}	e_{31}	e_{31}	$e_{6,20,25}$	0	0	2^{-8}
29	0	0	0	e_{31}	e_{31}	e_{31}	$e_{6,20,25}$	0	2^{-7}
30	0	0	0	0	e_{31}	e_{31}	e_{31}	$e_{6,20,25}$	2^{-4}
31	0	0	0	0	0	e_{31}	e_{31}	e_{31}	1
32	0	0	0	0	0	0	e_{31}	e_{31}	2^{-1}
33	0	0	0	0	0	0	0	e_{31}	1
34	e_{31}	0	0	0	e_{31}	0	0	0	2^{-11}
35	e_M	e_{31}	0	0	$e_{6,20,25}$	e_{31}	0	0	·

$M = \{6, 9, 18, 20, 25, 29\},\ M' = \{6, 9, 18, 20, 25, 29, 31\}$

differential probabilities incurred by the Ch and Maj functions in rounds 1 and 2 (for round 2, only the condition $a_{31}^1 = b_{31}^1$ is used).

The second part of our related-key rectangle distinguisher is a 10-round differential for rounds 25 to 34: $(e_{31}, e_{31},\ e_{6,9,18,20,25,29,31}, 0, 0, e_{9,13,19}, e_{18,29,31}, 0) \rightarrow (e_{6,9,18,20,25,29}, e_{31}, 0, 0, e_{6,20,25}, e_{31}, 0, 0)$, which holds with probability 2^{-65} (see Table 4.10).

To compute $\widehat{p^*}$ (resp. $\widehat{q}$), we need to sum the square of the probabilities of all the differentials with input difference α through E^0 (resp. all the differentials with output difference δ through E^1), which is computationally infeasible. As an approximative solution, to compute $\widehat{p^*}$ (resp. $\widehat{q}$), we have counted over various similar differentials by changing the last round of the first related-key differential (resp. the first round of the second differential), which results in $\widehat{p^*} = 2^{-37}$ and $\widehat{q} = 2^{-63.4}$. Therefore, we can obtain a lower bound $2^{-456.8}$ $(= (2^{-37} \cdot 2^{-63.4})^2 \cdot 2^{-256})$ for the probability of our 34-round related-key rectangle distinguisher (rounds 1 to 34).

4.6.2 Key Recovery Attack on 40-Round SHACAL-2

We are now ready to explain our related-key rectangle attack on 40-round SHACAL-2. Assume that 40-round SHACAL-2 uses related keys K and K^* with difference $(e_{31}, 0, 0, 0, 0, 0, 0, 0, 0, e_{31}, 0, 0, 0, 0, 0, 0)$. First, we use the 34-round related-key rectangle distinguisher to obtain a small portion of subkey candidates in rounds 0, 35, 36, 37, 38 and 39. Second, we perform an exhaustive search for the obtained subkey candidates and the remaining key bits to recover the 512-bit related keys K and K^*. In order to apply the 34-round distinguisher to this attack, we need to collect enough input pairs to round 1 which meet the

α difference and Eq. (4.6). For this, we use enough pairs of plaintext structures. The details of our attack are as follows:

1. Choose $2^{178.4}$ structures S_i of 2^{64} plaintexts $P_{i,l}$ each, $i = 1, 2, \cdots, 2^{178.4}$, $l = 1, 2, \cdots, 2^{64}$, where in each structure the 192 bits of the words A, B, C, E, F, G are fixed. With a chosen plaintext attack scenario, obtain all the corresponding ciphertexts under the key K, denoted $C_{i,l}$.

2. Compute $2^{178.4}$ structures S_i^* of 2^{64} plaintexts each by XORing the plaintexts in S_i with the 256-bit value $(0, e_{6,9,18,20,25,29}, e_{31}, 0, e_{9,13,19}, e_{18,29}, e_{31}, 0)$. With a chosen plaintext attack scenario, obtain all the corresponding ciphertexts under the key K^*.

3. Guess a 32-bit subkey K^0 in round 0 and compute $K^{*0} = K^0 \oplus e_{31}$. Encrypt each plaintext $P_{i,l}$ through round 0 with K^0 to get its intermediate value just after round 0. We denote the encrypted value by $x_{i,l}$. Check if $x_{i,l}$ meets Eq. (4.6). If it does, compute $x_{i,l}^* = x_{i,l} \oplus \alpha$ and then decrypt $x_{i,l}^*$ through round 0 with K^{*0} to get its plaintext, denoted by $P_{i,l}^*$. Find $P_{i,l}^*$ in S_i^*. We denote by $C_{i,l}^*$ the ciphertext corresponding to $P_{i,l}^*$.

4. Guess a 96-bit subkey pair $((K^{37}, K^{38}, K^{39}), (K^{*37}, K^{*38}, K^{*39}))$ in rounds 37, 38 and 39. For the guessed subkey pair, do the following:

   (a) Decrypt all the ciphertexts $C_{i,l}$ through rounds 37, 38 and 39 with K^{37}, K^{38} and K^{39} to get their intermediate values just before round 37. We denote these values by $C_{i,l}^{37}$. Keep them in a table. Again, decrypt all the ciphertexts $C_{i,l}^*$ through rounds 37, 38 and 39 with K^{*37}, K^{*38} and K^{*39} to get their intermediate values just before round 37. We denote these values by $C_{i,l}^{*37}$. Keep them in another table.

   (b) Check if $C_{i_0,l_0}^{37} \oplus C_{i_1,l_1}^{37}$ and $C_{i_0,l_0}^{*37} \oplus C_{i_1,l_1}^{*37}$ belong to $\delta(2)$, for all $1 \le i_0 < i_1 \le 2^{178.4}$, $1 \le l_0, l_1 \le 2^{64}$ and all $1 \le i_0 = i_1 \le 2^{178.4}$, $1 \le l_0 < l_1 \le 2^{64}$, where $\delta(2)$ is the set of all the possible differences caused by the δ difference after 2 rounds. Record $(K^0, K^{37}, K^{38}, K^{39})$ and all the qualified quartets and then go to Step 5.

5. Guess a 32-bit subkey pair (K^{36}, K^{*36}) in round 36. For the guessed subkey pair, do the following:

   (a) For each remaining quartet $(C_{i_0,l_0}^{37}, C_{i_1,l_1}^{37}, C_{i_0,l_0}^{*37}, C_{i_1,l_1}^{*37})$, decrypt C_{i_0,l_0}^{37} and C_{i_1,l_1}^{37} through round 36 with K^{36} to get their intermediate values just before round 36, and decrypt C_{i_0,l_0}^{*37} and C_{i_1,l_1}^{*37} through round 36 with K^{*36} to get their intermediate values just before round 36. We denote the decrypted quartet by $(C_{i_0,l_0}^{36}, C_{i_1,l_1}^{36}, C_{i_0,l_0}^{*36}, C_{i_1,l_1}^{*36})$.

(b) Check if $C^{36}_{i_0,l_0} \oplus C^{36}_{i_1,l_1}$ and $C^{*36}_{i_0,l_0} \oplus C^{*36}_{i_1,l_1}$ belong to $\delta(1)$, where $\delta(1)$ is the set of all the possible differences caused by the δ difference after 1 round. Record $(K^0, K^{36}, K^{37}, K^{38}, K^{39})$ and all the qualified quartets and then go to Step 6.

6. Guess a 32-bit subkey pair (K^{35}, K^{*35}) in round 35. For the guessed subkey pair, do the following:

   (a) For each remaining quartet $(C^{36}_{i_0,l_0}, C^{36}_{i_1,l_1}, C^{*36}_{i_0,l_0}, C^{*36}_{i_1,l_1})$, decrypt $C^{36}_{i_0,l_0}$ and $C^{36}_{i_1,l_1}$ through round 35 with K^{35} to get their intermediate values just before round 35, and decrypt $C^{*36}_{i_0,l_0}$ and $C^{*36}_{i_1,l_1}$ through round 35 with K^{*35} to get their intermediate values just before round 35. We denote the decrypted quartet by $(C^{35}_{i_0,l_0}, C^{35}_{i_1,l_1}, C^{*35}_{i_0,l_0}, C^{*35}_{i_1,l_1})$.

   (b) Check if $C^{35}_{i_0,l_0} \oplus C^{35}_{i_1,l_1} = C^{*35}_{i_0,l_0} \oplus C^{*35}_{i_1,l_1} = \delta$. If there exist more than 5 quartets passing this δ test, record $(K^0, K^{35}, K^{36}, K^{37}, K^{38}, K^{39})$ and go to Step 7. Otherwise, repeat Step 6 with another guessed key pair (if all the possible key pairs for round 35 are tested, then repeat Step 5 with another guessed key pair for round 36; if all the possible key pairs for round 36 are tested, then repeat Step 4 with another guessed key pair for rounds 37, 38 and 39; if all the possible key pairs for rounds 37, 38 and 39 are tested, then repeat Step 3 with another guessed key pair for round 0).

7. For a suggested $(K^0, K^{35}, K^{36}, K^{37}, K^{38}, K^{39})$, perform an exhaustive search for the remaining 320 key bits using trial encryption. If a 512-bit key is suggested, output it as the master key of the 40-round SHACAL-2. Otherwise, run the above steps with another guess of subkey pair.

This attack requires $2^{243.4}$ related-key chosen plaintexts. The required memory for this attack is dominated by Step 4, which is approximately $2^{243.4} \cdot 32 = 2^{247.4}$ memory bytes.

The time complexities of Steps 1 and 2 are $2^{243.4}$ 40-round SHACAL-2 encryptions each. The time complexity of Step 3 is about $(2^{242.4} + 2^{230.4}) \cdot 2^{32} \cdot \frac{1}{40} \cdot \frac{1}{2} = 2^{268.1}$ 40-round SHACAL-2 encryptions on average, for Eq. (4.6) has a 12-bit filtering. Moreover, for each guessed subkey pair, we have about $2^{230.4 \times 2}/2 = 2^{459.8}$ quartets tested in Step 4. Since the decryptions in Step 4 can be done independent of Step 3, Step 4 requires on average about $2^{231.4} \cdot 2^{192} \cdot \frac{3}{40} \cdot \frac{1}{2} = 2^{418.6}$ 40-round SHACAL-2 encryptions and about $2^{231.4} \cdot 2^{192} \cdot 2^{32} \cdot \frac{1}{2} = 2^{454.4}$ memory accesses.

From the difference δ, we can determine the differences in the words C, D, G, and H of every possible difference in the set $\delta(2)$. Moreover, we observe that there

are about 2^{28} possible differences in B and 2^{17} possible differences in F of $\delta(2)$. Hence, there are about $2^{64+28+17} = 2^{109}$ possible differences in $\delta(2)$. It follows that about $2^{459.8} \cdot 2^{(-256+109)\cdot 2} = 2^{165.8}$ quartets are suggested in Step 4. Since Step 5-(a) runs about 2^{287} times on average (equivalent to the number of guessed subkey pairs), it requires about $2^{165.8} \cdot 4 \cdot 2^{287} \cdot \frac{1}{40} = 2^{449.4}$ 40-round SHACAL-2 encryptions. Similarly, $\delta(1)$ and δ additionally have a 64-bit and a 45-bit filtering, so about $2^{165.8} \cdot 2^{-64\cdot 2} = 2^{37.8}$ and $2^{37.8} \cdot 2^{-45\cdot 2} = 2^{-52.2}$ quartets (for each wrong guess of subkey pairs) are expected to be suggested in Steps 5 and 6, respectively, and thus Step 6 requires on average about $2^{37.8} \cdot 4 \cdot 2^{352} \cdot \frac{1}{40} \cdot \frac{1}{2} = 2^{385.4}$ 40-round SHACAL-2 encryptions. By the Poisson distribution $X \sim \mathrm{Poi}(\lambda = 2^{-52.2})$, $Pr_X[X > 5] \approx 2^{-323}$, the expected number of wrong subkey pairs suggested in Step 6 is about $2^{-323} \cdot 2^{352} = 2^{29}$. It follows that the average time complexity of Step 7 is about 2^{348} $(= 2^{29} \cdot 2^{320} \cdot \frac{1}{2})$ 40-round SHACAL-2 encryptions. Therefore, the total average time complexity of this attack is about $2^{449.4}$ 40-round SHACAL-2 encryptions.

If the guessed subkey pair is right, then the expected number of the quartets suggested in Step 6 is about $2^{459.8} \cdot 2^{-456.8} = 2^3$, for about $2^{459.8}$ quartets are tested in this attack and the 34-round related-key rectangle distinguisher holds with probability $2^{-456.8}$. Thus, the probability that the number of remaining quartets for the right subkey pair is larger than 5 is 0.8 by the Poisson distribution, $X \sim \mathrm{Poi}(\lambda = 2^3)$, $Pr_X[X > 5] \approx 0.8$. Hence, this attack works with a success probability of 0.8.

Note: We can reduce the time complexity of our attack on 40-round SHACAL-2 down to $2^{447.4}$ 40-round SHACAL-2 encryptions by adopting the following two delicate improvements. First, we only guess the least significant 31 bits of the subkey K^0 in Step 3, due to the fact that the most significant bit in the key difference is fixed. Second, we guess the least significant 31 bits of the subkey pairs (K^{36}, K^{*36}) and the difference between their most significant bits to check the $\delta(1)$ test in Step 5, instead of guessing all the 32-bit values of the subkey pairs. In Step 6, we guess the least significant 31 bits of the subkey pairs (K^{35}, K^{*35}) and the difference between their most significant bits to check the δ test. Since the total time complexity of this attack is dominated by Step 5-(a), it is reduced by a factor of 4.

4.6.3 Key Recovery Attack on 42-Round SHACAL-2

We now improve the above attack to break 42-round SHACAL-2 by guessing additive differences between related subkey pairs, instead of guessing actual values of them. Our improved attack is based on the following observations.

Observation 1. If we know the actual values of $(A^i, B^i, \cdots, H^i)$ and $(A^{*i}, B^{*i}, \cdots, H^{*i})$, and the additive difference between K^{i-1} and K^{*i-1}, then we know the actual values of $(A^{i-1}, B^{i-1}, \cdots, G^{i-1})$ and $(A^{*i-1}, B^{*i-1}, \cdots, G^{*i-1})$, and the additive difference between H^{i-1} and H^{*i-1}.

Observation 2. If we know the actual values of $(A^{i-1}, B^{i-1}, \cdots, G^{i-1})$ and $(A^{*i-1}, B^{*i-1}, \cdots, G^{*i-1})$, and the additive difference between H^{i-1} and H^{*i-1}, then we know the actual values of $(A^{i-5}, B^{i-5}, C^{i-5})$ and $(A^{*i-5}, B^{*i-5}, C^{*i-5})$, and the additive difference between D^{i-5} and D^{*i-5}.

Observation 3. The additive difference between 32-bit words X and Y is the same as their XOR difference if $X \oplus Y = 0$ or $X \oplus Y = e_{31}$.

Based on these observations, the above attack algorithm can be improved to an attack on 42-round SHACAL-2, which uses the early abort technique one step earlier. The improved attack procedure goes as follows:

- We run the above Steps 1, 2 and 3.
- In Step 4, we guess a 64-bit subkey pair $((K^{40}, K^{41}), (K^{*40}, K^{*41}))$ and an additive difference between K^{39} and K^{*39}, and then decrypt all the ciphertexts to obtain the actual values of $(A^{39}, B^{39}, \cdots, G^{39})$ and $(A^{*39}, B^{*39}, \cdots, G^{*39})$, and the additive difference between H^{39} and H^{*39} (by Observation 1). It allows to deduce (A^{35}, B^{35}, C^{35}) and $(A^{*35}, B^{*35}, C^{*35})$, and the additive difference between D^{35} and D^{*35} (by Observation 2), so we can discard some wrong quartets by checking if the decrypted quartets satisfy the first half of the δ difference. Since it has a 256-bit filtering for each decrypted quartet, about $2^{459.8} \cdot 2^{-256} = 2^{203.8}$ quartets are suggested. This step requires about $2^{64 \cdot 2+32} \cdot 2^{231.4} \cdot \frac{7}{42} \cdot \frac{1}{2} = 2^{387.8}$ 42-round SHACAL-2 encryptions and $2^{64 \cdot 2+64} \cdot 2^{231.4} \cdot \frac{1}{2} = 2^{422.4}$ memory accesses on average.
- In Step 5, we guess a 64-bit subkey pair of (K^{38}, K^{39}) and (K^{*38}, K^{*39}) (note the additive difference between K^{39} and K^{*39} is fixed in the previous step), and then decrypt all the remaining quartets to obtain their input values of round 38. Since H^{38} is the same as E^{35}, we can discard all the quartets which do not satisfy the $e_{6,20,25}$ XOR difference in H^{38}. It has a 64-bit filtering for each decrypted quartet, so about $2^{203.8} \cdot 2^{-64} = 2^{139.8}$ quartets are suggested. This step requires on average about $2^{64 \cdot 4+32} \cdot 2^{203.8+2} \cdot \frac{1}{42} \cdot \frac{1}{2} = 2^{487.4}$ 42-round SHACAL-2 encryptions.
- In Step 6, we guess an additive difference between K^{37} and K^{*37} to check if the remaining quartets satisfy the e_{31} difference in H^{37}, which is the same as F^{35}. In Step 7, we guess a 64-bit subkey pair of (K^{36}, K^{37}) and

(K^{*36}, K^{*37}) (note the additive difference between K^{37} and K^{*37} is fixed in the previous step) to check if the remaining quartets have a zero difference in H^{36}, which is the same as G^{35}. In Step 8, we guess a 64-bit subkey pair of (K^{35}, K^{36}) and (K^{*35}, K^{*36}) (note the additive difference between K^{36} and K^{*36} is fixed in the previous step) to check if the remaining quartets have a zero difference in H^{35}. We go to the final step with the guessed subkey pair which has more than 5 remaining quartets. Finally, in Step 9, we perform an exhaustive search to find the 512-bit master keys. The time complexity of Steps 6, 7, 8 and 9 is substantially smaller than that of Step 5.

Therefore, the time complexity of the attack is dominated by Step 5, which is about $2^{487.4}$ 42-round SHACAL-2 encryptions. Obviously, the attack is faster than an exhaustive key search.

4.7 Related-Key Rectangle Attack on 10-Round AES-192

This section shows how to exploit the related-key rectangle attack to devise a key recovery attack on 10-round AES-192 with 256 related keys. At the end of this section, we also present related-key rectangle attacks on 8-round AES-192 with 2 or 4 related keys and on 9-round AES-256 with 4 related keys.

Denote the 10 rounds of AES-192 by $E = E^f \circ E^1 \circ E^0 \circ E^b$, where E^b is round 0 including the whitening key addition step and excluding the key addition step of round 0, E^0 is rounds 1-4 including the key addition step of round 0, E^1 is rounds 5-8 and E^f is round 9. In our 10-round AES-192 attack, we use a related-key truncated differential for E^0 depicted in Fig. 4.1 and another related-key truncated differential for E^1 depicted in Fig. 4.2. These related-key truncated differentials are built based on a slow difference propagation of the key schedule of AES-192. After we convert these related-key truncated differentials for E^0 and E^1 into a related-key rectangle distinguisher for $E^1 \circ E^0$, we apply it to recover some portions of the keys in E^b and E^f. Before describing our attack, we define some notation which is used in our attacks on AES.

- $K_w, K_w^*, K_w', K_w'^*$: whitening keys generated from master keys K, K^*, K', K'^*, respectively.
- $K_i, K_i^*, K_i', K_i'^*$: subkeys of round i generated from K, K^*, K', K'^*, respectively.
- P, P^*, P', P'^*: plaintexts encrypted under K, K^*, K', K'^*, respectively.
- $I_i, I_i^*, I_i', I_i'^*$: input values to round i caused by plaintexts P, P^*, P', P'^* under K, K^*, K', K'^*, respectively.

- a: a fixed nonzero byte value.
- b, c: output differences of S-box for the fixed nonzero input difference a.
- $*$: a variable and unknown byte.

4.7.1 8-Round Related-Key Rectangle Distinguisher of TYPE 3

In order to convert the two 4-round related-key truncated differentials depicted in Figs. 4.1 and 4.2 into an 8-round related-key rectangle distinguisher, we first make the following Assumptions 7, 8 and 9 (note that Assumptions 1-6 listed in Sect. 1.6.2).

Assumption 7. The key quartet (K, K^*, K', K'^*) is related as follows;

$$K \oplus K^* = K' \oplus K'^* = \Delta K,\ K \oplus K' = K^* \oplus K'^* = \Delta K' .$$

Assumption 8. A plaintext quartet (P, P^*, P', P'^*) is related as follows;

$$P \oplus P^*,\ P' \oplus P'^* \in \Delta P .$$

Assumption 9. $E^b_K(P) \oplus E^b_{K^*}(P^*) = E^b_{K'}(P') \oplus E^b_{K'^*}(P'^*) = \Delta K_0$.

As stated in our notation, $I_5 = E^0_K(E^b_K(P))$, $I^*_5 = E^0_{K^*}(E^b_{K^*}(P^*))$, $I'_5 = E^0_{K'}(E^b_{K'}(P'))$ and $I'^*_5 = E^0_{K'^*}(E^b_{K'^*}(P'^*))$. By the related-key truncated differential for E^0, $I_5 \oplus I^*_5$ is equal to $I'_5 \oplus I'^*_5$ with a probability of about $(2^{-32} \cdot 2^{-7})^2 \cdot (2^7 - 2) \cdot 2^{32} + (2^{-32} \cdot 2^{-6})^2 \cdot 2^{32} \approx 2^{-39}$. It follows from counting over all the differentials that can be generated by the active S-box with input difference a and the other four active S-boxes in round 4. Since ShiftRows and MixColumns are linear layers, they can be ignored in round 4 when computing the probability (see Fig. 4.1). Moreover, the probability that $I_5 \oplus I'_5,\ I^*_5 \oplus I'^*_5 \in \Delta I'_5$ is about 2^{-64} under the condition $I_5 \oplus I^*_5 = I'_5 \oplus I'^*_5$ (see Fig. 4.2 for $\Delta I'_5$). Hence the probability that $I_5 \oplus I'_5,\ I^*_5 \oplus I'^*_5 \in \Delta I'_5$ is about $2^{-39} \cdot 2^{-64} = 2^{-103}$. Since $e_K(I_5) \oplus e_{K'}(I'_5) = 0$ with probability 2^{-64} and $e_{K^*}(I^*_5) \oplus e_{K'^*}(I'^*_5) = 0$ with a probability of about 2^{-64} under the condition $I_5 \oplus I'_5,\ I^*_5 \oplus I'^*_5 \in \Delta I'_5$, where e is the encryption for round 5,

$$E^1_K(I_5) \oplus E^1_{K'}(I'_5),\ E^1_{K^*}(I^*_5) \oplus E^1_{K'^*}(I'^*_5) \in \Delta I'_9$$

with a probability of 2^{-231} (see Fig. 4.2 for $\Delta I'_9$). However, the same statement can be applied to a random cipher with probability $(2^{-128} \cdot (2^7 - 1))^2 \approx 2^{-242}$, since the number of elements in $\Delta I'_9$ is $2^7 - 1$. The first column of $\Delta I'_9$ is

$$\mathcal{B} = \{\mathrm{MC}(y, 0, 0, 0) \mid y = \mathrm{BS}(x) \oplus \mathrm{BS}(x \oplus a),\ x = 0, 1, 2, \cdots, 255\} . \qquad (4.7)$$

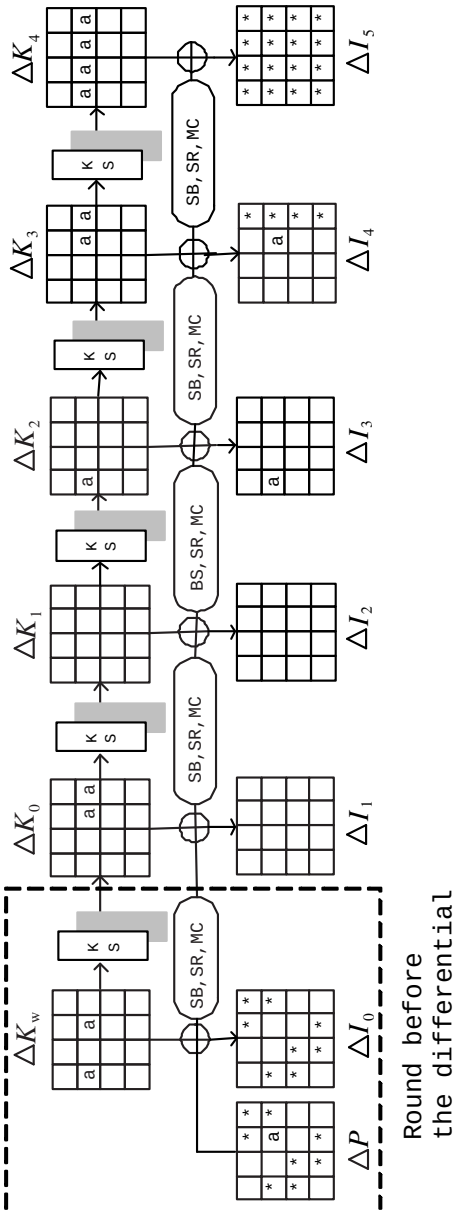


Figure 4.1: Related-Key Truncated Differential for Rounds 1-4 of AES-192

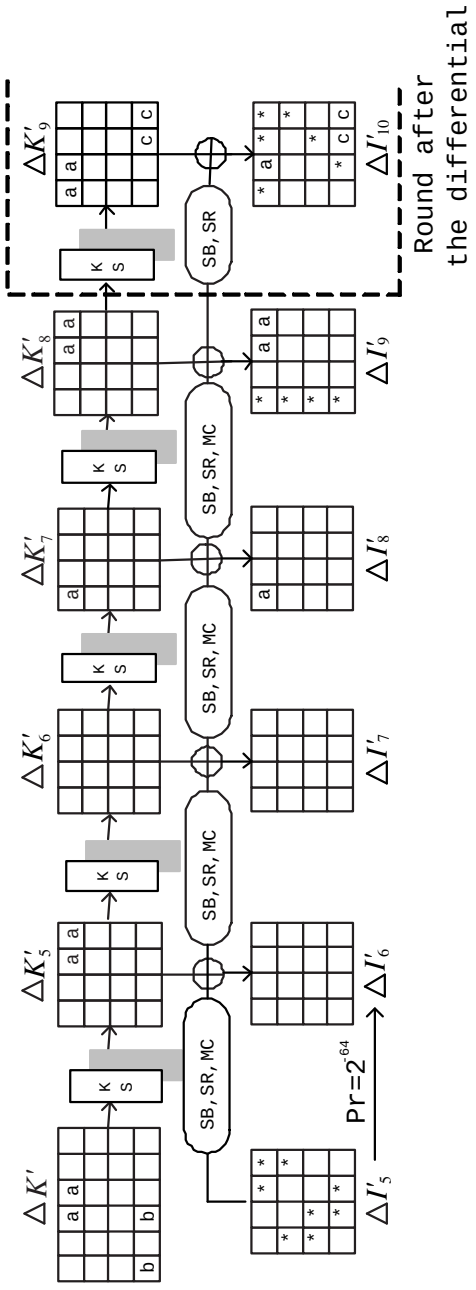


Figure 4.2: Related-Key Truncated Differential for Rounds 5-8 of AES-192

4.7.2 Key Recovery Attack on 10-Round AES-192

In order to produce the round-key differences depicted in Fig. 4.2, the 8-bit difference a should satisfy the 8-bit difference b after S-box during the key generation for the third column of $\Delta K'_3$. Given the 8-bit difference a there are 127 possible candidates for the b difference, hence the attack starts by gathering all possible key quartets $(K, K^*, \widetilde{K}', \widetilde{K}'^*)$ of which one satisfies the desired key condition. Note that the keys $K^* = K \oplus \Delta K$, $\widetilde{K}' = K \oplus \Delta\widetilde{K}'$ and $\widetilde{K}'^* = K \oplus \Delta K \oplus \Delta\widetilde{K}'$ where ΔK is fixed as ΔK_w and the first two columns of ΔK_0 in Fig. 4.1 and $\Delta\widetilde{K}'$ is one of the 127 possible differences; bytes 8 and 12 are both a, bytes 3 and 11 are both b' and other bytes are all zeros, where b' is one of the 127 possible candidates for the b difference. So the total number of required related keys is 256. We apply the rectangle attack to 10-round AES-192 for each key quartet. During this procedure, we stop our attack when we have found a key quartet $(K, K^*, \widetilde{K}', \widetilde{K}'^*)$ that satisfies the desired key condition $b' = b$, i.e., $\Delta\widetilde{K}' = \Delta K'$, $(K, K^*, \widetilde{K}', \widetilde{K}'^*) = (K, K^*, K', K'^*)$.

The aim of our attack is to recover bytes 1, 2, 6, 7, 8, 11, 12, 13 of the whitening key quartet $(K_w, K_w^*, K'_w, K'^*_w)$ and bytes 0, 7, 8, 10, 12, 13 of the subkey quartet $(K_9, K_9^*, K'_9, K'^*_9)$, for which the byte positions are marked as $*$ on ΔP and $\Delta I'_{10}$ depicted in Fig. 4.1 and Fig. 4.2. This attack distinguishes a right key quartet from wrong ones by analyzing enough plaintext quartets with each guessed key quartet. In this attack, we need 2^{64} guesses for the whitening key quartet and 2^{72} guesses for the subkey quartet in round 9, since bytes 0, 7, 8, 10, 12, 13 of ΔK_9 are d, 0, d, e, 0, f, respectively, where d, e and f are unknown 8-bit values (note that bytes 0, 7, 8, 10, 12, 13 of $\Delta K'_9$ are fixed by a, 0, 0, 0, 0, 0, respectively). Thus, taking into account the guessing of candidates for the difference b, we need about 2^{143} key guesses in total (in our attack it can be reduced by a factor of two on average).

The attack algorithm goes as follows:

1. Choose 2^{54} structures $S_1, S_2, \cdots, S_{2^{54}}$ of 2^{64} plaintexts each, where in each structure the 64 bits of bytes 0, 3, 4, 5, 9, 10, 14, 15 are fixed. With a chosen plaintext attack scenario, obtain all their corresponding ciphertexts under the key K. (Step 1 takes 2^{118} chosen plaintexts and about 2^{118} encryptions. Note that n encrpytions mean n 10-round AES-192 encryptions.)

2. Compute 2^{54} structures $S_1^*, S_2^*, \cdots, S_{2^{54}}^*$ of 2^{64} plaintexts each by XORing the plaintexts in $S_1, S_2, \cdots, S_{2^{54}}$ with a 128-bit value M of which byte 9 is a and all the other bytes are 0. With a chosen plaintext attack scenario, obtain all their corresponding ciphertexts under the key K^*, where $K^* = K \oplus \Delta K$. (Similarly, Step 2 takes 2^{118} chosen plaintexts and about 2^{118}

encryptions.)

3. Guess a candidate for the difference b and compute $\Delta\widetilde{K}'$. For the key difference $\Delta\widetilde{K}'$, do the following:

3.1 Choose 2^{54} structures $S'_1,\ S'_2, \cdots,\ S'_{2^{54}}$ of 2^{64} plaintexts each, where in each structure the 64 bits of bytes 0, 3, 4, 5, 9, 10, 14, 15 are fixed. With a chosen plaintext attack scenario, obtain all their corresponding ciphertexts under the key $\widetilde{K}'$, where $\widetilde{K}' = K \oplus \Delta\widetilde{K}'$. (For each guess of $\Delta\widetilde{K}'$, Step 3.1 takes 2^{118} chosen plaintexts and about 2^{118} encryptions.)

3.2 Compute 2^{54} structures $S'^*_1,\ S'^*_2, \cdots,\ S'^*_{2^{54}}$ of 2^{64} plaintexts each by XORing the plaintexts in $S'_1,\ S'_2, \cdots,\ S'_{2^{54}}$ with M. With a chosen plaintext attack scenario, obtain all their corresponding ciphertexts under the key $\widetilde{K}'^*$, where $\widetilde{K}'^* = K \oplus \Delta K \oplus \Delta\widetilde{K}'$. Go to Step 4. (For each guess of $\Delta\widetilde{K}'$, this step also takes 2^{118} chosen plaintexts and about 2^{118} encryptions.)

4. Guess a 64-bit subkey k_w in the position of bytes 1, 2, 6, 7, 8, 11, 12, 13 of the whitening key and compute $k^*_w = k_w \oplus \Delta k_w$, $k'_w = k_w \oplus \Delta\widetilde{k}'_w$, $k'^*_w = k_w \oplus \Delta k_w \oplus \Delta\widetilde{k}'_w$, where Δk_w and $\Delta\widetilde{k}'_w$ are the fixed 64-bit key differences in the position of bytes 1, 2, 6, 7, 8, 11, 12, 13 of ΔK_w (depicted in Fig. 4.1) and $\Delta\widetilde{K}'_w$, respectively. For the subkey quartet $(k_w, k^*_w, k'_w, k'^*_w)$, do the following:

4.1 Partially encrypt each plaintext P_{i,l_0} in S_i through E^b under k_w, $i = 1, 2, \cdots, 2^{54},\ l_0 = 1, 2, \cdots, 2^{64}$. We denote the partially encrypted value by x_{i,l_0}. Partially decrypt each $x_{i,l_0} \oplus \Delta K_{0,R}$ through E^b under k^*_w, and find the corresponding plaintext in S^*_i, denoted P^*_{i,l_0}. We denote the corresponding ciphertexts of P_{i,l_0} and P^*_{i,l_0} by C_{i,l_0} and C^*_{i,l_0}, respectively. (For each guess of k_w, Step 4.1 takes about $2^{64+1} \cdot (8/16) \cdot (1/10) = 2^{60.7}$ encryptions. Note that this step is independent of $\Delta\widetilde{K}'$ in Step 3, so there is no need to run this step for every iteration of Step 3.)

4.2 Partially encrypt each plaintext P'_{j,l_1} in S'_j through E^b under k'_w, $j = 1, 2, \cdots, 2^{54},\ l_1 = 1, 2, \cdots, 2^{64}$. We denote the partially encrypted value by x'_{j,l_1}. Partially decrypt each $x'_{j,l_1} \oplus \Delta K_{0,R}$ through E^b under k'^*_w, and find the corresponding plaintext in S'^*_j, denoted P'^*_{j,l_1}. We denote the corresponding ciphertexts of P'_{j,l_1} and P'^*_{j,l_1} by C'_{j,l_1} and C'^*_{j,l_1}, respectively. (For each guess of $(k_w, \Delta\widetilde{K}')$, Step 4.2 takes about $2^{64+1} \cdot (8/16) \cdot (1/10) = 2^{60.7}$ encryptions.)

4.3 Check that $C_{i,l_0} \oplus C'_{j,l_1} \in \Delta I'_{10}$ for all i, j, l_0 and l_1, where $I'_{10} = \{((*,0,0,0),(a,0,0,*),(b_1,0,*,b_2),(b_3,*,0,b_2))\}$, $*$ is any 8-bit value, and b_i is one of the output differences caused by the input difference a to the S-box (see Fig. 4.2). For each index quartet (i,j,l_0,l_1) satisfying the test, again check that $C^*_{i,l_0} \oplus C'^*_{j,l_1} \in \Delta I'_{10}$ (note that in the latter test bytes 11 and 15 of $C^*_{i,l_0} \oplus C'^*_{j,l_1}$ should be the same as those of $C_{i,l_0} \oplus C'_{j,l_1}$). Keep in a table all the ciphertext quartets $(C_{i,l_0}, C'_{j,l_1}, C^*_{i,l_0}, C'^*_{j,l_1})$ passing the both tests and go to Step 5 with this table. Since $\Delta I'_{10}$ in the former test has 2^{53} out of 2^{128} values and $\Delta I'_{10}$ in the latter test has 2^{46} out of 2^{128} values, the expected number of quartets kept in the table is about $2^{(54+64)\cdot 2} \cdot 2^{-128+53} \cdot 2^{-128+46} = 2^{79}$. (Since the tests can be performed efficiently with a hash table, Step 4.3 takes a relatively small time complexity.)

5. Guess an 8-bit subkey $k_{9,v}$ in the position of byte 12 in round 9 and set $k^*_{9,v} = k'_{9,v} = k'^*_{9,v} = k_{9,v}$. For the 8-bit subkey quartet $(k_{9,v}, k^*_{9,v}, k'_{9,v}, k'^*_{9,v})$, do the following:

5.1 For all the remaining ciphertext quartets $(C_{i,l_0}, C'_{j,l_1}, C^*_{i,l_0}, C'^*_{j,l_1})$, partially decrypt C_{i,l_0} and C'_{j,l_1} under $k_{9,v}$ and $k'_{9,v}$ through E^f, respectively. If the partially decrypted pairs do not have the difference a, then discard the corresponding ciphertext quartets. Since it has approximately a 7-bit filtering, the number of remaining quartets after this step is about 2^{72}. (The partial decryptions can be done after the remaining ciphertext quartets have been sorted by byte 12 of (C_{i,l_0}, C'_{j,l_1}) or this step can use a pre-computed table, so Step 5.1 takes a relatively small time complexity.)

5.2 For all the remaining ciphertext quartets $(C_{i,l_0}, C'_{j,l_1}, C^*_{i,l_0}, C'^*_{j,l_1})$, partially decrypt C^*_{i,l_0} and C'^*_{j,l_1} under $k^*_{9,v}$ and $k'^*_{9,v}$ through E^f, respectively. If the partially decrypted pairs do not have the difference a, discard the corresponding ciphertext quartets and then go to Step 6. It also imposes approximately a 7-bit filtering, hence the number of remaining quartets after this step is about 2^{65}. (Similarly, Step 5.2 can be performed efficiently.)

6. Guess an 8-bit subkey $k_{9,w}$ in the position of byte 8 in round 9 and set $k'_{9,w} = k_{9,w}$. For the 8-bit subkey pair ($k_{9,w}$,$k'_{9,w}$), do the following:

6.1 For all the remaining ciphertext quartets $(C_{i,l_0}, C'_{j,l_1}, C^*_{i,l_0}, C'^*_{j,l_1})$, partially decrypt C_{i,l_0} and C'_{j,l_1} under $k_{9,w}$ and $k'_{9,w}$ through E^f, respectively. If the partially decrypted pairs do not have the difference a,

then discard the corresponding ciphertext quartets. Since this imposes approximately a 7-bit filtering, the number of remaining quartets after this step is about 2^{58}.

6.2 Guess an 8-bit value d to form an 8-bit subkey pair ($k^*_{9,w} = k_{9,w} \oplus d$, $k'^*_{9,w} = k_{9,w} \oplus d$) in the position of byte 8 in round 9. For the 8-bit subkey pair ($k^*_{9,w}$,$k'^*_{9,w}$), do the following:

6.2.1 For all the remaining ciphertext quartets $(C_{i,l_0}, C'_{j,l_1}, C^*_{i,l_0}, C'^*_{j,l_1})$, partially decrypt C^*_{i,l_0} and C'^*_{j,l_1} under $k^*_{9,w}$ and $k'^*_{9,w}$ through E^f, respectively. If the partially decrypted pairs do not have the difference a, discard the corresponding ciphertext quartets and then go to Step 7. It also induces approximately a 7-bit filtering, hence the number of remaining quartets after this step is about 2^{51}. (Similarly, Step 6 can be performed efficiently.)

7. Guess a 32-bit subkey $k_{9,y}$ in the position of bytes 0, 7, 10, 13 in round 9 and compute $k'_{9,y} = k_{9,y} \oplus (a, 0, 0, 0)$. For the 32-bit subkey pair $(k_{9,y}, k'_{9,y})$, do the following:

7.1 For all the remaining ciphertext quartets $(C_{i,l_0}, C'_{j,l_1}, C^*_{i,l_0}, C'^*_{j,l_1})$, partially decrypt C_{i,l_0} and C'_{j,l_1} under $k_{9,y}$ and $k'_{9,y}$ through E^f, respectively. If the differences of the partially decrypted pairs are not in $\mathcal{B}$ (see Eq. (4.7)), then discard the corresponding ciphertext quartets. Since $\mathcal{B}$ has $2^7 - 1$ out of 2^{32} values, the remaining quartets after this step is about 2^{26}. (For each guess of $(k_{9,y}, d, k_{9,w}, k_{9,v}, k_w, \Delta\widetilde{K}')$, Step 7.1 takes $2^{51+1} \cdot (4/16) \cdot (1/10) = 2^{46.7}$ encryptions.)

7.2 Guess two 8-bit values e, f to form a 32-bit subkey pair ($k^*_{9,y} = k_{9,y} \oplus (d, 0, e, f)$, $k'^*_{9,y} = k_{9,y} \oplus (d \oplus a, 0, e, f)$) in the position of bytes 0, 7, 10, 13 in round 9. For the 32-bit subkey pair ($k^*_{9,y}$,$k'^*_{9,y}$), do the following:

7.2.1 For all the remaining ciphertext quartets $(C_{i,l_0}, C'_{j,l_1}, C^*_{i,l_0}, C'^*_{j,l_1})$, partially decrypt C^*_{i,l_0} and C'^*_{j,l_1} under $k^*_{9,y}$ and $k'^*_{9,y}$ through E^f, respectively. If the differences of the partially decrypted pairs are not in $\mathcal{B}$, discard the corresponding ciphertext quartets and then go to Step 8. This also induces approximately about a 25-bit filtering, hence the number of remaining quartets after this step is about 2 for each wrong key guess. (For each guess of $(e, f, k_{9,y}, d, k_{9,w}, k_{9,v}, k_w, \Delta\widetilde{K}')$, this step takes $2^{26+1} \cdot (4/16) \cdot (1/10) = 2^{21.7}$ encryptions.)

8. For the remaining ciphertext quartets $(C_{i,l_0}, C'_{j,l_1}, C^*_{i,l_0}, C'^*_{j,l_1})$, classify the quartets according to the differences of C_{i,l_0} and C'_{j,l_1} by byte 11. Discard

all the ciphertext quartets except for the group with the largest number of quartets and then go to Step 9. Since this results in approximately a 7-bit filtering for each pair of quartets, the remaining quartets after this step is expected to be about 2^{-6} for each wrong key guess. (It takes a relatively small time complexity.)

9. If there are more than 16 quartets in the table, then output the guessed subkey quartet as the right one. Otherwise, run the above steps with another guess for the subkey quartet, i.e., $(e, f, k_{9,y}, d, k_{9,w}, k_{9,v}, k_w, \Delta\widetilde{K}')$.

About 2^{125} chosen plaintexts in Steps 1, 2 and 3 are encrypted on average, hence the data complexity of this attack is about 2^{125} related-key chosen plaintexts and the time complexity of Steps 1, 2 and 3 is about 2^{125} encryptions. Step 4 runs about 2^{70} times, so the time complexity of Step 4 is about $2^{60.7+70} = 2^{130.7}$ encryptions (it can be improved by a factor of about 2^4 by using a pre-computed table[2]). As stated above, Steps 5, 6 and 8 take relatively small time complexities compared to other steps.

The time complexity for Step 7 depends on how many times this step runs, which can be measured by the number of guessed subkeys (including d, e and f). Since Steps 7.1 and 7.2 run in this attack 2^{126} and 2^{142} times on average, these steps take $2^{172.7}$ and $2^{163.7}$ encryptions, respectively. However, the time complexities of these steps can be improved by using a divide and conquer technique. In Step 7.1, two of the four bytes of the remaining ciphertext quartets are first decrypted (these partial decryptions can be performed after the remaining ciphertext quartets are sorted by these two bytes) and discard the ciphertext quartets of which the decrypted two bytes do not have a difference in $\mathcal{B}$ with respect to the two-byte position, and then do this test with other two bytes of the remaining ciphertext quartets byte by byte. With this divide and conquer technique, we can also run Step 7.2. This method allows Steps 7.1 and 7.2 to decrease their time complexities down to about $2^{135.7}$ and $2^{146.7}$ encryptions, respectively.

We can calculate the success rate of the attack by using the Poisson distribution as in our 8-round AES-192 attack. Since the expected number of remaining quartets for each wrong subkey quartet is 2^{-6}, the probability that the number of remaining quartets for each wrong subkey quartet is larger than 16 is 2^{-150} by the Poisson distribution, $X \sim \mathrm{Poi}(\lambda = 2^{-6})$, $Pr_X[X > 16] \approx 2^{-150}$. It follows that the probability that the attack outputs a wrong subkey quartet is quite low, since the total number of guessed wrong subkey quartets is about 2^{142}. On the other

[2]Before running this attack, we can pre-compute a table which keeps 2^{64} input pairs (I_0, I_0^*) to round 0, where $I_0^* = \mathrm{BS}^{-1}(\mathrm{SR}^{-1}(\mathrm{MC}^{-1}(\mathrm{MC}(\mathrm{SR}(\mathrm{BS}(I_0))) \oplus \Delta K_{0,R})))$. If Step 4.1 has access to this table for each guessed subkey (k_w, k_w^*), it can find plaintext pairs $(P_{i,l}, P_{i,l}^*)$ by XORing (k_w, k_w^*) with (I_0, I_0^*).

hand, the expected number of remaining quartets for the right subkey quartet is about $2^5 = 2^{236} \cdot 2^{-231}$ due to our 8-round related-key rectangle distinguisher. Thus, the probability that the number of remaining quartets for the right key quartet is larger than 16 is 0.99 by the Poisson distribution, $Y \sim \mathrm{Poi}(\lambda = 2^5)$, $Pr_Y[Y > 16] \approx 0.99$.

Therefore, this attack works with a data complexity of about 2^{125} related-key chosen plaintexts and with a time complexity of about $2^{146.7}$ encryptions and with a success rate of 0.99.

4.7.3 Related-Key Rectangle Attacks on 8-Round AES-192 and 9-Round AES-256

Similarly, we can construct related-key rectangle attacks on 8-round AES-192 with two and four related keys (TYPES 1 and 3) and on 9-round AES-256 with four related keys (TYPE 3). See Appendix A for schematic descriptions of our attacks and Table 4.1 for their complexities.

4.8 Conclusion

In this chapter, we have applied our combined attacks to the block ciphers SHACAL-1, SHACAL-2 and AES. We have presented

- a related-key rectangle attack on the full 80-round SHACAL-1;
- a square-nonlinear attack on 28-round SHACAL-2;
- a differential-nonlinear attack on 32-round SHACAL-2;
- a related-key differential-nonlinear attack on 35-round SHACAL-2;
- a related-key rectangle attack on 42-round SHACAL-2;
- a related-key rectangle attack on 10-round AES-192;
- a related-key rectangle attack on 9-round AES-256.

Our differential-nonlinear attack on 32-round SHACAL-2 leads to the best known attack on reduced SHACAL-2 that uses a single key and our related-key rectangle attacks on the full 80-round SHACAL-1, 42-round SHACAL-2 and 10-round AES-192 lead to the first known attack on the full SHACAL-1 and the best known attacks on SHACAL-2 and AES-192 that use related keys. They show the usefulness of our combined attacks in the security analysis of block ciphers.

We believe that the security against our combined attacks should be considered when one tries to design secure block ciphers.

Chapter 5

Applications to Hash Functions in Encryption Mode

5.1 Introduction

Recently, Biham et al. and Wang et al. have published several important cryptanalytic articles [9, 10, 126, 124, 127, 125] that demonstrate efficient collision search algorithms for the MD4-family of hash functions. In particular, the newly proposed neutral-bit and message modification techniques make it possible to significantly improve previous known collision attacks on MD4, MD5, HAVAL, RIPEMD, SHA-0 and SHA-1.

There have also been several cryptanalytic articles which investigate non-randomness of the compression functions of MD5 and HAVAL in encryption mode [116, 128]: differential cryptanalysis has been applied to show non-randomness for their outputs.

In this chapter, we check the security of the encryption modes of MD4, MD5 and HAVAL against the related-key boomerang attack, and we compare our results with the previous ones in terms of distinguishing attacks. Using the related-key boomerang attack we can distinguish the encryption modes of MD4, MD5 and 4-pass HAVAL from a randomly chosen cipher in practice. Furthermore, we can distinguish them more efficiently for a large class of weak keys (i.e., special subset of messages in hash mode). See Table 5.1 for a summary of our results and a comparison with the previous attacks.

Table 5.1: Distinguishing Attacks of MD4, MD5, HAVAL in Encryption Mode

Primitive	Attack (#K)	Data Complexity	#WK	Source
MD4	**B(2)**†	$\mathbf{2^{18}}$**RK-CP/**$\mathbf{2^{18}}$**RK-ACC**	·	(Sect. 5.2)
	B(2)†	**2RK-CP/2RK-ACC**	$\mathbf{2^{320}}$	(Sect. 5.2)
	B(4)†	$\mathbf{2^{6}}$**RK-CP/**$\mathbf{2^{6}}$**RK-ACC**	·	(Sect. 5.2)
	B(4)†	**2RK-CP/2RK-ACC**	$\mathbf{2^{384}}$	(Sect. 5.2)
MD5	D(1)	2^{50}CP	·	[116]
	B(2)	$2^{80.6}$RK-CP/$2^{78.6}$RK-ACC	·	(Sect. 5.3)
	B(2)†	**12RK-CP/12RK-ACC**	$\mathbf{2^{96}}$	(Sect. 5.3)
	B(4)†	$\mathbf{2^{13.6}}$**RK-CP/**$\mathbf{2^{11.6}}$**RK-ACC**	·	(Sect. 5.3)
	B(4)†	**6RK-CP/6RK-ACC**	$\mathbf{2^{352}}$	(Sect. 5.3)
HAVAL (4 passes)	D(1)	2^{127}CP	·	[128]
	B(2)†	$\mathbf{2^{37.9}}$**RK-CP/**$\mathbf{2^{35.9}}$**RK-ACC**	·	(Sect. 5.3)
	B(2)†	$\mathbf{2^{12.3}}$**RK-CP/**$\mathbf{2^{12.3}}$**RK-ACC**	$\mathbf{2^{576}}$	(Sect. 5.3)
	B(4)†	$\mathbf{2^{11.6}}$**RK-CP/**$\mathbf{2^{9.6}}$**RK-ACC**	·	(Sect. 5.3)
	B(4)†	**32RK-CP/32RK-ACC**	$\mathbf{2^{896}}$	(Sect. 5.3)
HAVAL (5 passes)	D(1)	2^{170}CP	·	[128]
	B(2)	$2^{127.9}$RK-CP/$2^{125.9}$RK-ACC	·	(Sect. 5.3)
	B(4)	2^{63}RK-CP/2^{61}RK-ACC	·	(Sect. 5.3)

†: The attack can be implemented in real time,
#K: Number of Keys, #WK: Number of Weak Keys,
D: Differential, B: related-key Boomerang, RK: Related-Key,
CP: Chosen Plaintexts, ACC: Adaptively Chosen Ciphertexts,
Time complexity is the same as the amount of data complexity.

5.2 Related-Key Boomerang Attacks on MD4

In MD4 the message expansion algorithm is a linear function: in each pass every message word is used exactly once in a specified order. It means that in the encryption mode of MD4 the key scheduling algorithm is the same linear function as the message expansion algorithm of MD4. We exploit this simple linear key scheduling algorithm in our distinguishers. The main idea behind our constructions of related-key boomerang distinguishers based on two related keys (TYPE 1, in which consecutive two related-key differentials are used) is to give a nonzero difference in one key word for which the interval between the first and third passes is as wide as possible. Let the round numbers involved in such a key word in the three passes be r_1, r_2 and r_3. Then, we can make probability-one differentials

Table 5.2: Boomerang Distinguishers of MD4 (Two Related Keys)

i	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔK^i	Prob.
0	0	e_{31}	0	0	0	1
1	0	0	e_{31}	0	0	2^{-1}
2	0	0	0	e_{31}	0	2^{-1}
3	e_{31}	0	0	0	$e_{31}(=\Delta K^3)$	1
4	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
27	0	0	0	0	0	1
28	0	0	0	0		$p^* = 2^{-2}$
28	e_{31}	0	0	0	$e_{31}(=\Delta K^3)$	1
29	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
43	0	0	0	0	0	1
44	0	0	0	0	$e_{31}(=\Delta K^3)$	1
45	0	e_2	0	0	0	2^{-1}
46	0	e_{11}	e_2	0	0	2^{-2}
47	0	$e_{13,22}$	e_{11}	e_2	0	2^{-4}
48	e_2	$e_{5,17,26,28}$	$e_{13,22}$	e_{11}		$q^* = 2^{-7}$
BOO-2	$(0 \rightarrow 27)$, $(47 \rightarrow 28)^2$, $(27 \rightarrow 3)$					Pr[BOO-2] $\approx 2^{-16}$
BOO^W*-2*	Fixed $K^{0,1,2,7,11,15}$, $(3 \rightarrow 27)$, $(44 \rightarrow 28)^2$, $(27 \rightarrow 3)$					Pr[BOO-2] = 1

for rounds r_1-r_2' and r_2'-r_3 by giving appropriate input differences α (to round 0) and γ (to round r_2'), respectively, where r_2' is a specific integer between r_1 and r_2. Therefore, in order to find distinguishers with high probablilities we should find one key word for which the interval of r_1 and r_3 is as wide as possible.

In our observation giving a nonzero difference in the 3-rd key word provides the best probabilities to our distinguishers, which are described as follows. In MD4 there exist a related-key differential $(0, e_{31}, 0, 0) \rightarrow (0, 0, 0, 0)$ for rounds 0-27 with probability $p^* = 2^{-2}$ and a related-key differential $(e_{31}, 0, 0, 0) \rightarrow (e_2, e_{5,17,26,28}, e_{13,22}, e_{11})$ for rounds 28-47 with probability $q^* = 2^{-7}$ under the key difference $\Delta K = (0, 0, 0, \Delta K^3 = e_{31}, 0, \cdots, 0)$. See Table 5.2 for more details. The notation used in Table 5.2 is essential in our distinguishing attacks. The *BOO-2* row represents a probability of a related-key boomerang distinguisher of TYPE 1, denoted $Pr[BOO-2]$, and the BOO^W*-2* row represents a weak key class as well as a probability of a related-key boomerang distinguisher of TYPE 1 under a weak key class. The notation $(r \rightarrow r')^l$ means related-key differentials for rounds from r to r' used in our distinguishers, where $l = 1$ or 2. Here, the superscript l represents how many times related-key differentials are used in our distinguishers. Note that if $r > r'$ then the related-key differential works through the decryption process. In this chapter, $Pr[BOO-2]$ and $Pr[BOO-4]$

represent the probabilities of the related-key boomerang distinguishers of TYPE 1 and TYPE 3 that take all possible differences after the first subcipher (cf. Note 1 in Sect. 3.4.4).

In order to estimate $Pr[BOO\text{–}2]$ we have carried out experiments on a number of related keys with 2^{23} chosen plaintext pairs and 2^{23} adaptively chosen ciphertext pairs each and we have observed 136, 115, 136, 125, 132, 130, 132, 131, 119, 144, $\cdots$ boomerangs returning for each related-key. This simulation result confirms that the probability $Pr[BOO\text{–}2]$ is approximately 2^{-16} (which can be also calculated from the probabilities of related-key differentials in Table 5.2).

Based on the distinguisher described in Table 5.2, we can exploit a boomerang technique to distinguish MD4 from a random cipher. In a boomerang technique we use $Pr[BOO\text{–}2] \approx 2^{-16}$. Since we use related-key differentials for rounds 27-3 in the upper right in Fig. 3.5, our desired α before round 0 is any one of the differences which can be derived from the input difference of round 3, $(e_{31}, 0, 0, 0)$, through the inverse direction. It is easy to see that all the possible $\alpha's$ are $(0, e_{31}, 0, 0)$, $(e_{31}, e_{31}, 0, 0)$, $(0, e_{31}, e_{31}, e_{31})$ and $(e_{31}, e_{31}, e_{31}, e_{31})$. We denote the set of all these possible $\alpha's$ by $\mathcal{I}$. Our distinguishing attack on the encryption mode of MD4 is as follows:

1. Prepare 2^{17} plaintext pairs (P_i, P_i^*), $i = 0, 1, \cdots, 2^{17} - 1$ with difference $(0, e_{31}, 0, 0)$ and $c_{31} = d_{31} = 0$, where c_j and d_j represent the j-th bits of words C and D of P_i, respectively (the $c_{31} = d_{31} = 0$ condition is required for our distinguisher).

2. Obtain the 2^{17} corresponding ciphertext pairs (C_i, C_i^*), i.e., $C_i = E_K(P_i)$ and $C_i^* = E_{K^*}(P_i^*)$, where E is either MD4 or a random cipher and $K \oplus K^* = (0, 0, 0, \Delta K^3 = e_{31}, 0, \cdots, 0)$.

3. Calculate $C_i' = C_i \oplus \delta$ and $C_i'^* = C_i^* \oplus \delta$, where $\delta = (e_2, e_{5,17,26,28}, e_{13,22}, e_{11})$, and obtain the 2^{17} corresponding plaintext pairs $(P_i', P_i'^*)$, i.e., $P_i' = E_K^{-1}(C_i')$ and $P_i'^* = E_{K^*}^{-1}(C_i'^*)$.

4. If there exists at least one plaintext pair such that $P_i' \oplus P_i'^* \in \mathcal{I}$ for $0 \leq i \leq 2^{17} - 1$, we identify E= MD4. Otherwise, we identify E= a randomly chosen cipher.

Since our related-key boomerang distinguisher has a probability of $2^{-2} \cdot (2^{-7})^2 = 2^{-16}$, if E is MD4, this attack will succeed with a probability of $1 - (1 - 2^{-16})^{2^{17}} \approx 0.86$. In order to verify this estimation we have performed hundreds of simulations using 2^{17} chosen plaintext pairs and 2^{17} adaptively chosen ciphertext pairs each (in each simulation we used randomly chosen related keys and plaintext/ciphetext pairs). In our simulations we could check that on

average about 88 among 100 tests satisfy the above distinguishing attack. This result is quite close to our estimation.

On the other hand, if E is a randomly chosen cipher, the probability that each plaintext pair satisfies one of the four α's is $\frac{4}{2^{128}} = 2^{-126}$, so, in this case this attack will succeed with a probability of $(1-2^{-126})^{2^{17}} \approx 1$. Therefore, the success rate of this attack is about $\frac{1}{2} \cdot 0.86 + \frac{1}{2} \cdot 1 = 0.93$.

Moreover, we can increase the boomerang probability from 2^{-16} to 1 by using some weak key class. Assume that the first three and the last three round keys $K^0, K^1, K^2, K^7, K^{11}$ and K^{15} are fixed and known to the cryptanalyst. Then we can use $p^* = 1$ for rounds 3-27 and $q^* = 1$ for rounds 44-28 in our attack under the weak key class assumption. Below we describe our distinguishing attack on the encryption mode of MD4 with a weak key using the related-key boomerang distinguisher.

1. Choose one input pair (X, X^*) of round 3 with difference $(e_{31}, 0, 0, 0)$ and calculate the corresponding plaintext pair (P, P^*) by using the known keys $K^0, K^1, K^2, K^{*0}, K^{*1}, K^{*2}$ and MD4.

2. Obtain the corresponding ciphertext pair (C, C^*), i.e., $C = E_K(P)$ and $C^* = E_{K^*}(P^*)$, where E is either MD4 or a random cipher and $K \oplus K^* = (0, 0, 0, \Delta K^3 = e_{31}, 0, \cdots, 0)$.

3. Calculate the corresponding input pair (Y, Y^*) of round 45 with the known keys $K^7, K^{11}, K^{15}, K^{*7}, K^{*11}, K^{*15}$ and MD4, and calculate $Y' = Y \oplus (0, e_2, 0, 0)$ and $Y'^* = Y^* \oplus (0, e_2, 0, 0)$. Again, calculate the corresponding ciphertext pair (C', C'^*) by using the known keys $K^7, K^{11}, K^{15}, K^{*7}, K^{*11}, K^{*15}$ and MD4.

4. Obtain the corresponding plaintext pair (P', P'^*), i.e., $P' = E_K^{-1}(C')$ and $P'^* = E_{K^*}^{-1}(C'^*)$.

5. Calculate the corresponding input pair (X', X'^*) of round 3 by using the known keys $K^0, K^1, K^2, K^{*0}, K^{*1}, K^{*2}$ and MD4.

6. If $X' \oplus X'^* = (e_{31}, 0, 0, 0)$, we identify E= MD4. Otherwise, we identify E= a randomly chosen cipher.

If E is MD4, this attack will succeed with probability one (we have checked with thousands of simulations that this attack always works for MD4), but if E is a randomly chosen cipher, this attack will succeed with probability $1 - 2^{-128}$. Therefore, the success rate of this attack is very close to 1.

Similarly, we can construct related-key boomerang distinguishers of TYPE 3 based on four related keys and distinguish MD4 from a randomly chosen cipher

by using them. As a compensation of the use of four related keys, these attacks are more efficient than those with two related keys. See Table B.1 in Appendix B for our distinguisher and Table 5.1 for our results. In order to estimate $Pr[BOO{-}4]$ we have carried out experiments on a number of related keys with 2^{11} chosen plaintext pairs and 2^{11} adaptively chosen ciphertext pairs each and we have observed 121, 122, 118, 132, 113 144, 141, 134, 135, 113, $\cdots$ boomerangs returning for each related-key. It confirms that the probability $Pr[BOO{-}4]$ is approximately 2^{-4}. We have also performed hundreds of simulations using 2^5 chosen plaintext pairs and 2^5 adaptively chosen ciphertext pairs in order to check the success rate of our distinguishing attack $1-(1-2^{-4})^{2^5} \approx 0.87$ when $E =$ MD4. In our simulations we could check that on average about 88 among 100 tests satisfy the distinguishing attack when $E =$ MD4.

5.3 Related-Key Boomerang Attacks on MD5 and HAVAL

Similarly, in the MD5 and HAVAL attacks, we first find consecutive two related-key differentials with high probabilities which are independent of each other, and then we estimate the probability $Pr[BOO{-}k]$ on the basis of those differentials by a series of simulations, where k is the number of source keys (k is equal to 2 or 4). As for 5-pass HAVAL, we can carry out an experiment on a reduced-round variant (which is truncated for the first and the last several rounds) to get $Pr[BOO{-}k]$ for the reduced variant and then we can use the obtained value as well as probabilities for the truncated rounds of the consecutive two related-key differentials (which were found in the first stage) to estimate $Pr[BOO{-}2]$ for the full 5-pass HAVAL. See Appendix B for our distinguishers of MD5 and HAVAL.

The related-key boomerang attacks on MD5 and HAVAL are slightly different from those of MD4. The boomerang attack works by finding not only a chosen plaintext pair but also an adaptively chosen ciphertext pair that satisfy a boomerang distinguisher. For MD5 and HAVAL, once we obtain a ciphertext pair by asking for the encryption of a chosen plaintext pair, we know whether or not the adaptively chosen ciphertexts can be a boomerang candidate. For example, consider the boomerang distinguisher of Table B.2 in Appendix B. Assume that the ciphertext pair obtained by asking for the encryption of a chosen plaintext pair is (C, C^*) and (a_{31}, c_{31}, d_{31}) of C or $(a^*_{31}, c^*_{31}, d^*_{31})$ of C^* is in $\{(0,0,0),(0,1,0),(1,0,1),(1,1,1)\}$. Then the adaptively chosen ciphertext pair $(C \oplus \delta, C^* \oplus \delta)$ cannot satisfy our boomerang distinguisher, where $\delta = (e_5, e_5, e_5, e_5)$. That is, in this case ΔA^{63} cannot be of the form e_5 since the difference induced by the Boolean function of the last round is 0 for $(C \oplus \delta, C^* \oplus \delta)$. (Note that in the boomerang attacks on MD4 we cannot use this procedure since

the Boolean function used in the last round of MD4 is linear.) This is the reason why the required number of queries for the decryption process is smaller than that for the encryption process.

We have also performed a series of simulations to verify related-key boomerang attacks on the encryption modes of MD5 and HAVAL, which are indicated in Table 5.1 by the symbol †. According to our probabilities of related-key boomerang distinguishers in Appendix B and the data complexity in Table 5.1, related-key boomerang attacks work with a success rate of about 0.87, when E is MD5 or HAVAL. During our simulations, we have observed that the simulation results correspond to our estimation of success rate. As an example of our simulations, we give in Appendix B.4 a related-key quartet, a chosen plaintext pair and an adaptively chosen ciphertext pair of MD5 obtained by the boomerang distinguisher described in Table B.3.

5.4 Conclusion

In this chapter, we have applied the related-key boomerang attack to the encryption modes of MD4, MD5 and HAVAL. The MD4, MD5 and HAVAL used in encryption modes are all vulnerable to the related-key boomerang attack. The presented attacks have been experimentally tested and run milliseconds on a PC.

Our results show that one should be very careful when using existing hash functions in encryption mode.

Chapter 6

Applications to MAC Algorithms

6.1 Introduction

HMAC is proved to be a pseudorandom function under the assumption that the compression function of the underlying hash function is a pseudorandom function [2] (the security proof of pseudorandomness provides the MAC security [4]). However, this does not guarantee the security of HMAC if it is instantiated with a specific cryptographic hash function such as MD5 or SHA-1. The recent attacks of Wang et al. [126, 124, 125, 127, 129] and Biham et al. [9, 10] have undermined the confidence in the most popular collision resistant hash functions such as MD5 and SHA-1. However, it is widely assumed that these attacks have no impact on the security of MAC algorithms based on these hash functions such as HMAC since they use a keyed initial value.

This thesis is the first work which presents a detailed analysis of distinguishing and forgery attacks on HMAC based on MD5, SHA-1 and other MDx-type hash functions. Our results allow to quantify to which extent the vulnerabilities of these hash functions carry over to the HMAC construction. This is achieved by the introduction of two novel distinguishers of the general structure of HMAC. We use a message pair which induces a collision in its corresponding MAC pair for designing *a differential distinguisher* of HMAC and also use a message quartet which induces two collisions in its corresponding MAC quartet for designing *a rectangle distinguisher*[1] of HMAC. With these two distinguishers we discuss the

[1]The related-key rectangle technique is applied to HMAC, but our distinguisher works on HMAC with a single secret key. For clarification, we use the terminology a *rectangle distinguisher* rather than a *related-key rectangle distinguisher* for HMAC.

security of HMAC based on HAVAL [130], MD4 [111], MD5 [112], SHA-0 [40] and SHA-1 [41].

First, we construct new differentials of the full 3-pass HAVAL and reduced MD5 to form rectangle distinguishers of HMAC, and we use them to distinguish HMAC with the full 3-pass HAVAL and reduced MD5 from HMAC with a random function. Second, we investigate how effectively the differentials of MD4, SHA-0 and SHA-1 found by Wang et al. [126, 124, 125, 127, 129] and Biham et al. [9, 10] are applied to our differential and rectangle distinguishers in HMAC. After converting their differentials into our differential and rectangle distinguishers, we devise distinguishing and forgery attacks on HMAC based on reduced or full versions of MD4, SHA-0 and SHA-1. In particular, we show how to distinguish HMAC with the full SHA-0 and MD4 from HMAC with a random function and present a forgery attack on HMAC with the full MD4. See for details of the results Table 6.3 in Sect. 6.5 (the function h_2 and the probabilities $\hat{p}$ and q in Table 6.3 will be defined in the following sections). Our distinguishing and forgery attacks can be mounted on NMAC based on HAVAL, MD4, MD5, SHA-0 and SHA-1 with the same complexity. Furthermore, we show that our differential and rectangle distinguishers can lead to second-preimage attacks on HMAC and NMAC.

6.2 Some General Attacks on HMAC

Using the birthday paradox we can induce a general distinguishing attack on HMAC as follows [108]:

1. Collect $2^{l/2}$ randomly chosen messages with a t-bit length, denoted M_i, and ask for their MAC values, denoted C_i (recall that the bit-length of the MAC values is l).

2. Find message pairs M_j and M_k such that $C_j = C_k$.

3. For each of (M_j, M_k) pairs such that $C_j = C_k$, ask for a MAC pair of $M_j||P$ and $M_k||P$, where P is some non-empty string. If there is at least one MAC pair that collides in this step, output the MAC algorithm = HMAC.

This attack requires about $2^{l/2}$ messages and works with a probability of 0.63 by the birthday paradox when the MAC algorithm is HMAC. This is due to the fact that if there exists at least one message pair (M_j, M_k) such that their outputs of h_2 or H_2 are the same, this attack always works. This attack can also easily be converted into a general forgery attack on HMAC. Once we get a MAC pair that collides in Step 3, we again ask for the corresponding MAC of $M_j||P||P'$,

denoted C, where P' is some non-empty string. We can then construct a forgery, i.e., a new message $M_k||P||P'$ with a valid MAC: C with the same success rate.

These general attacks show that interesting distinguishing and forgery attacks on HMAC need to require fewer than $2^{l/2}$ message queries. We thus consider attacks that distinguish HMAC from a random function, and forgery attacks on HMAC which work with a data complexity less than $2^{l/2}$ messages. In addition to these two kinds of attacks, we also consider attacks of distinguishing instantiated HMAC (by existing hash functions) from HMAC with a random function. In this distinguishing attack, it does not matter whether or not they require more than $2^{l/2}$ message queries, since there does not exist a general attack based on the birthday paradox which can distinguish HMAC with existing hash functions from HMAC with a random function. In order to clarify the difference, we denote the first and second distinguishing attacks by *distinguishing-R* and *distinguishing-H attacks*, respectively. The distinguishing-R attack is useful when the cryptanalyst wants to check whether output strings are produced from HMAC (in this case, the cryptanalyst does not know whether the output producing algorithm is HMAC), while the distinguishing-H attack is useful when the cryptanalyst wants to check which cryptographic hash function is embedded in HMAC (in this case, the cryptanalyst somehow already knows that the output producing algorithm is HMAC, for instance, by the distinguishing-R attack, but he does not know the underlying hash function in HMAC).

6.3 Differential and Rectangle Distinguishers of HMAC

In this section, we present two distinguishers of the general structure of HMAC, which can lead to distinguishing or forgery attacks if HMAC is instantiated with some cryptographic hash function with a slow difference propagation. These two distinguishers, called *differential* and *rectangle distinguishers*, are both built based on internal collisions.[2] We focus on HMAC with one-block messages, which is the main target in our attacks.

6.3.1 Differential Distinguisher of HMAC

By using MAC collisions we construct a differential distinguisher of HMAC. It works as follows:

- Choose a message M_i at random and compute another message $M_i' = M_i \oplus \alpha$, where M_i has the same length as α ($\neq 0$).

[2]The internal collisions represent the collisions of the output pairs for the function h_2.

- With a chosen message attack scenario, obtain the MAC values $C_i = \text{HMAC}(K, M_i)$ and $C_i' = \text{HMAC}(K, M_i')$.
- Check if $C_i \oplus C_i' = 0$.

For HMAC, the last test holds with a probability of approximately $q = Pr_X[h_2(IV', X) \oplus h_2(IV', X \oplus \alpha) = 0]$, where $IV' = h_1(IV, K \oplus ipad)$. In fact, the last test holds with a probability of approximately $q + (1 - q) \cdot 2^{-l}$. Because even if the M_i and M_i' do not cause a collision after the function h_2, their MAC values can still have the same value. However, in the computation of the probability for our differential distinguisher we do not consider this case.

On the other hand, for a random function (resp. HMAC with a random function[3]), the last test holds with a probability of 2^{-l} (resp. 2^{-l+1}). Hence, we have the following differential distinguisher of HMAC.

Theorem 6.3.1 *[A Differential Distinguisher of HMAC] HMAC can be distinguished from a random function (resp. HMAC with a random function) if $q > 2^{-l}$ (resp. $q > 2^{-l+1}$), where $q = Pr_X[h_2(IV', X) \oplus h_2(IV', X \oplus \alpha) = 0]$ and $IV' = h_1(IV, K \oplus ipad)$.*

In order for this differential distinguisher to be used in distinguishing-R and forgery attacks, the probability q should be larger than $2^{-l/2}$, which allows these attacks to work with fewer than $2^{l/2}$ message queries (details are described in Sect. 6.5).

6.3.2 Rectangle Distinguisher of HMAC

The rectangle distinguisher of HMAC can be built by the rectangle technique which is widely used in analyzing block ciphers (cf. Sect. 3.4). In block ciphers the rectangle technique can be mounted based on their bijectivity. However, in MACs it can exploit the non-bijectivity, i.e., two different messages may correspond to the same MAC value or the same intermediate value (an internal collision). We use this non-bijective property to devise our rectangle distinguisher of HMAC. Our rectangle distinguisher of HMAC works as follows (refer to Fig. 6.1):

- Choose two messages M_i and M_j at random and compute two other messages $M_i' = M_i \oplus \alpha$ and $M_j' = M_j \oplus \alpha$, where M_i and M_j both have the same length as α ($\neq 0$).

[3]The factor that makes the output distribution of HMAC with a random function different from that of a random function is a collision after the function h_2 of HMAC. It follows that for each message pair, this internal collision occurs with probability 2^{-l}. Hence, for HMAC with a random function the last test holds with probability $2^{-l} + 2^{-l}(1 - 2^{-l}) \approx 2^{-l+1}$.

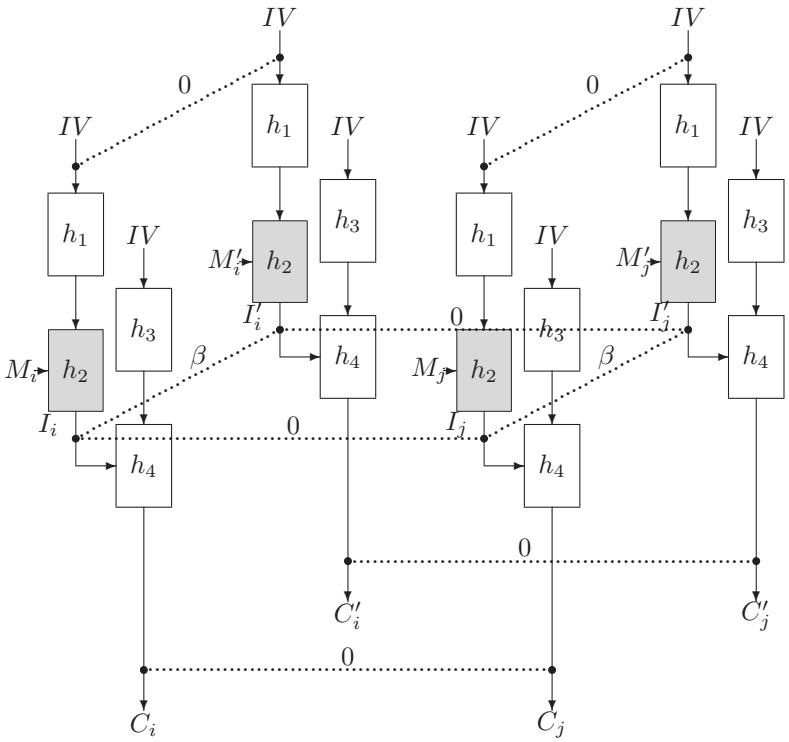


Figure 6.1: Rectangle Distinguisher of HMAC ($M_i \oplus M'_i = M_j \oplus M'_j = \alpha$)

- With a chosen message attack scenario, obtain the MAC values C_i = HMAC(K, M_i), C'_i = HMAC(K, M'_i), C_j = HMAC(K, M_j) and C'_j = HMAC(K, M'_j).
- Check if $C_i \oplus C_j = C'_i \oplus C'_j = 0$ or $C_i \oplus C'_j = C'_i \oplus C_j = 0$.

We denote by I_i, I'_i, I_j and I'_j the outputs of $h_2 \circ h_1$ for the messages M_i, M'_i, M_j and M'_j, respectively (see Fig. 6.1). Note that in Fig. 6.1 $K \oplus ipad$ and $K \oplus opad$ are inserted into the message parts of the functions h_1 and h_3, respectively. In order to compute the probability to satisfy the last test we should consider the following probabilities: $p = Pr_X[h_2(IV', X) \oplus h_2(IV', X \oplus \alpha) = \beta]$ and $\hat{p} = \sqrt{\sum_\beta (p^2)}$, where $IV' = h_1(IV, K \oplus ipad)$.

By the definition of the probability p, we get $I_i \oplus I'_i = I_j \oplus I'_j = \beta$ with probability p^2. Since the function h_2 is not a permutation (here, the domain of h_2 is the message space and its co-domain is the space of hash values), we expect $I_i \oplus I_j = 0$ with probability 2^{-l} under the assumption that the output values of h_2 are distributed uniformly at random. Once we get $I_i \oplus I'_i = I_j \oplus I'_j = \beta$ and $I_i \oplus I_j = 0$, we have the following equation:

$$I'_i \oplus I'_j = (I_i \oplus \beta) \oplus (I_j \oplus \beta) = I_i \oplus I_j = 0\,.$$

These equations allow us to get $C_i \oplus C_j = C'_i \oplus C'_j = 0$ and thus the probability of satisfying $C_i \oplus C_j = C'_i \oplus C'_j = 0$ is approximately

$$\sum_{\beta} p^2 \cdot 2^{-l} = \hat{p}^2 \cdot 2^{-l}\,.$$

(Note that the probability of satisfying $C_i \oplus C_j = C'_i \oplus C'_j = 0$ is slightly larger than $\hat{p}^2 \cdot 2^{-l}$. Because even if the I_i and I_j (or the I'_i and I'_j) are not the same, still there is a chance to have $C_i \oplus C_j = C'_i \oplus C'_j = 0$. However, we believe that a simplified analysis is sufficient for the computation of the probability for our rectangle distinguisher.) Similarly, we get $I_i \oplus I'_j = 0$ with a probability of 2^{-l} and thus $C_i \oplus C'_j = C'_i \oplus C_j = 0$ holds with the same probability $\hat{p}^2 \cdot 2^{-l}$.

On the other hand, for a random function (resp. HMAC with a random function), $C_i \oplus C_j = C'_i \oplus C'_j = 0$ and $C_i \oplus C'_j = C'_i \oplus C_j = 0$ hold with a probability of approximately 2^{-2l} (resp. 2^{-2l+2}), respectively, since each requires a $2l$-bit restriction (resp. a $(2l-2)$-bit restriction[4]) to be satisfied. Hence, we have the following rectangle distinguisher of HMAC.

Theorem 6.3.2 *[A Rectangle Distinguisher of HMAC] HMAC can be distinguished from a random function (resp. HMAC with a random function) if* $\hat{p}^2 \cdot 2^{-l} > 2^{-2l}$*, i.e.,* $\hat{p} > 2^{-l/2}$ *(resp.* $\hat{p}^2 \cdot 2^{-l} > 2^{-2l+2}$*, i.e.,* $\hat{p} > 2^{-l/2+1}$*), where* $\hat{p} = \sqrt{\sum_{\beta}(p^2)}$*,* $p = Pr_X[h_2(IV', X) \oplus h_2(IV', X \oplus \alpha) = \beta]$ *and* $IV' = h_1(IV, K \oplus ipad)$*.*

Our rectangle distinguisher cannot be used in distinguishing-R and in forgery attacks, since its required data complexity is always larger than $2^{l/2}$ messages (details are described in Sect. 6.5). This is due to the fact that the rectangle probability is always less than or equal to 2^{-l}.

Unlike the differential distinguisher of HMAC, the rectangle distinguisher uses a number of differentials without any restriction for output differences, while its

[4] As mentioned before, for HMAC with a random function, a collision occurs with probability 2^{-l+1} and thus two collisions of the two pairs chosen form a MAC quartet occur with probability 2^{-2l+2}.

requirement to work is more expensive than that of the differential distinguisher, i.e., it uses probability $2^{-l/2}$ (or $2^{-l/2+1}$) instead of 2^{-l} (or 2^{-l+1}) for its comparison. If it is easy to get some nonzero output difference from the compression function of the underlying hash function, but it is difficult to get a zero output difference, i.e., a collision, then this rectangle distinguisher would be useful.

The success of our two distinguishers for HMAC depends significantly on the strength of h_2, which means the distinguishers do not depend strongly on the properties of h_1, h_3 and h_4. Even if h_1, h_3 and h_4 employ cryptographically strong compression functions (even iterated hash functions), our distinguishers can still work if h_2 has a weak difference propagation.

6.4 Differentials of HAVAL, MD4, MD5, SHA-0 and SHA-1

First, we check how many rounds of the compression functions of HAVAL, MD4, MD5, SHA-0 and SHA-1 can be used for h_2 in our rectangle distinguisher, i.e., we investigate for how many rounds of each compression function $\hat{p} > 2^{-l/2}$ holds. Second, we discuss how to extend one-block messages (corresponding to h_2) into multi-block messages (corresponding to H_2) in order to apply them to our rectangle distinguisher. Third, we deal with differentials with probabilities q such that $q > 2^{-l}$ or $q > 2^{-l/2}$.

6.4.1 One-Block Differentials for Rectangle Distinguishers

In order to compute the number of rounds for each compression function such that $\hat{p} > 2^{-l/2}$, we investigate a differential with probability p from which the probability $\hat{p}$ can be estimated. We first consider the compression function of 3-pass HAVAL.

In the compression function of HAVAL we insert a one-bit difference to two message words to produce a collision after the first pass with a high probability. This enables us to get probability-one differentials through many rounds in the first and second passes. More precisely, if we denote by r_1, r_2, r_3, r_4, r_5 and r_6 the round numbers involved in two such message words in the three passes where $r_1 < r_2 < \cdots < r_6$, we can construct a 96-round differential with the following probability: for rounds 0-r_1 probability 1, for each of the rounds (r_1+1)-r_2 probability 2^{-1}, for rounds (r_2+1)-(r_3-1) probability 1, for each of the rounds r_3-r_4 probability 2^{-1}, for each of the rounds (r_4+1)-(r_5-1) probability 2^{-2}, for each of the rounds r_5-r_6 probability 2^{-3} and for each of the rounds (r_6+1)-95 probability 2^{-4} (this can be achieved by computing the differential probabilities derived from the differential distributions of Boolean functions and the use of both XOR and

modular additions). These probabilities may be slightly different according to in which message word between the two a difference 0x80000000 is given. But the total probability is the same: $2^{-(r_2-r_1+r_4-r_3+1+2(r_5-r_4-1)+3(r_6-r_5+1)+4(95-r_6))}$.

As a result of an exhaustive search,[5] inserting a one-bit difference to the third and eleventh message words provides the best probability $p = 2^{-102}$ (when $\Delta m^2 = e_{31}$ and $\Delta m^{10} = e_{20}$). See Tables 6.1 and 6.2 for more details. Note that we use the XOR difference as the measure of difference and in the computation of the probability p in Table 6.2 the modular additions of the unknown initial value and the last output value are considered. In our analysis we take into account the probability that the last output difference is preserved through the final modular additions.

In order to calculate $\hat{p}$ we should sum the square of the probability of all differentials with message difference α. However, it is computationally infeasible and thus we have carried out experiments on the last three rounds (rounds 93-95) to estimate a lower bound for $\hat{p}$ (our simulation is based on the assumption that chosen message pairs follow the first 93-round differential in Tables 6.1 and 6.2). For this work, we have randomly chosen a number of IVs with 2^{28} message pairs M_i, M_i^* and 2^{28} input pairs of round 93 I_i, I_i^* each and computed $M_i^{'} = M_i \oplus \alpha$, $M_i^{*'} = M_i^* \oplus \alpha$ and $I_i^{'} = I_i \oplus \delta$ and $I_i^{*'} = I_i^* \oplus \delta$, where α is the message difference and δ is the input difference of round 93 in Table 6.2. We have then encrypted through rounds 93-95 I_i, $I_i^{'}$, I_i^* and $I_i^{*'}$ with M_i, $M_i^{'}$, M_i^* and $M_i^{*'}$ to obtain outputs O_i, $O_i^{'}$, O_i^* and $O_i^{*'}$. Finally, we have checked if $(O_i + IV) \oplus (O_i^{'} + IV) = (O_i^* + IV) \oplus (O_i^{*'} + IV)$. In our experiments we have observed that the number of such quartets was ranging from 320 to 2130 for each IV. This simulation result suggests that the square of the probability $\hat{p}$ for rounds 93-95 is approximately $2^{-18.2}$ and thus we can estimate the probability $\hat{p} \approx 2^{-9.1} \cdot 2^{-90} = 2^{-99.1}$ since the differential probability for rounds 0-92 in Tables 6.1 and 6.2 is 2^{-90}. Furthermore, we can extend this differential up to 101 rounds such that $\hat{p} > 2^{-128}$. See Table 6.2 for this extension. We have also performed a series of simulations on the last two rounds and from the simulation result we can estimate $\hat{p} \approx 2^{-124.4}$ for rounds 0-101.

Similarly, we have investigated differentials for the compression function of MD5 with high probabilities by inserting a one-bit difference in two or three message words to produce a collision after the first pass. As a result, we can construct a 33-round differential on MD5 with probability $p = 2^{-56}$, which can be used to construct differentials with probability $\hat{p}$. See Table C.1 in Appendix C for details of our reduced MD5 differential. Our investigations on HAVAL and MD5 have started from the assumption that low-weight differentials work out best when we can not use neutral bits and message modifications. However,

[5] An exhaustive computer search has been performed over all possible r_1, r_2, r_3, r_4, r_5 and r_6 which can produce a collision after the first pass.

Table 6.1: Differential for Rounds 0-79 of HAVAL

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	ΔF^i	ΔG^i	ΔH^i	Δm^i	Prob.
0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	e_{31}	1
3	0	0	0	0	0	0	0	e_{31}	0	2^{-1}
4	0	0	0	0	0	0	e_{31}	0	0	2^{-1}
5	0	0	0	0	0	e_{31}	0	0	0	2^{-1}
6	0	0	0	0	e_{31}	0	0	0	0	2^{-1}
7	0	0	0	e_{31}	0	0	0	0	0	2^{-1}
8	0	0	e_{31}	0	0	0	0	0	0	2^{-1}
9	0	e_{31}	0	0	0	0	0	0	0	2^{-1}
10	e_{31}	0	0	0	0	0	0	0	e_{20}	2^{-1}
11	0	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
44	0	0	0	0	0	0	0	0	0	1
45	0	0	0	0	0	0	0	0	e_{20}	2^{-1}
46	0	0	0	0	0	0	0	e_{20}	0	2^{-1}
47	0	0	0	0	0	0	e_{20}	0	0	2^{-1}
48	0	0	0	0	0	e_{20}	0	0	0	2^{-1}
49	0	0	0	0	e_{20}	0	0	0	0	2^{-1}
50	0	0	0	e_{20}	0	0	0	0	0	2^{-1}
51	0	0	e_{20}	0	0	0	0	0	0	2^{-1}
52	0	e_{20}	0	0	0	0	0	0	0	2^{-1}
53	e_{20}	0	0	0	0	0	0	0	0	2^{-1}
54	0	0	0	0	0	0	0	e_9	0	2^{-1}
55	0	0	0	0	0	0	e_9	0	0	2^{-1}
56	0	0	0	0	0	e_9	0	0	0	2^{-1}
57	0	0	0	0	e_9	0	0	0	0	2^{-1}
58	0	0	0	e_9	0	0	0	0	0	2^{-1}
59	0	0	e_9	0	0	0	0	0	0	2^{-1}
60	0	e_9	0	0	0	0	0	0	e_{31}	2^{-1}
61	e_9	0	0	0	0	0	0	e_{31}	0	2^{-2}
62	0	0	0	0	0	0	e_{31}	e_{30}	0	2^{-2}
63	0	0	0	0	0	e_{31}	e_{30}	0	0	2^{-2}
64	0	0	0	0	e_{31}	e_{30}	0	0	0	2^{-2}
65	0	0	0	e_{31}	e_{30}	0	0	0	0	2^{-2}
66	0	0	e_{31}	e_{30}	0	0	0	0	0	2^{-2}
67	0	e_{31}	e_{30}	0	0	0	0	0	0	2^{-2}
68	e_{31}	e_{30}	0	0	0	0	0	0	0	2^{-2}
69	e_{30}	0	0	0	0	0	0	e_{20}	0	2^{-2}
70	0	0	0	0	0	0	e_{20}	e_{19}	0	2^{-2}
71	0	0	0	0	0	e_{20}	e_{19}	0	0	2^{-2}
72	0	0	0	0	e_{20}	e_{19}	0	0	0	2^{-2}
73	0	0	0	e_{20}	e_{19}	0	0	0	0	2^{-2}
74	0	0	e_{20}	e_{19}	0	0	0	0	0	2^{-2}
75	0	e_{20}	e_{19}	0	0	0	0	0	0	2^{-2}
76	e_{20}	e_{19}	0	0	0	0	0	0	0	2^{-2}
77	e_{19}	0	0	0	0	0	0	e_9	0	2^{-2}
78	0	0	0	0	0	0	e_9	e_8	0	2^{-2}
79	0	0	0	0	0	e_9	e_8	0	0	2^{-2}
80	0	0	0	0	e_9	e_8	0	0	0	

Table 6.2: Differential for Rounds 80-101 of HAVAL

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	ΔF^i	ΔG^i	ΔH^i	Δm^i	Prob.
80	0	0	0	0	e_9	e_8	0	0	0	2^{-2}
81	0	0	0	e_9	e_8	0	0	0	0	2^{-2}
82	0	0	e_9	e_8	0	0	0	0	0	2^{-2}
83	0	e_9	e_8	0	0	0	0	0	0	2^{-2}
84	e_9	e_8	0	0	0	0	0	0	0	2^{-2}
85	e_8	0	0	0	0	0	0	e_{30}	0	2^{-2}
86	0	0	0	0	0	0	e_{30}	e_{29}	0	2^{-2}
87	0	0	0	0	0	e_{30}	e_{29}	0	0	2^{-2}
88	0	0	0	0	e_{30}	e_{29}	0	0	0	2^{-2}
89	0	0	0	e_{30}	e_{29}	0	0	0	0	2^{-2}
90	0	0	e_{30}	e_{29}	0	0	0	0	0	2^{-2}
91	0	e_{30}	e_{29}	0	0	0	0	0	e_{20}	2^{-3}
92	e_{30}	e_{29}	0	0	0	0	0	e_{20}	0	2^{-3}
93	e_{29}	0	0	0	0	0	e_{20}	e_{19}	0	2^{-3}
94	0	0	0	0	0	e_{20}	e_{19}	e_{18}	0	2^{-3}
95	0	0	0	0	e_{20}	e_{19}	e_{18}	0	e_{31}	2^{-3}
96	0	0	0	e_{20}	e_{19}	e_{18}	0	e_{31}	0	2^{-4}
0-95	$p = 2^{-102}$, $\hat{p} = 2^{-99.1}$ (3-pass HAVAL)									
97	0	0	e_{20}	e_{19}	e_{18}	0	e_{31}	0	0	2^{-4}
98	0	e_{20}	e_{19}	e_{18}	0	e_{31}	0	0	0	2^{-4}
99	e_{20}	e_{19}	e_{18}	0	e_{31}	0	0	0	0	2^{-4}
100	e_{19}	e_{18}	0	e_{31}	0	0	0	e_9	e_{31}	2^{-4}
101	e_{18}	0	e_{31}	0	0	0	e_9	$e_{8,31}$	0	2^{-5}
102	0	e_{31}	0	0	0	e_9	$e_{8,31}$	e_7		
0-101	$p = 2^{-127}$, $\hat{p} = 2^{-124.4}$ (reduced 4-pass HAVAL)									

there is still a possibility that HAVAL and MD5 have stronger differentials which can be derived by other methods (refer to [26]).

For MD4, SHA-0 and SHA-1, we have used the previous differentials in our distinguishers, i.e., a 48-round differential for MD4 with probability 2^{-56} in [129], a 65-round differential for SHA-0 with probability 2^{-78} in [9, 10] and a 43-round differential for SHA-1 with probability 2^{-80} in [10]. The 43-round differential for SHA-1 is an extension of the 34-round differential described in [10], and the differential probabilities for SHA-0 and SHA-1 have been recomputed. The main difference in the computations of differential probabilities between [9, 10] and our analysis is the use of neutral bits. In SHA-0 and SHA-1 initial values are known, which enables us to use neutral bits on message pairs to improve differential probabilities. However, in our analysis of HMAC initial values are determined by a secret key K, which implies they are unknown. See Appendix C for the recomputed differentials of SHA-0 and SHA-1. We have also carried out the same experiments on the last few rounds to estimate each $\hat{p}$ and from

our simulations we can estimate $\hat{p} \approx 2^{-56}$, $2^{-47.6}$, 2^{-78} and $2^{-73.4}$ for 48-round MD4, 33-round MD5, 65-round SHA-0 and 43-round SHA-1, respectively.

6.4.2 Multi-Block Differentials for Rectangle Distinguishers

We now discuss the probability $\hat{p}$ for HMAC using multi-block messages. Assume that two multi-block messages M and M' inserted to H_2 are divided into n block sub-messages $M^1||M^2||\cdots||M^n$ and $M'^1||M'^2||\cdots||M'^n$ with difference $\alpha = \alpha_1 ||\alpha_2||\cdots||\alpha_n$. Then the initial values for M^i and M'^i are the same as the hash values of the underlying hash function for the sub-messages $M^1||M^2||\cdots||M^{i-1}$ and $M'^1||M'^2||\cdots||M'^{i-1}$ for $2 \leq i \leq n$. Recall that the initial values for M^1 and M'^1 are the same, namely the output of the compression function for $K \oplus ipad$. Assuming that for the i-th compression function of H_2 there exist differentials $\beta_{i-1} \rightarrow \beta_i$ with probability p_i under the message difference α_i, where $\beta_0 = 0$, i.e., for a sub-message pair (M^i, M'^i) the input difference β_{i-1} goes to output difference β_i with probability p_i, we get

$$\hat{p} = \sqrt{\sum_{\beta_1,\beta_2,\cdots,\beta_n} (p_1 \times p_2 \times \cdots \times p_n)^2}$$

under the message difference $\alpha_1||\alpha_2||\cdots||\alpha_n$. This is due to the fact that the initial value to the first compression function of the H_2 is not known. If the initial value is known, the probability $\hat{p}$ is much higher than when it is unknown since the known initial value allows us to find specific sub-messages M_i and M'_i with probability p_i to produce two outputs with difference β_i from $i = 1$ till $i = n$ in order. This method has been introduced in [126, 127, 125, 10]. So it is much more difficult to apply multi-block messages in our rectangle distinguisher of HMAC compared to the use of one-block messages (in terms of the same number of rounds of the compression function). A similar argument applies to multi-block differentials for differential distinguishers, hence we omit the details of multi-block differentials of HAVAL, MD4, MD5, SHA-0 and SHA-1.

Note that for HMAC with a random function that uses multi-block messages, the differential and rectangle tests pass with higher probabilities than when it uses one-block messages. This is due to the fact that if HMAC with a random function uses multi-block messages, there are more than one differential path to yield a collision after the function H_2 unlike when it uses one-block messages.

6.4.3 Differentials for Differential Distinguishers

As stated above, our differential distinguisher works based on a differential which causes a zero difference, i.e., a collision, after the function h_2. We use the forego-

ing differentials or the previously known differentials on MD4, SHA-0 and SHA-1 in our distinguishing and forgery attacks:

- For SHA-0, the 65-round differential with probability 2^{-78} in Tables C.2 and C.3 can be extended into a 82-round differential with probability 2^{-98} ($\approx q$), which causes a collision (this extended differential has appeared in [9], but the differential probability is lower than that in [9] since we cannot use neutral bits.)
- For SHA-1, the first 34-round differential with probability 2^{-52} in Tables C.4 and C.5 can be used as our differential distinguisher.
- For the full MD4, there exists a differential with probability 2^{-56} ($\approx q$), which causes a zero output difference from an unknown initial value [129].
- For the full SHA-0, there exists a differential with probability 2^{-107} ($\approx q$), which causes a zero output difference from an unknown initial value [124, 127].

6.5 Distinguishing and Forgery Attacks on HMAC Based on HAVAL, MD4, MD5, SHA-0 and SHA-1

We use the probabilities $\hat{p}$ and q to show two distinguishing attacks and a forgery attack on the HMAC construction, and apply these attacks to HMAC based on HAVAL, MD4, MD5, SHA-0 and SHA-1.

Our first distinguishing attack on HMAC using $\hat{p}$ and a rectangle distinguisher is described as follows:

1. Collect $2^{(l+1)/2} \cdot \hat{p}^{-1}$ message pairs (M_i, M_i') with difference α, where all the M_i and M_i' have the same bit-length t.
2. With a chosen message attack scenario, ask for MAC pairs of all the (M_i, M_i'). We denote the corresponding MAC pairs by (C_i, C_i'). We assume that the MAC algorithm is either an instantiated HMAC or a random function (or HMAC with a random function) which maps t bits to l bits.
3. Check if $C_i \oplus C_j = C_i' \oplus C_j' = 0$ or $C_i \oplus C_j' = C_i' \oplus C_j = 0$ for all $i,\ j$ such that $1 \leq i < j \leq 2^{(l+1)/2} \cdot \hat{p}^{-1}$. If there is at least one MAC quartet that satisfies this test, output the MAC algorithm = HMAC, otherwise, output the MAC algorithm = a random function (or HMAC with a random function).

The data complexity of this attack is $2^{1+(l+1)/2} \cdot \hat{p}^{-1}$ chosen messages and this attack requires a memory of $2^{1+(l+1)/2} \cdot \hat{p}^{-1}$ l-bit blocks for storing all the MAC values. The time complexity of this attack is dominated by Step 1 (the data collection time) and Step 3, which seeks colliding MAC quartets. Since it can be done efficiently by sorting the MAC pairs (C_i, C_i')'s by C_i's, the time complexity of this attack is thus a fraction of the time required to compute the MAC values for the chosen messages (Step 1).

The success rate of this attack is calculated as follows. In Step 1 the $2^{(l+1)/2} \cdot \hat{p}^{-1}$ message pairs form $2^l \cdot \hat{p}^{-2}$ message quartets $((M_i, M_i'),(M_j, M_j'))$ corresponding to MAC quartets $((C_i, C_i'),(C_j, C_j'))$ for $1 \leq i < j \leq 2^{(l+1)/2} \cdot \hat{p}^{-1}$. Since for HMAC $C_i \oplus C_j = C_i' \oplus C_j' = 0$ holds with a probability of $2^{-l} \cdot \hat{p}^2$, and $C_i \oplus C_j' = C_i' \oplus C_j = 0$ also holds with the same probability (this probability has been computed in Sect. 6.3), the expected number of MAC quartets satisfying the last test is 2 $(= (2^l \cdot \hat{p}^{-2}) \cdot (2^{-l} \cdot \hat{p}^2) + (2^l \cdot \hat{p}^{-2}) \cdot (2^{-l} \cdot \hat{p}^2))$. On the other hand, for a random function, $C_i \oplus C_j = C_i' \oplus C_j' = 0$ holds with a probability of 2^{-2l}, and $C_i \oplus C_j' = C_i' \oplus C_j = 0$ also holds with the same probability and thus the expectation of satisfying the test is $2^{-l+1} \cdot (\hat{p}^{-2}) (= 2^{-2l} \cdot (2^l \cdot \hat{p}^{-2}) + 2^{-2l} \cdot (2^l \cdot \hat{p}^{-2}))$. Hence, the success rate of this attack is

$$\frac{1 - (1 - 2^{-l} \cdot \hat{p}^2)^{2^{l+1} \cdot \hat{p}^{-2}}}{2} + \frac{(1 - 2^{-2l})^{2^{l+1} \cdot \hat{p}^{-2}}}{2} \approx \frac{1 - e^{-2}}{2} + \frac{e^{-2^{-l+1} \cdot \hat{p}^{-2}}}{2}.$$

Here, the first term is approximately 0.43. Our second distinguishing attack on HMAC using q and a differential distinguisher is described as follows:

1. Collect $2 \cdot q^{-1}$ message pairs (M_i, M_i') with difference α, where all the M_i and M_i' have the same bit-length t.
2. With a chosen message attack scenario, ask for MAC pairs of all the (M_i, M_i'). We denote the corresponding MAC pairs by (C_i, C_i'). We assume that the MAC algorithm is either an instantiated HMAC or a random function (or HMAC with a random function) which maps t bits to l bits.
3. Check if $C_i \oplus C_i' = 0$. If there is at least one MAC pair that satisfies this test, output the MAC algorithm = HMAC, otherwise, output the MAC algorithm = a random function (or HMAC with a random function).

The data complexity of this attack is $2^2 \cdot q^{-1}$ chosen messages; this attack does not require any storage and the time complexity of this attack itself is a fraction of the time required to compute the MAC values for the chosen messages. Similarly, the success rate of this attack is computed as follows:

$$\frac{1 - (1 - q)^{2 \cdot q^{-1}}}{2} + \frac{(1 - 2^{-l})^{2 \cdot q^{-1}}}{2} \approx \frac{1 - e^{-2}}{2} + \frac{e^{-2^{-l+1} \cdot q^{-1}}}{2}.$$

Table 6.3: Distinguishing and Forgery Attacks on HMAC with HAVAL, MD4, MD5, SHA-0 and SHA-1

Hash Function	Type of Attack	#R in h_2	Prob. of Distinguisher	Data Complexity	Success Rate
HAVAL (96 rounds)	**Dist.† (R)**	**96**	$\mathbf{\hat{p} = 2^{-99.1}}$	$\mathbf{2^{228.6}}$	**0.93**
HAVAL (128 rounds)	Dist. (R)	102	$\hat{p} = 2^{-124.4}$	$2^{253.9}$	0.93
MD4 (48 rounds)	**Dist.† (R)**	**48**	$\mathbf{\hat{p} = 2^{-56}}$	$\mathbf{2^{121.5}}$	**0.93**
	Forgery† (D)	**48**	$\mathbf{q = 2^{-56}}$	$\mathbf{2^{58}}$	**0.86**
MD5 (64 rounds)	Dist. (R)	33	$\hat{p} = 2^{-47.6}$	$2^{113.1}$	0.92
SHA-0 (80 rounds)	Dist. (R)	65	$\hat{p} = 2^{-78}$	$2^{159.5}$	0.87
	Dist.† (D)	**82**	$\mathbf{q = 2^{-98}}$	$\mathbf{2^{100}}$	**0.93**
	Dist.† (D)	**80**	$\mathbf{q = 2^{-107}}$	$\mathbf{2^{109}}$	**0.93**
	Forgery (D)	54	$q = 2^{-61}$	2^{63}	0.86
	Forgery (D)	65	$q = 2^{-78}$	2^{80}	0.86
SHA-1 (80 rounds)	Dist. (R)	43	$\hat{p} = 2^{-73.4}$	$2^{154.9}$	0.93
	Forgery (D)	34	$q = 2^{-51}$	2^{53}	0.86

†: the attacks can work on HMAC based on full-round hash functions,
#R: Number of Rounds, Dist.: Distinguishing-H attack,
R: Rectangle Distinguisher, D: Differential Distinguisher,
Data complexity is the number of chosen message queries.

Note that even if the output producing algorithm is either an instantiated HMAC or HMAC with a random function in the above two attacks, the success rates of the attacks are almost the same as the computed ones.

Finally, our forgery attack on HMAC using q and a differential distinguisher is described as follows:

1. Run Step 1 in the second distinguishing attack.
2. Run Step 2 in the second distinguishing attack, but we assume that the MAC algorithm is an instantiated HMAC.
3. Check if $C_i \oplus C'_i = 0$ and ask for the MAC pair of $M_i||P$ and $M'_i||P$, where M_i and M'_i have a same MAC value and P is some non-empty string. If

the obtained MAC pair collides, again ask for the MAC value of $M_i||P||P'$, where P' is some non-empty string. We denote this obtained MAC value by C. Output C as the MAC value of $M_i'||P||P'$. Otherwise, restart this step until we have checked all the MAC pairs (C_i, C_i').

It is easy to see that this forgery attack works with the same data complexity as our second distinguishing attack and a success rate of approximately $1-(1-q)^{2\cdot q^{-1}} \approx 1-e^{-2} = 0.86$.

We can easily apply these three attacks to HMAC based on HAVAL, MD4,[6] MD5, SHA-0 and SHA-1 by using their probabilities $\hat{p}$ and q. Table 6.3 shows the results of distinguishing and forgery attacks on those instantiations of HMAC. In Table 6.3 the memory complexity for the rectangle attack is the same as the data complexity, and forgery attacks also imply distinguishing-R and distinguishing-H attacks.

Note: Our distinguishing and forgery attacks are also applicable to HMAC in which the four components $h_1,\ h_2,\ h_3,\ h_4$ are instantiated with different compression functions (see for example the pseudorandom functions of SSL 3.0. In SSL 3.0, MD5 and SHA-1 are used in the outer and inner hash functions of HMAC, respectively.). For example, if HMAC employs full-round MD-5, full-round MD-4, full-round MD5 and full-round MD5 for $h_1,\ h_2,\ h_3$ and h_4, respectively, it can be forged with a data complexity of 2^{58} chosen messages. This is due to the fact that our distinguishing and forgery attacks depend strongly on the function h_2.

6.6 Applications to NMAC

Due to the similar structure, our differential and rectangle distinguishers of HMAC can also be applied to NMAC. Thus, the distinguishing and forgery attacks described in Sect. 6.5 also work on NMAC and the results of Table 6.3 are applied to NMAC based on HAVAL, MD4, MD5, SHA-0 and SHA-1.

6.7 Some Implications of the Differential and Rectangle Distinguishers of HMAC and NMAC

Our differential and rectangle distinguishers can be useful to construct second-preimage attacks on HMAC and NMAC.

[6] For the HMAC-MD4 attacks, a more delicate method for choosing messages is required. For details, see [26].

It is natural to define a second-preimage resistance for MAC algorithms from that for hash functions.

Second-preimage resistance on MAC algorithms: for any message M, it is computationally infeasible to find another message M' such that $\mathrm{MAC}(K, M') = \mathrm{MAC}(K, M)$, where K is a randomly chosen key.

It follows that for any message M, it should be computationally infeasible for an attacker to find another message M' such that $\mathrm{MAC}(K, M') = \mathrm{MAC}(K, M)$ with a probability larger than 2^{-l}. Since our differential distinguisher uses a probability larger than 2^{-l}, which can find a second preimage with the same differential probability, our differential attacks on HMAC and NMAC imply second-preimage attacks on HMAC.

The second-preimage resistance of MAC algorithms also implies that for any message pair (M_i, M'_j), an attacker cannot find another message pair (M'_i, M_j) such that $\mathrm{MAC}(K, M_i) = \mathrm{MAC}(K, M_j)$ and $\mathrm{MAC}(K, M'_i) = \mathrm{MAC}(K, M'_j)$ with a probability larger than 2^{-2l}. This implies that our rectangle distinguisher is applicable to second-preimage attacks on HMAC and NMAC. For example, consider the distinguishing-H attack on HMAC-HAVAL (3-pass) in Table 6.3. From the probability $\hat{p} = 2^{-99.1}$, we know that for a given message pair (M_i, M'_j), our rectangle distinguisher can find another message pair (M'_i, M_j) such that $f(K, M_i) = f(K, M_j)$ and $f(K, M'_i) = f(K, M'_j)$ with a probability of approximately $(2^{-99.1})^2 \cdot 2^{-256} = 2^{-452.2}$ which is much larger than $(2^{-256})^2 = 2^{-512}$, where $f =$ HMAC-HAVAL (3-pass) (refer to Fig. 2., Tables 6.1 and 6.2).

The second-preimage resistance on MAC algorithms is a weakened security notion of forgery. Indeed, a second-preimage attack implies a forgery. However, the converse dose not hold since second-preimage attacks are first given a target message. This security notion is also very important if meaningful messages are considered.

Related Work: Recently, there has been an article [109] to improve our results on HMAC; [109] shows that the forgery and distinguishing-H attacks can be applied to HMAC based on reduced 37-round and 53-round SHA-1 with data/time complexities of 2^{66} and $2^{98.5}$, respectively. More recently, there has been another article [26] that independently analyzes HMAC; [26] shows that the forgery and partial key recovery attacks can be applied to HMAC/NMAC based on the full 80-round SHA-0 with a data/time complexity of 2^{84} and to HMAC/NMAC based on reduced 34-round SHA-1 with a data/time complexity of 2^{34}. [26] also shows that with a data/time complexity of 2^{47} the forgery and partial key recovery attacks can be applied to NMAC based on the full 64-round MD5 that uses related keys.

6.8 Conclusion

We have presented differential and rectangle distinguishers on HMAC, which are derived from its structural property. They allow to present distinguishing and forgery attacks on HMAC that can be mounted when HMAC employs hash functions with slow difference propagations. With these distinguishing and forgery attacks we have shown that HMAC with the full versions of 3-pass HAVAL and SHA-0 can be distinguished from HMAC with a random function, and HMAC with the full version of MD4 can be forged. These distinguishing and forgery attacks have also been applied to HMAC based on reduced versions of MD5 and SHA-1. We have also shown that our distinguishing and forgery attacks can be mounted on NMAC (which is a generalized version of HMAC) with the same complexity. Furthermore, we have presented second-preimage attacks on HMAC and NMAC by using our differential and rectangle distinguishers. None of these attacks contradict the security proof of HMAC, but they improve our understanding of the security of HMAC based on existing cryptographic hash functions.

Our differential distinguisher on HMAC works only if the underlying hash function has a differential with a zero output difference with probability larger than 2^{-l}, where l is the bit-length of MAC values. Our rectangle distinguisher on HMAC works only if the underlying hash function has differentials such that the sum of the squares of their probabilities is larger than 2^{-l}. Unlike the previous attacks on hash functions, our analysis on the hash function embedded in HMAC should be done under an unknown fixed initial value which is determined by a secret key. This fact makes difficult to use the recently proposed message modification technique (Wang et al.'s attacks) and neutral-bit technique (Biham et al.'s attacks) in analyzing HMAC based on specific cryptographic hash functions. However, it is interesting to investigate if their methods can be applied to HMAC with some new other techniques when HMAC is instantiated with a specific cryptographic hash function. We expect that the method developed in this thesis would be useful for the further analysis of HMAC.

Chapter 7

Conclusions and Further Research

7.1 Conclusions

In this thesis, we have introduced several new combined differential, linear and related-key attacks and showed their usefulness in analyzing existing symmetric-key algorithms.

Each of our combined attacks treats a cipher as a cascade of two sub-ciphers, applies a known differential-style, linear-style or related-key distinguisher to each sub-cipher and then combines them to obtain a new distinguisher on the cipher. Combining differential-style, linear-style and related-key distinguishers we have devised the *differential-nonlinear attack*, the *square-(non)linear attack*, the *related-key differential-(non)linear attack* and the *related-key rectangle and boomerang attacks*. The cryptanalytic conditions on which our combined attacks work have been analyzed in this thesis.

Applying our combined attacks to existing symmetric-key algorithms we have obtained the following results:

- the first known attack on the full SHACAL-1 that uses related keys;
- the best known attack on reduced SHACAL-2 that uses a single key;
- the best known attacks on reduced SHACAL-2 and reduced AES-192 that uses related keys;
- the best known attacks on the full MD4, MD5 and HAVAL in encryption mode;

- the first known attacks on HMAC with the full 3-pass HAVAL and the full MD4.

Our cryptanalytic results presented in this thesis show that our combined attacks are useful tools for block ciphers and MAC algorithms.

The results in this thesis have been published in the papers [35, 49, 63, 64, 67, 68, 89, 119]. The remaining papers [25, 47, 48, 52, 61, 62, 65, 66, 69, 70, 71, 83, 84, 85, 86, 87, 90, 91, 121] published during this doctoral research are not included in this thesis.

7.2 Further Research

As stated earlier, differential cryptanalysis (DC), linear cryptanalysis (LC) and related-key cryptanalysis (RKC) are the most widely used cryptanalytic techniques. Since the introduction of DC and LC, various variants of DC and LC have been proposed: differential-style cryptanalysis – truncated differential cryptanalysis (TDC), higher order differential cryptanalysis (HODC), square cryptanalysis (SC), impossible differential cryptanalysis (IDC), boomerang cryptanalysis (BC) and rectangle cryptanalysis (RC), and linear-style cryptanalysis – multiple linear cryptanalysis (MLC), nonlinear cryptanalysis (NLC), bilinear cryptanalysis (BLC).

Besides our combined attacks differential-(non)linear cryptanalysis (D-(N)LC), square-(non)linear cryptanalysis (S-(N)LC), related-key differential-(non)linear cryptanalysis (RK-D-(N)LC), related-key rectangle cryptanalysis (RK-RC) and related-key boomerang cryptanalysis (RK-BC), recently, several new combined attacks have been proposed, which are called differential-bilinear cryptanalysis (D-BLC) [14], higher order differential-linear cryptanalysis (HOD-LC) [14], differential-linear-boomerang cryptanalysis (D-L-BC) [14] and differential-bilinear-boomerang cryptanalysis (D-BL-BC) [14].

In the future work, it would be interesting to investigate the following research questions:

- Can the attacks on existing symmetric-key algorithms presented in this thesis be further improved? Note that our related-key rectangle attacks on SHACAL-1, SHACAL-2 and AES are all improved results over previously known attacks.

- Can the existing combined attacks be applied to other symmetric-key algorithms? We have demonstrated in this thesis the applicability to the combined attacks on block ciphers (including hash functions in encryption mode) and hash function based MAC algorithms. Are they applicable to stream ciphers, hash functions themselves and block cipher based MAC algorithms?

- Can differential-style, linear-style and related-key attacks be further combined to make new attacks, for instance, differential-multiple linear cryptanalysis (D-MLC), higher order differential-multiple linear cryptanalysis (HOD-MLC), differential-nonlinear boomerang cryptanalysis (D-NL-BC), differential-linear rectangle cryptanalysis (D-L-RC), related-key differential-multiple linear cryptanalysis (RK-D-MLC), related-key differential-linear boomerang cryptanalysis (RK-D-L-BC), related-key differential-bilinear-boomerang cryptanalysis (RK-D-BL-BC), related-key differential-nonlinear boomerang cryptanalysis (RK-D-NL-BC) and related-key differential-linear rectangle cryptanalysis (RK-D-L-RC)? See Fig. 7.1 for a schematic description of combined attacks.

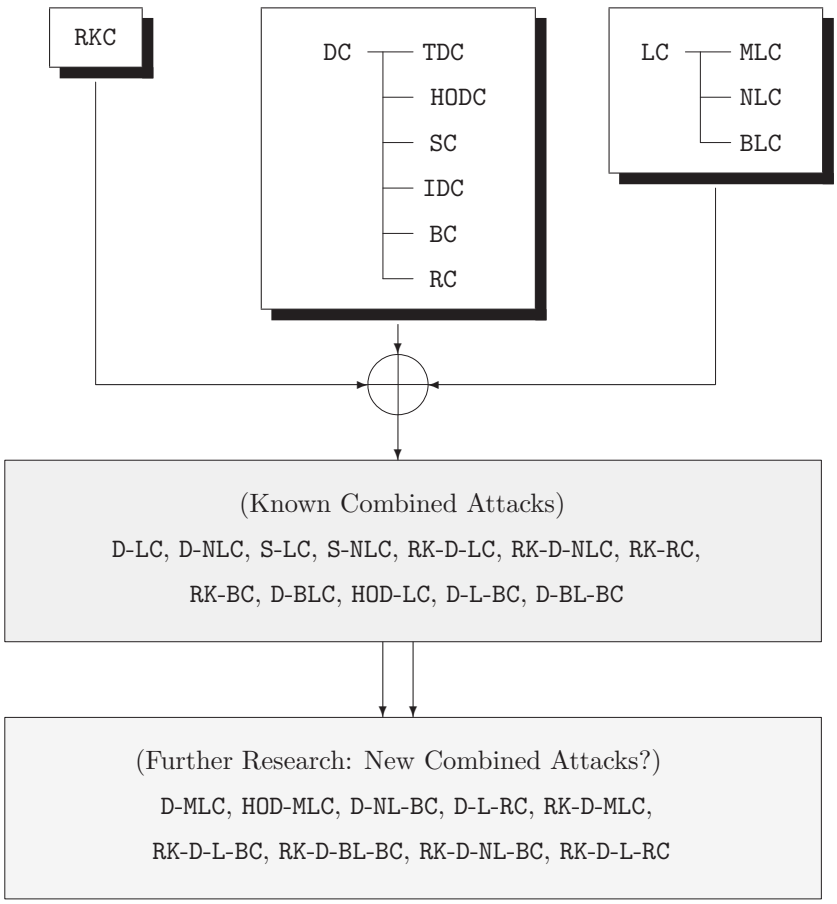


Figure 7.1: Further Research on Combined Attacks

Appendix A

Attacks on Reduced AES-192 and AES-256

We here present the following three types of related-key rectangle attacks on reduced AES-192 and AES-256:

- **Related-Key Rectangle Attack on 8-Round AES-192 with 2 related keys (TYPE 1):** This attack recovers bytes 1, 2, 6, 7, 8, 11, 12, 13 of the whitening key pair (K_w, K_w^*) and bytes 3, 6, 9, 12 of the subkey pair (K_7, K_7^*) with a data complexity of about 2^{94} related-key chosen plaintexts, a time complexity of about 2^{120} encryptions and a success rate of 0.9. See Appendix A.1 for a schematic description of this attack (note than Fig. A.1 is the same as Fig. 4.1 in Sect. 4.7.1).
- **Related-Key Rectangle Attack on 8-Round AES-192 with 4 related keys (TYPE 3):** This attack recovers bytes 3, 5, 6, 9, 12 of the subkey quartet $(K_7, K_7^*, K_7', K_7'^*)$ with a data complexity of about $2^{86.5}$ related-key chosen plaintexts, a time complexity of about $2^{86.5}$ encryptions and a success rate of 0.76. See Appendix A.2 for a schematic description of this attack.
- **Related-Key Rectangle Attack on 9-Round AES-256 with 4 related keys (TYPE 3):** This attack recovers bytes 1, 2, 6, 7, 8, 11, 12, 13 of the whitening key quartet $(K_w, K_w^*, K_w', K_w'^*)$ and bytes 0, 4, 8, 12 of the subkey quartet $(K_8, K_8^*, K_8', K_8'^*)$ with a data complexity of about 2^{99} related-key chosen plaintexts, a time complexity of about 2^{120} encryptions and a success rate of 0.9. See Appendix A.3 for a schematic description of this attack.

A.1 Related-Key Rectangle Attack on 8-Round AES-192 (TYPE 1)

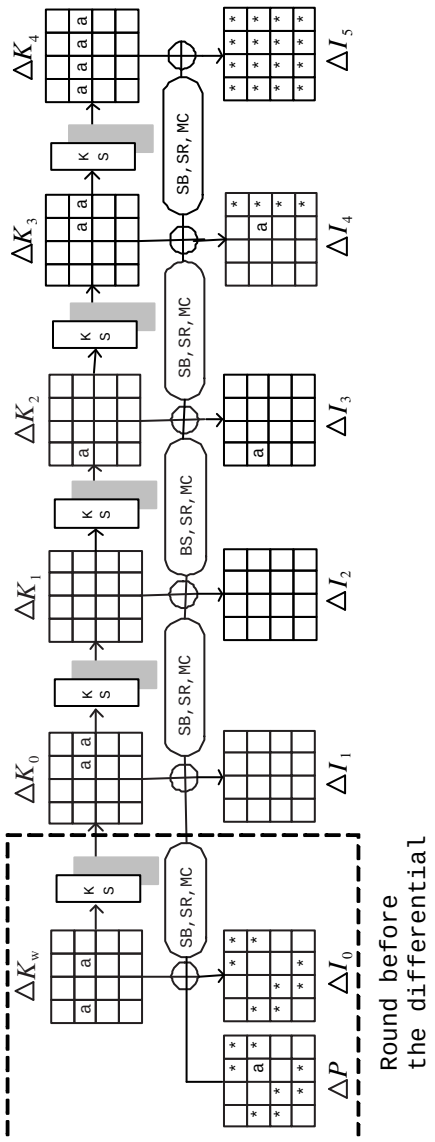


Figure A.1: Related-Key Truncated Differential for Rounds 1-4 of AES-192

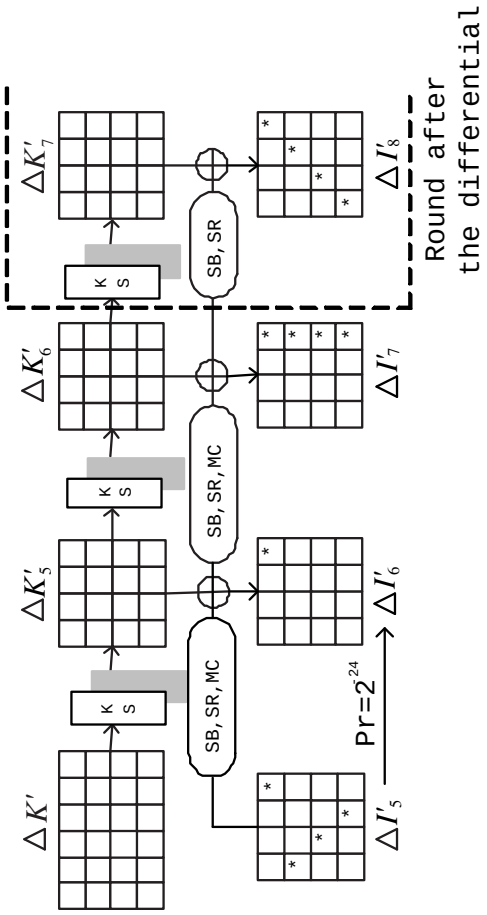


Figure A.2: Truncated Differential for Rounds 5-6 of AES-192

A.2 Related-Key Rectangle Attack on 8-Round AES-192 (TYPE 3)

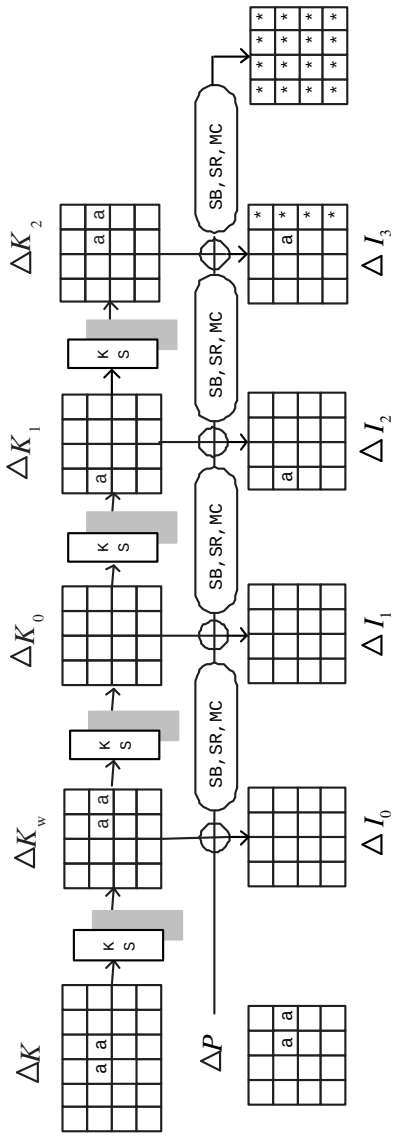


Figure A.3: Related-Key Truncated Differential for Rounds 0-3 of AES-192

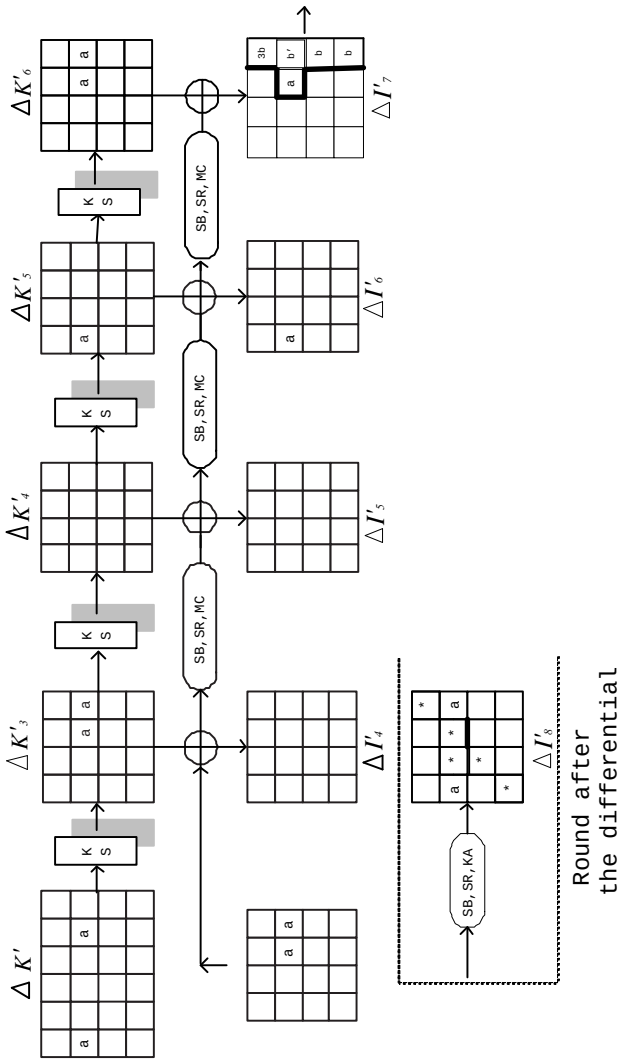


Figure A.4: Related-Key Truncated Differential for Rounds 4-6 of AES-192

A.3 Related-Key Rectangle Attack on 9-Round AES-256 (TYPE 3)

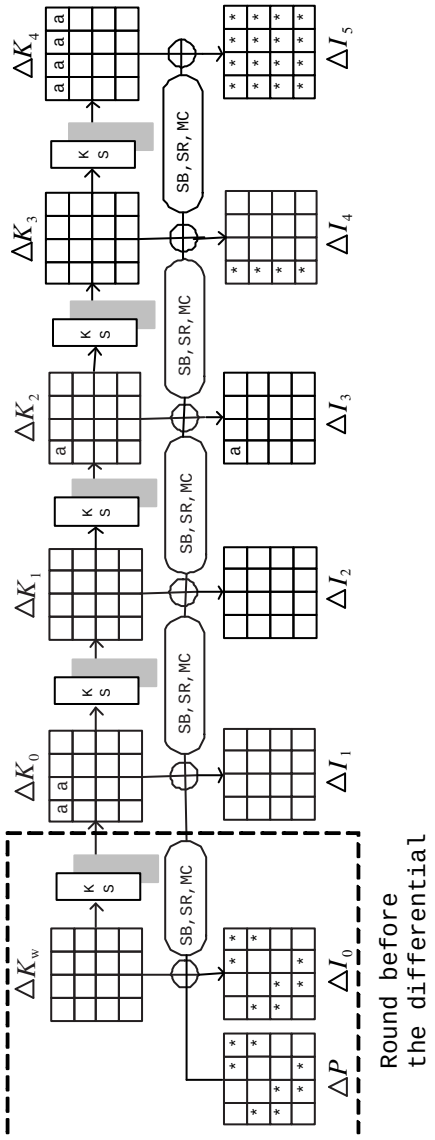


Figure A.5: Related-Key Truncated Differential for Rounds 1-4 of AES-256

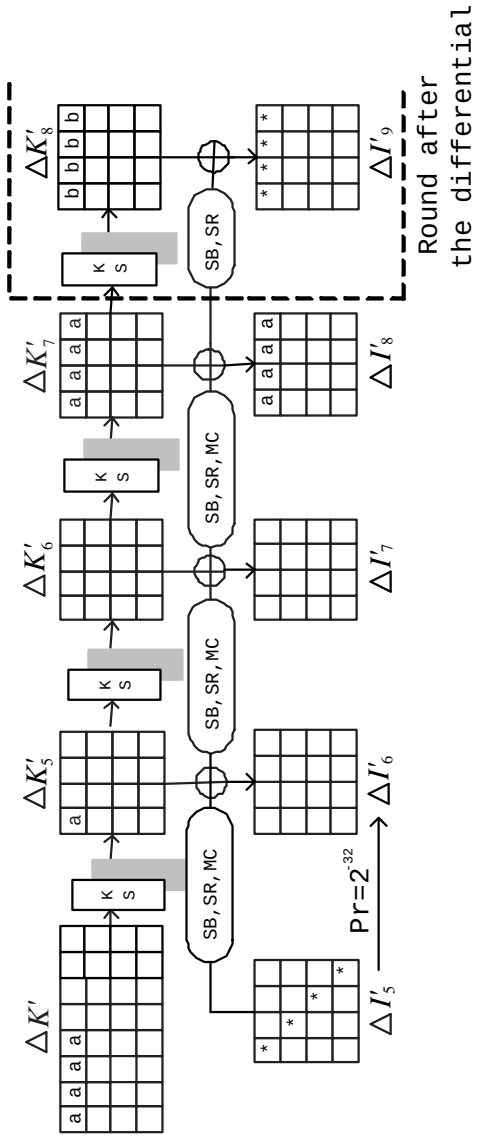


Figure A.6: Related-Key Truncated Differential for Rounds 5-7 of AES-256

Appendix B

Distinguishers of MD4, MD5 and HAVAL in Encryption Mode

In our distinguishers we use zero key differences with the following exceptions:

- in Table B.1, $\Delta K^3 = e_{31}$ in the first related-key differential and $\Delta K^7 = e_{31}$ in the second related-key differential,
- in Table B.2, $\Delta K^8 = e_{31}$ in the first and second related-key differentials,
- in Table B.3, $\Delta K^2 = e_{31}$ in the first related-key differential and $\Delta K^{11} = e_{31}$ in the second related-key differential,
- in Tables B.4 and B.5 , $\Delta K^9 = e_{31}$ in the first and second related-key differentials,
- in Table B.7, $\Delta K^2 = e_{31}$ in the first related-key differential and $\Delta K^{17} = e_{31}$ in the second related-key differential,
- in Tables B.8 and B.9, $\Delta K^2 = e_{31}$ in the first related-key differential and $\Delta K^4 = e_{31}$ in the second related-key differential.

B.1 Distinguishers of MD4 and their Probabilities

Table B.1: Boomerang Distinguishers of MD4 (Four Related Keys)

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔK^i	Prob.
0	0	e_{31}	0	0	0	1
1	0	0	e_{31}	0	0	2^{-1}
2	0	0	0	e_{31}	0	2^{-1}
3	e_{31}	0	0	0	e_{31}	1
4	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
27	0	0	0	0	0	1
28	0	0	0	0	e_{31}	1
	0	e_2	0	0		$p^* = 2^{-2}$
29	e_{31}	0	0	0	e_{31}	1
30	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
45	0	0	0	0	0	1
46	0	0	0	0	e_{31}	1
47	0	e_{10}	0	0	0	2^{-1}
48	0	e_{25}	e_{10}	0	0	$q^* = 2^{-1}$
BOO-4	$(0 \rightarrow 28)$, $(47 \rightarrow 29)^2$, $(28 \rightarrow 3)$					Pr[BOO-4] $\approx 2^{-4}$
BOO^W*-4*	Fixed $K^{0,1,2,15}$, $(3 \rightarrow 28)$, $(46 \rightarrow 29)^2$, $(28 \rightarrow 3)$					Pr[BOO-4] $= 1$

B.2 Distinguishers of MD5 and their Probabilities

Table B.2: Boomerang Distinguishers of MD5 (Two Related Keys)

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔK^i	Prob.
0	e_{17}	0	e_2	$e_{7,12}$	0	2^{-2}
1	$e_{7,12}$	e_{24}	0	e_2	0	2^{-6}
2	e_2	e_{19}	e_{24}	0	0	2^{-4}
3	0	0	e_{19}	e_{24}	0	2^{-2}
4	e_{24}	0	0	e_{19}	0	2^{-2}
5	e_{19}	e_{31}	0	0	0	2^{-2}
6	0	0	e_{31}	0	0	2^{-1}
7	0	0	0	e_{31}	0	2^{-1}
8	e_{31}	0	0	0	e_{31}	1
9	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
27	0	0	0	0	e_{31}	2^{-1}
28	0	e_{19}	0	0	0	2^{-2}
29	0	e_{19}	e_{19}	0	0	2^{-3}
30	0	$e_{19,28}$	e_{19}	e_{19}	0	$p^* = 2^{-26}$
30	$e_{17,31}$	e_{31}	e_{31}	0	0	2^{-1}
31	0	0	e_{31}	e_{31}	0	2^{-1}
32	e_{31}	0	0	e_{31}	0	1
33	e_{31}	0	0	0	e_{31}	1
34	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
56	0	0	0	0	e_{31}	2^{-1}
57	0	e_5	0	0	0	2^{-2}
58	0	e_5	e_5	0	0	2^{-2}
59	0	e_5	e_5	e_5	0	2^{-2}
60	e_5	e_5	e_5	e_5	0	2^{-3}
61	e_5	e_5	e_5	e_5	0	2^{-3}
62	e_5	e_5	e_5	e_5	0	2^{-3}
63	e_5	e_5	e_5	e_5	0	2^{-3}
64	e_5	e_5	e_5	e_5		$q^* = 2^{-21}$
BOO-2	$(0 \rightarrow 29)$, $(63 \rightarrow 30)^2$, $(29 \rightarrow 1)$					Pr[BOO-2] $\approx 2^{-78.6}$
BOO^W-2	Fixed $K^{0,1,2,3,4,5,6,7,8,9,11,13,15}$ $(8 \rightarrow 29)$, $(55 \rightarrow 30)^2$, $(29 \rightarrow 8)$					Pr[BOO-2] $\approx 2^{-1.6}$

Table B.3: Boomerang Distinguishers of MD5 (Four Related Keys)

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔK^i	Prob.
0	0	0	e_{31}	0	0	1
1	0	0	0	e_{31}	0	2^{-1}
2	e_{31}	0	0	0	e_{31}	1
3	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
28	0	0	0	0	0	1
29	0	0	0	0	e_{31}	2^{-1}
30	0	e_8	0	0	0	2^{-2}
31	0	e_8	e_8	0		$p^* = 2^{-4}$
31	$e_{11,31}$	e_{31}	e_{31}	0	0	2^{-1}
32	0	0	e_{31}	e_{31}	0	1
33	e_{31}	0	0	e_{31}	0	1
34	e_{31}	0	0	0	e_{31}	1
35	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
60	0	0	0	0	0	1
61	0	0	0	0	e_{31}	2^{-1}
62	0	e_9	0	0	0	2^{-2}
63	0	e_9	e_9	0	0	2^{-2}
64	0	e_9	e_9	e_9		$q^* = 2^{-6}$
BOO-4	$(0 \rightarrow 30)$, $(63 \rightarrow 31)^2$, $(30 \rightarrow 2)$					Pr[BOO-4] $\approx 2^{-11.6}$
BOO^W-4	Fixed $K^{0,1,2,9,11}$, $(2 \rightarrow 30)$, $(60 \rightarrow 31)^2$, $(30 \rightarrow 2)$					Pr[BOO-4] $\approx 2^{-0.6}$

B.3 Distinguishers of HAVAL and their Probabilities

Table B.4: Boomerang Distinguishers of 4-Pass HAVAL (Two Related Keys: the First Differential)

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	ΔF^i	ΔG^i	ΔH^i	ΔK^i	Prob.
0	0	e_{21}	0	0	0	0	0	0	0	1
1	e_{21}	0	0	0	0	0	0	0	0	2^{-1}
2	0	0	0	0	0	0	0	e_{10}	0	2^{-1}
3	0	0	0	0	0	0	e_{10}	0	0	2^{-1}
4	0	0	0	0	0	e_{10}	0	0	0	2^{-1}
5	0	0	0	0	e_{10}	0	0	0	0	2^{-1}
6	0	0	0	e_{10}	0	0	0	0	0	2^{-1}
7	0	0	e_{10}	0	0	0	0	0	0	2^{-1}
8	0	e_{10}	0	0	0	0	0	0	0	2^{-1}
9	e_{10}	0	0	0	0	0	0	0	e_{31}	1
10	0	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
51	0	0	0	0	0	0	0	0	e_{31}	1
52	0	0	0	0	0	0	0	e_{31}	0	2^{-1}
53	0	0	0	0	0	0	e_{31}	0	0	2^{-1}
54	0	0	0	0	0	e_{31}	0	0	0	2^{-1}
55	0	0	0	0	e_{31}	0	0	0	0	2^{-1}
56	0	0	0	e_{31}	0	0	0	0	0	2^{-1}
57	0	0	e_{31}	0	0	0	0	0	0	2^{-1}
58	0	e_{31}	0	0	0	0	0	0	0	2^{-1}
59	e_{31}	0	0	0	0	0	0	0	0	2^{-1}
60	0	0	0	0	0	0	0	e_{20}	0	2^{-1}
61	0	0	0	0	0	0	e_{20}	0	0	2^{-1}
62	0	0	0	0	0	e_{20}	0	0	0	2^{-1}
63	0	0	0	0	e_{20}	0	0	0	0	2^{-1}
64	0	0	0	e_{20}	0	0	0	0		$p^* = 2^{-20}$

Table B.5: Boomerang Distinguishers of 4-Pass HAVAL (Two Related Keys: the Second Differential)

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	ΔF^i	ΔG^i	ΔH^i	ΔK^i	Prob.
64	0	e_{10}	0	0	0	0	0	0	0	2^{-1}
65	e_{10}	0	0	0	0	0	0	0	e_{31}	1
66	0	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
119	0	0	0	0	0	0	0	0	e_{31}	1
120	0	0	0	0	0	0	0	e_{31}	0	2^{-1}
121	0	0	0	0	0	0	e_{31}	0	0	2^{-1}
122	0	0	0	0	0	e_{31}	0	0	0	2^{-1}
123	0	0	0	0	e_{31}	0	0	0	0	2^{-1}
124	0	0	0	e_{31}	0	0	0	0	0	2^{-1}
125	0	0	e_{31}	0	0	0	0	0	0	2^{-1}
126	0	e_{31}	0	0	0	0	0	0	0	2^{-1}
127	e_{31}	0	0	0	0	0	0	0	0	2^{-1}
128	0	0	0	0	0	0	0	e_{20}	0	$q^* = 2^{-9}$
BOO-2	$(0 \rightarrow 63)$, $(127 \rightarrow 64)^2$, $(63 \rightarrow 7)$									Pr[BOO-2] $\approx 2^{-35.9}$
BOO^W-2	Fixed $K^{0,1,2,3,4,5,6,7,8,15,16,22,23,25}$, $(9 \rightarrow 63)$, $(119 \rightarrow 64)^2$, $(63 \rightarrow 9)$									Pr[BOO-2] $\approx 2^{-10.3}$

Table B.6: Boomerang Distinguishers of 5-Pass HAVAL (Two Related Keys) – Extension of the Distinguishers for 4-Pass HAVAL

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	ΔF^i	ΔG^i	ΔH^i	ΔK^i	Prob.
128	0	0	0	0	0	0	0	e_{20}	0	2^{-1}
129	0	0	0	0	0	0	e_{20}	0	0	2^{-1}
130	0	0	0	0	0	e_{20}	0	0	0	2^{-1}
131	0	0	0	0	e_{20}	0	0	0	0	2^{-1}
132	0	0	0	e_{20}	0	0	0	0	0	2^{-1}
133	0	0	e_{20}	0	0	0	0	0	0	2^{-1}
134	0	e_{20}	0	0	0	0	0	0	0	2^{-1}
135	e_{20}	0	0	0	0	0	0	0	0	2^{-1}
136	0	0	0	0	0	0	0	e_9	0	2^{-1}
137	0	0	0	0	0	0	e_9	0	0	2^{-1}
138	0	0	0	0	0	e_9	0	0	0	2^{-1}
139	0	0	0	0	e_9	0	0	0	0	2^{-1}
140	0	0	0	e_9	0	0	0	0	0	2^{-1}
141	0	0	e_9	0	0	0	0	0	0	2^{-1}
142	0	e_9	0	0	0	0	0	0	0	2^{-1}
143	e_9	0	0	0	0	0	0	0	0	2^{-1}
144	0	0	0	0	0	0	0	e_{30}	0	2^{-1}
145	0	0	0	0	0	0	e_{30}	0	e_{31}	2^{-1}
146	0	0	0	0	0	e_{30}	0	e_{31}	0	2^{-2}
147	0	0	0	0	e_{30}	0	e_{31}	0	0	2^{-2}
148	0	0	0	e_{30}	0	e_{31}	0	0	0	2^{-2}
149	0	0	e_{30}	0	e_{31}	0	0	0	0	2^{-2}
150	0	e_{30}	0	e_{31}	0	0	0	0	0	2^{-2}
151	e_{30}	0	e_{31}	0	0	0	0	0	0	2^{-2}
152	0	e_{31}	0	0	0	0	0	e_{19}	0	2^{-2}
153	e_{31}	0	0	0	0	0	e_{19}	0	0	2^{-2}
154	0	0	0	0	0	e_{19}	0	e_{20}	0	2^{-2}
155	0	0	0	0	e_{19}	0	e_{20}	0	0	2^{-2}
156	0	0	0	e_{19}	0	e_{20}	0	0	0	2^{-2}
157	0	0	e_{19}	0	e_{20}	0	0	0	0	2^{-2}
158	0	e_{19}	0	e_{20}	0	0	0	0	0	2^{-2}
159	e_{19}	0	e_{20}	0	0	0	0	0	0	2^{-2}
160	0	e_{20}	0	0	0	0	0	e_8		$q^* = 2^{-54}$
BOO-2	$(0 \rightarrow 63)$, $(159 \rightarrow 64)^2$, $(63 \rightarrow 4)$									Pr[BOO-2] $\approx 2^{-125.9}$

Table B.7: Boomerang Distinguishers of 4-Pass HAVAL (Four Related Keys)

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	ΔF^i	ΔG^i	ΔH^i	ΔK^i	Prob.
0	0	0	e_{10}	0	0	0	0	0	0	1
1	0	e_{10}	0	0	0	0	0	0	0	2^{-1}
2	e_{10}	0	0	0	0	0	0	0	e_{31}	1
3	0	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
59	0	0	0	0	0	0	0	0	0	1
60	0	0	0	0	0	0	0	0	e_{31}	1
61	0	0	0	0	0	0	0	e_{31}	0	2^{-1}
62	0	0	0	0	0	0	e_{31}	0	0	2^{-1}
63	0	0	0	0	0	e_{31}	0	0	0	2^{-1}
64	0	0	0	0	e_{31}	0	0	0		$p^* = 2^{-4}$
64	0	0	0	0	0	e_{10}	0	0	0	2^{-1}
65	0	0	0	0	e_{10}	0	0	0	0	2^{-1}
66	0	0	0	e_{10}	0	0	0	0	0	2^{-1}
67	0	0	e_{10}	0	0	0	0	0	0	2^{-1}
68	0	e_{10}	0	0	0	0	0	0	0	2^{-1}
69	e_{10}	0	0	0	0	0	0	0	e_{31}	1
70	0	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
123	0	0	0	0	0	0	0	0	0	1
124	0	0	0	0	0	0	0	0	e_{31}	1
125	0	0	0	0	0	0	0	e_{31}	0	2^{-1}
126	0	0	0	0	0	0	e_{31}	0	0	2^{-1}
127	0	0	0	0	0	e_{31}	0	0	0	2^{-1}
128	0	0	0	0	e_{31}	0	0	0	0	$q^* = 2^{-8}$
BOO-4	$(0 \to 63)$, $(127 \to 64)^2$, $(63 \to 2)$									Pr[BOO-4] $\approx 2^{-9.6}$
BOO^W-4	Fixed $K^{0,1,15,24}$, $(2 \to 63)$, $(124 \to 64)^2$, $(63 \to 2)$									Pr[BOO-4] $\approx 2^{-3}$

Table B.8: Boomerang Distinguishers of 5-Pass HAVAL (Four Related Keys: the First Differential)

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	ΔF^i	ΔG^i	ΔH^i	ΔK^i	Prob.
0	0	0	e_{10}	0	0	0	0	0	0	1
1	0	e_{10}	0	0	0	0	0	0	0	2^{-1}
2	e_{10}	0	0	0	0	0	0	0	e_{31}	1
3	0	0	0	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
59	0	0	0	0	0	0	0	0	0	1
60	0	0	0	0	0	0	0	0	e_{31}	1
61	0	0	0	0	0	0	0	e_{31}	0	2^{-1}
62	0	0	0	0	0	0	e_{31}	0	0	2^{-1}
63	0	0	0	0	0	e_{31}	0	0	0	2^{-1}
64	0	0	0	0	e_{31}	0	0	0		2^{-1}
65	0	0	0	e_{31}	0	0	0	0	0	2^{-1}
66	0	0	e_{31}	0	0	0	0	0	0	2^{-1}
67	0	e_{31}	0	0	0	0	0	0	0	2^{-1}
68	e_{31}	0	0	0	0	0	0	0	0	2^{-1}
69	0	0	0	0	0	0	0	e_{20}	0	2^{-1}
70	0	0	0	0	0	0	e_{20}	0	0	2^{-1}
71	0	0	0	0	0	e_{20}	0	0	0	2^{-1}
72	0	0	0	0	e_{20}	0	0	0	0	2^{-1}
73	0	0	0	e_{20}	0	0	0	0	0	2^{-1}
74	0	0	e_{20}	0	0	0	0	0	0	2^{-1}
75	0	e_{20}	0	0	0	0	0	0	0	2^{-1}
76	e_{20}	0	0	0	0	0	0	0	0	2^{-1}
77	0	0	0	0	0	0	0	e_9	0	2^{-1}
78	0	0	0	0	0	0	e_9	0	0	2^{-1}
79	0	0	0	0	0	e_9	0	0	0	2^{-1}
80	0	0	0	0	e_9	0	0	0	0	2^{-1}
81	0	0	0	e_9	0	0	0	0	0	2^{-1}
82	0	0	e_9	0	0	0	0	0	0	2^{-1}
83	0	e_9	0	0	0	0	0	0	0	2^{-1}
84	e_9	0	0	0	0	0	0	0	0	2^{-1}
85	0	0	0	0	0	0	0	e_{30}	0	2^{-1}
86	0	0	0	0	0	0	e_{30}	0	0	2^{-1}
87	0	0	0	0	0	e_{30}	0	0	0	2^{-1}
88	0	0	0	0	e_{30}	0	0	0		2^{-1}
89	0	0	0	e_{30}	0	0	0	0	0	2^{-1}
90	0	0	e_{30}	0	0	0	0	0	0	2^{-1}
91	0	e_{30}	0	0	0	0	0	0	0	2^{-1}
92	e_{30}	0	0	0	0	0	0	0	0	2^{-1}
93	0	0	0	0	0	0	0	e_{19}	0	2^{-1}
94	0	0	0	0	0	0	e_{19}	0	0	2^{-1}
95	0	0	0	0	0	e_{19}	0	0	0	$p^* = 2^{-33}$

Table B.9: Boomerang Distinguishers of 5-Pass HAVAL (Four Related Keys: the Second Differential)

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	ΔF^i	ΔG^i	ΔH^i	ΔK^i	Prob.
95	0	0	e_{10}	0	0	0	0	0	0	2^{-1}
96	0	e_{10}	0	0	0	0	0	0	0	2^{-1}
97	e_{10}	0	0	0	0	0	0	0	e_{31}	1
98	0	0	0	0	0	0	0	0	0	1
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
155	0	0	0	0	0	0	0	0	0	1
156	0	0	0	0	0	0	0	0	e_{31}	1
157	0	0	0	0	0	0	0	e_{31}	0	2^{-1}
158	0	0	0	0	0	0	e_{31}	0	0	2^{-1}
159	0	0	0	0	0	e_{31}	0	0	0	2^{-1}
160	0	0	0	0	e_{31}	0	0	0	0	$q^* = 2^{-5}$
BOO-4	$(0 \rightarrow 94)$, $(159 \rightarrow 95)^2$, $(94 \rightarrow 2)$									Pr[BOO-4] $\approx 2^{-61}$

B.4 An Example of Experimental Results: A Boomerang Quartet for MD5

- Four Related Keys:

K = 0x08b870e5 0x33b24180 0x7cec25d3 0x3c8b1b4d 0x50b44c2a
0x14b9206c 0x4aa22bc5 0x51f907af 0x1e096337 0x2ee81e54
0x2c0734bb 0x74231c91 0x55c31f6a 0x7cad2870 0x43f418b1
0x59917add

K^* = 0x08b870e5 0x33b24180 0xfcec25d3 0x3c8b1b4d 0x50b44c2a
0x14b9206c 0x4aa22bc5 0x51f907af 0x1e096337 0x2ee81e54
0x2c0734bb 0x74231c91 0x55c31f6a 0x7cad2870 0x43f418b1
0x59917add

K' = 0x08b870e5 0x33b24180 0x7cec25d3 0x3c8b1b4d 0x50b44c2a
0x14b9206c 0x4aa22bc5 0x51f907af 0x1e096337 0x2ee81e54
0x2c0734bb 0xf4231c91 0x55c31f6a 0x7cad2870 0x43f418b1
0x59917add

K'^* = 0x08b870e5 0x33b24180 0xfcec25d3 0x3c8b1b4d 0x50b44c2a
0x14b9206c 0x4aa22bc5 0x51f907af 0x1e096337 0x2ee81e54
0x2c0734bb 0xf4231c91 0x55c31f6a 0x7cad2870 0x43f418b1
0x59917add

- Chosen Plaintext Pair:

P = 0x6a951691 0x44c50ce4 0x4f533b21 0x66c053b8
P^* = 0x6a951691 0x44c50ce4 0xcf533b21 0x66c053b8

- Corresponding Ciphertext Pair
$(C = MD5_K(P), C^* = MD5_{K^*}(P^*))$:

C = 0xef54db89 0xdc642d4e 0x5b10bd8f 0xf8ab0cd7
C^* = 0xf989429c 0x8583799e 0xe3e1603f 0x81f0c43c

- Adaptively Chosen Ciphertext Pair
$(C' = C \oplus (0, e_9, e_9, e_9), C'^* = C^* \oplus (0, e_9, e_9, e_9))$:

C' = 0xef54db89 0xdc642f4e 0x5b10bf8f 0xf8ab0ed7
C'^* = 0xf989429c 0x85837b9e 0xe3e1623f 0x81f0c63c

- Corresponding Plaintext Pair
$(P = MD5^{-1}_{K'}(C'), P'^* = MD5^{-1}_{K'^*}(C'^*))$:

P' = 0x393c8bdc 0x2c6a7690 0x37d728f1 0xd778127f
P'^* = 0x393c8bdc 0x2c6a7690 0xb7d728f1 0xd778127f

Appendix C

Differentials of MD5, SHA-0 and SHA-1 in HMAC

In this appendix, we present differentials of MD5, SHA-0 and SHA-1 that can be used in HMAC.

- **For MD5:** we insert differences e_{24}, e_{19} and e_{31} into message words 8, 9 and 12 to devise a differential for rounds 0 to 32 with probability 2^{-56}. This differential is used as a rectangle distinguisher with probability $\hat{p} = 2^{-47.6}$ in HMAC with 33-round MD5.
- **For SHA-0:** we first present a 65-round differential with probability 2^{-78} and then a 82-round differential with probability 2^{-98} (these differentials are the same as those in [9, 10], but they have different probabilities). The first differential is used as a rectangle distinguisher with probability $\hat{p} = 2^{-78}$ in HMAC with 65-round SHA-0 and the second differential is used as a differential distinguisher with probability $q = 2^{-98}$ in HMAC with 82-round SHA-0.
- **For SHA-1:** we present a 43-round differential with probability 2^{-75} (this differential is the same as that in [10], but it has a different probability). This differential is used as a rectangle distinguisher with probability $\hat{p} = 2^{-73.4}$ in HMAC with 43-round SHA-1.

C.1 Differential of MD5 and its Probability

Table C.1: Differential for Rounds 0-32 of MD5

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	Δm^i	Prob.
0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
7	0	0	0	0	0	1
8	0	0	0	0	e_{24}	2^{-1}
9	0	e_{31}	0	0	e_{19}	2^{-2}
10	0	0	e_{31}	0	0	2^{-1}
11	0	0	0	e_{31}	0	2^{-1}
12	e_{31}	0	0	0	e_{31}	1
13	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮
23	0	0	0	0	0	1
24	0	0	0	0	e_{19}	2^{-2}
25	0	e_{24}	0	0	0	2^{-2}
26	0	e_{24}	e_{24}	0	0	2^{-3}
27	0	$e_{6,24}$	e_{24}	e_{24}	e_{24}	2^{-5}
28	e_{24}	$e_{6,24}$	$e_{6,24}$	e_{24}	0	2^{-6}
29	e_{24}	$e_{6,11,24}$	$e_{6,24}$	$e_{6,24}$	0	2^{-7}
30	$e_{6,24}$	$e_{6,11,24}$	$e_{6,11,24}$	$e_{6,24}$	0	2^{-9}
31	$e_{6,24}$	$e_{6,11,24}$	$e_{6,11,24}$	$e_{6,11,24}$	e_{31}	2^{-9}
32	$e_{6,11,24}$	$e_{6,11,19,24}$	$e_{6,11,24}$	$e_{6,11,24}$	0	2^{-8}
33	$e_{6,11,24}$	$e_{6,11,23}$	$e_{6,11,19,24}$	$e_{6,11,24}$		
0-32	$p = 2^{-56}$, $\hat{p} = 2^{-47.6}$					

C.2 Differential of SHA-0 and its Probability

Table C.2: Differential for Rounds 0-44 of SHA-0

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	Δm^i	Prob.
0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1
2	0	0	0	0	0	0	1
3	0	0	0	0	0	e_1	2^{-1}
4	e_1	0	0	0	0	e_6	2^{-1}
5	0	e_1	0	0	0	e_1	2^{-2}
6	0	0	e_{31}	0	0	e_{31}	2^{-1}
7	0	0	0	e_{31}	0	e_{31}	2^{-1}
8	0	0	0	0	e_{31}	$e_{1,31}$	2^{-1}
9	e_1	0	0	0	0	e_6	2^{-1}
10	0	e_1	0	0	0	0	2^{-2}
11	e_1	0	e_{31}	0	0	$e_{6,31}$	2^{-2}
12	0	e_1	0	e_{31}	0	$e_{1,31}$	2^{-3}
13	0	0	e_{31}	0	e_{31}	e_1	2^{-2}
14	e_1	0	0	e_{31}	0	$e_{6,31}$	2^{-2}
15	0	e_1	0	0	e_{31}	$e_{1,31}$	2^{-2}
16	0	0	e_{31}	0	0	e_{31}	2^{-1}
17	0	0	0	e_{31}	0	$e_{1,31}$	2^{-2}
18	e_1	0	0	0	e_{31}	$e_{1,6,31}$	2^{-2}
19	e_1	e_1	0	0	0	e_6	2^{-3}
20	e_1	e_1	e_{31}	0	0	$e_{6,31}$	2^{-2}
21	e_1	e_1	e_{31}	e_{31}	0	$e_{1,6}$	2^{-2}
22	0	e_1	e_{31}	e_{31}	e_{31}	$e_{1,31}$	2^{-1}
23	0	0	e_{31}	e_{31}	e_{31}	$e_{1,31}$	2^{-1}
24	e_1	0	0	e_{31}	e_{31}	e_6	2^{-1}
25	0	e_1	0	0	e_{31}	e_{31}	2^{-1}
26	e_1	0	e_{31}	0	0	$e_{1,6,31}$	2^{-2}
27	e_1	e_1	0	e_{31}	0	$e_{1,6,31}$	2^{-2}
28	0	e_1	e_{31}	0	e_{31}	e_1	2^{-1}
29	0	0	e_{31}	e_{31}	0	0	1
30	0	0	0	e_{31}	e_{31}	0	1
31	0	0	0	0	e_{31}	e_{31}	1
32	0	0	0	0	0	e_1	2^{-1}
33	e_1	0	0	0	0	$e_{1,6}$	2^{-2}
34	e_1	e_1	0	0	0	e_6	2^{-2}
35	e_1	e_1	e_{31}	0	0	$e_{1,6,31}$	2^{-2}
36	0	e_1	e_{31}	e_{31}	0	e_1	2^{-1}
37	0	0	e_{31}	e_{31}	e_{31}	e_{31}	1
38	0	0	0	e_{31}	e_{31}	0	1
39	0	0	0	0	e_{31}	e_{31}	1
40	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
45	0	0	0	0	0	0	1

Table C.3: Differential for Rounds 45-81 of SHA-0 (Extension of the Previous Differential)

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	Δm^i	Prob.
45	0	0	0	0	0	0	1
46	0	0	0	0	0	e_1	2^{-1}
47	e_1	0	0	0	0	$e_{1,6}$	2^{-2}
48	e_1	e_1	0	0	0	$e_{1,6}$	2^{-3}
49	0	e_1	e_{31}	0	0	$e_{1,31}$	2^{-3}
50	0	0	e_{31}	e_{31}	0	0	2^{-1}
51	0	0	0	e_{31}	e_{31}	0	2^{-1}
52	0	0	0	0	e_{31}	e_{31}	1
53	0	0	0	0	0	0	1
54	0	0	0	0	0	e_1	2^{-1}
55	e_1	0	0	0	0	$e_{1,6}$	2^{-2}
56	e_1	e_1	0	0	0	$e_{1,6}$	2^{-3}
57	0	e_1	e_{31}	0	0	e_{31}	2^{-3}
58	e_1	0	e_{31}	e_{31}	0	$e_{1,6}$	2^{-3}
59	e_1	e_1	0	e_{31}	e_{31}	$e_{1,6}$	2^{-4}
60	0	e_1	e_{31}	0	e_{31}	e_1	2^{-1}
61	0	0	e_{31}	e_{31}	0	0	1
62	0	0	0	e_{31}	e_{31}	0	1
63	0	0	0	0	e_{31}	e_{31}	1
64	0	0	0	0	0	0	1
65	0	0	0	0	0	e_1	2^{-1}
0-64	$p = 2^{-78}$, $\hat{p} = 2^{-78}$						
66	e_1	0	0	0	0	$e_{1,6}$	2^{-2}
67	e_1	e_1	0	0	0	$e_{1,6}$	2^{-2}
68	0	e_1	e_{31}	0	0	$e_{1,31}$	2^{-1}
69	0	0	e_{31}	e_{31}	0	0	1
70	0	0	0	e_{31}	e_{31}	e_1	2^{-1}
71	e_1	0	0	0	e_{31}	$e_{6,31}$	2^{-1}
72	0	e_1	0	0	0	0	2^{-1}
73	e_1	0	e_{31}	0	0	$e_{1,6,31}$	2^{-2}
74	e_1	e_1	0	e_{31}	0	$e_{1,6,31}$	2^{-2}
75	0	e_1	e_{31}	0	e_{31}	0	2^{-1}
76	e_1	0	e_{31}	e_{31}	0	$e_{1,6}$	2^{-2}
77	e_1	e_1	0	e_{31}	e_{31}	$e_{1,6}$	2^{-2}
78	0	e_1	e_{31}	0	e_{31}	e_1	2^{-1}
79	0	0	e_{31}	e_{31}	0	0	1
80	0	0	0	e_{31}	e_{31}	0	2^{-1}
81	0	0	0	0	e_{31}	e_{31}	1
82	0	0	0	0	0		
0-81	$q = 2^{-98}$						

C.3 Differential of SHA-1 and its Probability

Table C.4: Differential for Rounds 0-32 on SHA-1

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	Δm^i	Prob.
0	0	0	0	0	0	e_1	2^{-1}
1	e_1	0	0	0	0	e_6	1
2	0	e_1	0	0	0	0	2^{-2}
3	e_1	0	e_{31}	0	0	$e_{6,31}$	2^{-2}
4	0	e_1	0	e_{31}	0	e_{31}	2^{-3}
5	e_1	0	e_{31}	0	e_{31}	e_6	2^{-2}
6	0	e_1	0	e_{31}	0	$e_{0,31}$	2^{-4}
7	$e_{0,1}$	0	e_{31}	0	e_{31}	$e_{5,6}$	2^{-3}
8	0	$e_{0,1}$	0	e_{31}	0	$e_{0,1,31}$	2^{-5}
9	0	0	$e_{30,31}$	0	e_{31}	$e_{1,30}$	2^{-4}
10	e_1	0	0	$e_{30,31}$	0	$e_{6,30,31}$	2^{-4}
11	0	e_1	0	0	$e_{30,31}$	$e_{1,30,31}$	2^{-3}
12	0	0	e_{31}	0	0	e_{31}	2^{-1}
13	0	0	0	e_{31}	0	e_{31}	2^{-1}
14	0	0	0	0	e_{31}	$e_{1,31}$	2^{-1}
15	e_1	0	0	0	0	e_6	2^{-1}
16	0	e_1	0	0	0	e_1	2^{-2}
17	0	0	e_{31}	0	0	e_{31}	2^{-1}
18	0	0	0	e_{31}	0	e_{31}	2^{-1}
19	0	0	0	0	e_{31}	e_{31}	1
20	0	0	0	0	0	e_1	2^{-1}
21	e_1	0	0	0	0	e_6	2^{-1}
22	0	e_1	0	0	0	0	2^{-1}
23	e_1	0	e_{31}	0	0	$e_{6,31}$	2^{-1}
24	0	e_1	0	e_{31}	0	$e_{1,31}$	2^{-1}
25	0	0	e_{31}	0	e_{31}	0	1
26	0	0	0	e_{31}	0	e_{31}	1
27	0	0	0	0	e_{31}	e_{31}	1
28	0	0	0	0	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
33	0	0	0	0	0	0	1

Table C.5: Differential for Rounds 33-42 on SHA-1 (Extension of the Previous Differential)

Round (i)	ΔA^i	ΔB^i	ΔC^i	ΔD^i	ΔE^i	Δm^i	Prob.
33	0	0	0	0	0	0	1
34	0	0	0	0	0	e_2	2^{-1}
35	e_2	0	0	0	0	e_7	2^{-1}
36	0	e_2	0	0	0	e_2	2^{-1}
37	0	0	e_0	0	0	$e_{0,3}$	2^{-2}
38	e_3	0	0	e_0	0	$e_{0,2,8}$	2^{-3}
39	e_2	e_3	0	0	e_0	$e_{0,3,7}$	2^{-3}
40	0	e_2	e_1	0	0	$e_{1,2,4}$	2^{-5}
41	e_4	0	e_0	e_1	0	$e_{0,1,3,9}$	2^{-6}
42	e_3	e_4	0	e_0	e_1	$e_{0,1,3,4,8}$	2^{-7}
43	e_3	e_3	e_2	0	e_0		
0-42	$p = 2^{-75}$, $\hat{p} = 2^{-73.4}$						

Bibliography

[1] K. Aoki and K. Ohta, *Strict Evaluation of the Maximum Average of Differential Probability and the Maximem Average of Linear Probability*, IEICE Transactions Fundamentals of Electronics, Communications and Computer Sciences, Vol. E-80A, No. 1, pp. 2-8, 1997.

[2] M. Bellare, *New Proofs for NMAC and HMAC: Security without Collision-Resistance*, Advances in Cryptology – Proceedings of CRYPTO 2006, to appear, and Cryptology ePrint Archive, Report 2006/043, Available Online at *http://eprint.iacr.org/2006/043.pdf*.

[3] M. Bellare, R. Canetti and H. Krawczyk, *Keying Hash Functions for Message Authentication*, Advances in Cryptology – Proceedings of CRYPTO 1996, LNCS 1109, pp. 1-15, Springer-Verlag, 1996.

[4] M. Bellare, J. Kilian and P. Rogaway, *The Security of the Cipher Block Chaining Message Authentication Code*, Journal of Computer and System Sciences, Vol. 61, No. 3, pp. 362-399, 2000.

[5] M. Bellare, P. Rogaway and D. Wagner, *The EAX Mode of Operation*, Proceedings of FSE 2004, LNCS 3017, pp. 389-407, Springer-Verlag, 2004.

[6] D.J. Bernstein, *The Poly1305-AES Message-Authentication Code*, Proceedings of FSE 2005, LNCS 3557, pp. 32-49, Springer-Verlag, 2005.

[7] E. Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Journal of Crytology, Vol. 7, No. 4, pp. 229-246, 1994.

[8] E. Biham, A. Biryukov and A. Shamir, *Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials*, Journal of Crytology, Vol. 18, No. 4, pp. 291-311, 2005.

[9] E. Biham and R. Chen, *Near-Collisions of SHA-0*, Advances in Cryptology – Proceedings of CRYPTO 2004, LNCS 3152, pp. 290-305, Springer-Verlag, 2004.

[10] E. Biham, R. Chen, A. Joux, P. Carribault, C. Lemuet and W. Jalby, *Collisions of SHA-0 and Reduced SHA-1*, Advances in Cryptology – Proceedings of EUROCRYPT 2005, LNCS 3494, pp. 22-35, Springer-Verlag, 2005.

[11] E. Biham, O. Dunkelman and N. Keller, *The Rectangle Attack - Rectangling the Serpent*, Advances in Cryptology – Proceedings of EUROCRYPT 2001, LNCS 2045, pp. 340-357, Springer-Verlag, 2001.

[12] E. Biham, O. Dunkelman and N. Keller, *Enhanced Differential-Linear Cryptanalysis*, Advances in Cryptology – Proceedings of ASIACRYPT 2002, LNCS 2501, pp. 254-266, Springer-Verlag, 2002.

[13] E. Biham, O. Dunkelman and N. Keller, *Rectangle Attacks on 49-Round SHACAL-1*, Proceedings of FSE 2003, LNCS 2887, pp. 22-35, Springer-Verlag, 2003.

[14] E. Biham, O. Dunkelman and N. Keller, *New Combined Attacks on Block Ciphers*, Proceedings of FSE 2005, LNCS 3557, pp. 126-144, Springer-Verlag, 2005.

[15] E. Biham, O. Dunkelman and N. Keller, *Related-Key Boomerang and Rectangle Attacks*, Advances in Cryptology – Proceedings of EUROCRYPT 2005, LNCS 3494, pp. 507-525, Springer-Verlag, 2005.

[16] E. Biham, O. Dunkelman and N. Keller, *Related-Key Impossible Differential Attacks on AES-192*, Topics in Cryptology – Proceedings of CT-RSA 2006, LNCS 3860, pp. 21-31, Springer-Verlag, 2006.

[17] E. Biham, O. Dunkelman and N. Keller, *A Simple Related-Key Attack on the Full SHACAL-1*, Topics in Cryptology – Proceedings of CT-RSA 2007, to appear.

[18] E. Biham and A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, Advances in Cryptology – Proceedings of CRYPTO 1990, LNCS 537, pp. 2-21, Springer-Verlag, 1990.

[19] E. Biham and A. Shamir, *Differential Cryptanalysis of the Full 16-Round DES*, Advances in Cryptology – Proceedings of CRYPTO 1992, LNCS 740, pp. 487-496, Springer-Verlag, 1992.

[20] J. Black, S. Halevi, H. Krawczyk, T. Krovetz and P. Rogaway, *UMAC: Fast and Secure Message Authentication*, Advances in Cryptology – Proceedings of CRYPTO 1999, LNCS 1666, pp. 216-233, Springer-Verlag, 1999.

[21] A. Biryukov, *Methods of Cryptanalysis*, Doctoral dissertation, Technion, 1999.

[22] A. Biryukov and A. Shamir, *Structural Cryptanalysis of SASAS*, Advances in Cryptology – Proceedings of EUROCRYPT 2001, LNCS 2045, pp. 394-405, Springer-Verlag, 2001.

[23] M. Blunden and A. Escott, *Related Key Attacks on Reduced Round KASUMI*, Proceedings of FSE 2001, LNCS 2355, pp. 277-285, Springer-Verlag, 2001.

[24] F. Chabaud and A. Joux, *Differential Collisions in SHA-0*, Advances in Cryptology – Proceedings of CRYPTO 1998, LNCS 1462, pp. 56-71, Springer-Verlag, 1999.

[25] J. Choi, J. Kim, J. Sung, S. Lee and J. Lim, *Related-Key and Meet-in-the-Middle Attacks on Triple-DES and DES-EXE*, Proceedings of ICCSA 2005, LNCS 3481, pp. 567-576, Springer-Verlag, 2005.

[26] S. Contini and Y.L. Yin, *Forgery and Partial Key-Recovery Attacks on HMAC and NMAC Using Hash Collisions*, Advances in Cryptology – Proceedings of ASIACRYPT 2006, to appear, and Cryptology ePrint Archive, Report 2006/319, Available Online at *http://eprint.iacr.org/2006/319.pdf*.

[27] N.T. Courtois, *Feistel Schemes and Bi-Linear Cryptanalysis*, Advances in Cryptology – Proceedings of CRYPTO 2004, LNCS 3152, pp. 23-40, Springer-Verlag, 2004.

[28] N.T. Courtois and J. Pieprzyk, *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*, Advances in Cryptology – Proceedings of ASIACRYPT 2002, LNCS 2501, pp. 267-287, Springer-Verlag, 2002.

[29] J. Daemen, L.R. Knudsen and V. Rijmen, *The Block Cipher Square*, Proceedings of FSE 1997, LNCS 1267, pp. 149-165, Springer-Verlag, 1997.

[30] J. Daemen and V. Rijmen, *The Design of Rijndael*, Springer-Verlag, 2002.

[31] I. Damgård, *A Design Principle for Hash Functions*, Advances in Cryptology – Proceedings of CRYPTO 1989, LNCS 435, pp. 416-427, Springer-Verlag, 1989.

[32] W. Diffie and M.E. Hellman, *New Directions in Cryptography*, IEEE Transactions on Information Theory, Vol. 22, No. 6, pp. 644-654, 1976.

[33] H. Dobbertin, A. Bosselaers and B. Preneel, *RIPEMD-160: A Strengthened Version of RIPEMD*, Proceedings of FSE 1996, LNCS 1039, pp. 71-82, Springer-Verlag, 1996.

[34] O. Dunkelman, *Techniques for Cryptanalysis of Block Ciphers*, Doctoral dissertation, Technion, 2006.

[35] O. Dunkelman, N. Keller and J. Kim, *Related-Key Rectangle Attack on the Full SHACAL-1*, Proceedings of SAC 2006, to appear.

[36] N. Ferguson, J. Kelsey, B. Schneier, M. Stay, D. Wagner and D. Whiting, *Improved Cryptanalysis of Rijndael*, Proceedings of FSE 2000, LNCS 1978, pp. 213-230, Springer-Verlag, 2001.

[37] A. Fiat and A. Shamir, *How to Prove Yourself: Practical Solutions to Identification and Signature Problems*, Advances in Cryptology – Proceedings of CRYPTO 1986, LNCS 263, pp. 186-194, Springer-Verlag, 1987.

[38] FIPS 46-3: *Data Encryption Standard*, NIST, Oct. 1999, Specifies the use of Triple-DES.

[39] FIPS 186-2: *Digital Signature Standard (DSS)*, NIST, Jan. 2000.

[40] FIPS 180: *Secure Hash Standard, Federal Information Processing Standards Publication*, NIST, May 1993.

[41] FIPS 180-1: *Secure Hash Standard, Federal Information Processing Standards Publication*, NIST, April 1995.

[42] FIPS 180-2: *Secure Hash Standard, Federal Information Processing Standards Publication*, NIST, August 2002.

[43] H. Handschuh, L.R. Knudsen and M.J. Robshaw, *Analysis of SHA-1 in Encryption Mode*, Proceedings of CT-RSA 2001, LNCS 2020, pp. 70-83, Springer-Verlag, 2001.

[44] H. Handschuh and D. Naccache, *SHACAL*, preproceedings of NESSIE first workshop, Leuven, 2000.

[45] H. Handschuh and D. Naccache, *SHACAL: A Family of Block Ciphers*, Submission to the NESSIE project, 2002.

[46] P. Hawkes, *Differential-Linear Weak-Key Classes of IDEA*, Advances in Cryptology – Proceedings of EUROCRYPT 1998, LNCS 1403, pp. 112-126, Springer-Verlag, 1998.

[47] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim and S. Chee, *HIGHT: A New Block Cipher Suitable for Low-Resource Device*, Proceedings of CHES 2006, LNCS 4249, pp. 46-59, Springer-Verlag, 2006.

[48] S. Hong, J. Kim, G. Kim, J. Sung, C. Lee and S. Lee, *Impossible Differential Attack on 30-Round SHACAL-2*, Progress in Cryptology – Proceedings of INDOCRYPT 2003, LNCS 2904, pp. 97-106, Springer-Verlag, 2003.

[49] S. Hong, J. Kim, S. Lee and B. Preneel, *Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192*, Proceedings of FSE 2005, LNCS 3557, pp. 368-383, Springer-Verlag, 2005.

[50] S. Hong, S. Lee, J. Lim, J. Sung, D. Choen and I. Cho, *Provable Security against Differential and Linear Cryptanalysis for the SPN Structure*, Proceedings of FSE 2000, LNCS 1978, pp. 273-283, Springer-Verlag, 2001.

[51] S. Hong, J. Sung, S. Lee, J. Lim and J. Kim, *Provable Security for 13 round Skipjack-like Structure*, Information Processing Letters, Vol. 82, No. 5, pp. 243-246, 2002.

[52] H. Im, G. Kim, J. Kim, J. Sung and S. Lee, *Related-Key Differential-Linear Attack on DES*, Proceedings of CISC 2004, Conference on Information Security and Cryptology, Vol. 14, No. 1, pp. 511-514, 2004.

[53] ISO/IEC 9797, *Data Cryptographic Techniques – Data Integrity Mechanism Using a Cryptographic Check Function Employing a Block Cipher Algorithm*, 1989.

[54] T. Iwata and K. Kurosawa, *OMAC: One-Key CBC MAC*, Proceedings of FSE 2003, LNCS 2887, pp. 129-153. Springer-Verlag, 2003.

[55] G. Jakimoski and Y. Desmedt, *Related-Key Differential Cryptanalysis of 192-bit Key AES Variants*, Proceedings of SAC 2003, LNCS 3006, pp. 208-221, Springer-Verlag, 2004.

[56] E. Jaulmes, A. Joux and F. Valette, *On the Security of Randomized CBC-MAC Beyond the Birthday Paradox Limit: A New Construction*, Proceedings of FSE 2002, LNCS 2365, pp. 237-251, Springer-Verlag, 2002.

[57] B.S. Kaliski Jr. and M.J.B. Robshaw, *Linear Cryptanalysis Using Multiple Approximations*, Advances in Cryptology – Proceedings of CRYPTO 1994, LNCS 839, pp. 26-39, Springer-Verlag, 1994.

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

[59] J. Kelsey, B. Schneier and D. Wagner, *Key Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES*, Advances in Cryptology – Proceedings of CRYPTO 1996, LNCS 1109, pp. 237-251, Springer-Verlag, 1996.

[60] J. Kelsey, B. Schneir and D. Wagner, *Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA*, Proceedings of ICICS 1997, LNCS 1334, pp. 233-246, Springer-Verlag, 1997.

[61] G. Kim, J. Kim, S. Hong and S. Lee, *Cryptanalysis of SHACAL-1*, 14th Joint Conference on Communications and Information (JCCI), 2004.

[62] G. Kim, J. Kim, J. Sung, S. Lee and J. Lim, *Provable Security for Feistel-variant Structures against Differential Cryptanalysis and Linear Cryptanalysis*, Proceedings of CISC 2004, Conference on Information Security and Cryptology, Vol. 14, No. 1, pp. 341-345, 2004.

[63] J. Kim, A. Biryukov, B. Preneel and S. Hong, *On the Security of HMAC and NMAC Based on HAVAL, MD4, MD5, SHA-0 and SHA-1*, Proceedings of SCN 2006, LNCS 4116, pp. 242-256, Springer-Verlag, 2006, and Cryptology ePrint Archive, Report 2006/187, Available Online at *http://eprint.iacr.org/2006/187.pdf*.

[64] J. Kim, A. Biryukov, B. Preneel and S. Lee, *On the Security of Encryption Modes of MD4, MD5 and HAVAL*, Proceedings of ICICS 2005, LNCS 3783, pp. 147-158, Springer-Verlag, 2005, and Cryptology ePrint Archive, Report 2005/327, Available Online at *http://eprint.iacr.org/2005/327.ps*.

[65] J. Kim, S. Hong, S. Lee, J. Song and H. Yang, *Truncated Differential Attacks on 8-Round CRYPTON*, Proceedings of ICISC 2003, LNCS 2971, pp. 446-456, Springer-Verlag, 2004.

[66] J. Kim, S. Hong, J. Sung, S. Lee, J. Lim and S. Sung, *Impossible Differential Cryptanalysis for Block Cipher Structures*, Progress in Cryptology – Proceedings of INDOCRYPT 2003, LNCS 2904, pp. 82-96, Springer-Verlag, 2003.

[67] J. Kim, G. Kim, S. Hong, S. Lee and D. Hong, *The Related-Key Rectangle Attack – Application to SHACAL-1*, Proceedings of ACISP 2004, LNCS 3108, pp. 123-136, Springer-Verlag, 2004.

[68] J. Kim, G. Kim, S. Lee, J. Lim and J. Song, *Related-Key Attacks on Reduced Rounds of SHACAL-2*, Proceedings of INDOCRYPT 2004, LNCS 3348, pp. 175-189, Springer-Verlag, 2004.

[69] J. Kim, D. Moon, W. Lee, S. Hong, S. Lee and S. Jung, *Amplified Boomerang Attack against Reduced-Round SHACAL*, Advances in Cryptology – Proceedings of ASIACRYPT 2002, LNCS 2501, pp. 243-253, Springer-Verlag, 2002.

[70] T. Kim, G. Kim, J. Kim, J. Sung and S. Lee, *Collision Attacks on CRYPTON*, 15th Workshop on Information Security and Cryptology (WISC 2004), 2004.

[71] T. Kim, T. Lee, C. Lee, J. Kim, J. Sung and S. Lee, *Padding Oracle Attacks on the CBC Modes of Operation for the Block Cipher SEED*, Proceedings of CISC 2004, Conference on Information Security and Cryptology, Vol. 14, No. 1, pp. 346-351, 2004.

[72] L.R. Knudsen, *Cryptanalysis of LOKI91*, Advances in Cryptology – Proceedings of AUSCRYPT 1992, LNCS 718, pp. 196-208, Springer-Verlag, 1993.

[73] L.R. Knudsen, *Block Ciphers – Analysis, Design and Applications*, Doctoral dissertation, Aarhus University, 1994.

[74] L.R. Knudsen, *Trucated and Higher Order Differentials*, Proceedings of FSE 1994, LNCS 1008, pp. 196-211, Springer-Verlag, 1995.

[75] L.R. Knudsen and J.E. Mathiassen, *A Chosen-Plaintext Linear Attack on DES*, Proceedings of FSE 2000, LNCS 1978, pp. 262-272, Springer-Verlag, 2001.

[76] L.R. Knudsen and M.J.B. Robshaw, *Non-Linear Approximations in Linear Cryptanalysis*, Advances in Cryptology – Proceedings of EUROCRYPT 1996, LNCS 1070, pp. 224-236, Springer-Verlag, 1996.

[77] L.R. Knudsen and D. Wagner, *Integral Cryptanalysis*, Proceedings of FSE 2002, LNCS 2365, pp. 112-127, Springer-Verlag, 2002.

[78] Y. Ko, S. Hong, W. Lee, S. Lee and J. Kang, *Related Key Differential Attacks on 26 Rounds of XTEA and Full Rounds of GOST*, Proceedings of FSE 2004, LNCS 3017, pp. 299-316, Springer-Verlag, 2004.

[79] N. Koblitz, *Elliptic Curve Cryptosystems*, Mathematics of Computation, Vol. 48, No. 177, pp. 203-209, 1987.

[80] T. Kohno, J. Viega and D. Whiting, *CWC: A High-Performance Conventional Authenticated Encryption Mode*, Proceedings of FSE 2004, LNCS 3017, pp. 408-426, Springer-Verlag, 2004.

[81] K. Kurosawa and T. Iwata, *TMAC: Two-Key CBC MAC*, Topics in Cryptology – Proceedings of CT-RSA 2003, LNCS 2612, pp. 33-49, Springer-Verlag, 2003.

[82] S.K. Langford and M.E. Hellman, *Differential-Linear Cryptanalysis*, Advances in Cryptology – Proceedings of CRYPTO 1994, LNCS 839, pp. 17-25, Springer-Verlag, 1994.

[83] C. Lee, J. Kim, S. Hong, J. Sung and S. Lee, *Related-Key Differential Attacks on Cobra-S128, Cobra-F64a, and Cobra-F64b*, Proceedings of Mycrypt 2005, LNCS 3715, pp. 244-262, Springer-Verlag, 2005.

[84] C. Lee, J. Kim, J. Sung, S. Hong and S. Lee, *Provable Security for an RC6-like Structure and a MISTY-FO-like Structure against Differential Cryptanalysis*, Proceedings of ICCSA 2006, LNCS 3982, pp. 446-455, Springer-Verlag, 2006.

[85] C. Lee, J. Kim, J. Sung, S. Hong and S. Lee, *Forgery and Key Recovery Attacks on PMAC and Mitchell's TMAC Variant*, Proceedings of ACISP 2006, LNCS 4058, pp. 421-431, Springer-Verlag, 2006.

[86] C. Lee, J. Kim, J. Sung, S. Hong, S. Lee and D. Moon, *Related-Key Differential Attacks on Cobra-H64 and Cobra-H128*, Proceedings of CCC 2005, LNCS 3796, pp. 201-219, Springer-Verlag, 2005.

[87] T. Lee, J. Kim, C. Lee, J. Sung, S. Lee and D. Hong, *Padding Oracle Attacks on Multiple Modes of Operation*, Proceedings of ICISC 2004, LNCS 3506, pp. 343-351, Springer-Verlag, 2004.

[88] H. Lipmaa, *On Differential Properties of Pseudo-Hadamard Transform and Related Mappings*, Progress in Cryptology – Proceedings of INDOCRYPT 2002, LNCS 2551, pp. 48-61, Springer-Verlag, 2002.

[89] J. Lu, J. Kim, N. Keller and O. Dunkelman, *Related-Key Rectangle Attack on 42-Round SHACAL-2*, Proceedings of ISC 2006, LNCS 4176, pp. 85-100, Springer-Verlag, 2006.

[90] J. Lu, J. Kim, N. Keller and O. Dunkelman, *Differential and Rectangle Attacks on Reduced-Round SHACAL-1*, Proceedings of INDOCRYPT 2006, to appear.

[91] J. Lu, C. Lee and J. Kim, *Related-Key Attacks on the Full-Round Cobra-F64a and Cobra-F64b*, Proceedings of SCN 2006, LNCS 4116, pp. 95-110, Springer-Verlag, 2006.

[92] S. Lucks, *Attacking Seven Rounds of Rijndael under 192-Bit and 256-Bit Keys*, Proceedings of AES3, NIST.

[93] S. Lucks, *The Saturation Attack – a Bait for Twofish*, Proceedings of FSE 2001, LNCS 1039, pp. 189-203, Springer-Verlag, 2001.

[94] M. Matsui, *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology – Proceedings of EUROCRYPT 1993, LNCS 765, pp. 386-397, Springer-Verlag, 1994.

[95] M. Matsui, *New Structure of Block Ciphers with Provable Security against Differential and Linear Cryptanalysis*, Proceedings of FSE 1996, LNCS 1039, pp. 205-218, Springer-Verlag, 1996.

[96] D.A. McGrew and J. Viega, *The Galois/Counter Mode of Operation (GCM)*, *http://csrc.nist.gov/CryptoToolkit/modes/proposedmodes/gcm-spec.pdf*.

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

[98] R.C. Merkle, *One Way Hash Functions and DES*, Advances in Cryptology – Proceedings of CRYPTO 1989, LNCS 435, pp. 428-446, Springer-Verlag, 1989.

[99] NESSIE — NEW European Schemes for Signatures, Integrity and Encryption, http://www.cryptonessie.org.

[100] B.C. Neuman and T. Ts'o, *Kerberos: An Authentication Service for Computer Networks*, IEEE Communications, Vol. 32, No. 9, pp. 33-38, 1994.

[101] K. Nyberg *Generalized Feistel Networks*, Advances in Cryptology – Proceedings of ASIACRYPT 1996, LNCS 1163, pp. 91-104, Springer-Verlag, 1996.

[102] K. Nyberg and L.R. Knudsen, *Provable Security Against a Differential Attack*, Journal of Cryptology, Vol. 8, No. 1, pp. 27-37, 1995.

[103] R. C.-W. Phan, *Impossible Differential Cryptanalysis of 7-Round Advanced Encryption Standard (AES)*, Information Processing Letters, Vol. 91, No. 1, pp. 33-38, Elsevier, 2004.

[104] R. C.-W. Phan and H. Handschuh, *On Related-Key and Collision Attacks: The Case for the IBM 4758 Cryptoprocessor*, Proceedings of ISC 2004, LNCS 3225, pp. 111-122, Springer-Verlag, 2004.

[105] R. C.-W. Phan and S. Yen, *Amplifying Side-Channel Attacks with Techniques from Block Cipher Cryptanalysis*, Proceedings of CARDIS 2006, LNCS 3928, pp. 135-150, Springer-Verlag, 2006.

[106] B. Preneel, *Analysis and Design of Cryptographic Hash Functions*, Doctoral dissertation, K.U. Leuven, 1993.

[107] B. Preneel, A. Bosselaers, R. Govaerts and J. Vandewalle, *A Chosen Text Attack on the Modified Cryptographic Checksum Algorithm of Cohen and Huang*, Advances in Cryptology – Proceedings of CRYPTO 1989, LNCS 435, pp. 154-163, Springer-Verlag, 1990.

[108] B. Preneel and P.C. van Oorschot, *On the Security of Iterated Message Authentication Codes*, IEEE Transactions on Information Theory, Vol. 45, No. 1, pp. 188-99, 1999 (Preliminary version, entitled *MDx-MAC and Building Fast MACs from Hash Functions*, Advances in Cryptology – Proceedings of CRYPTO 1995, LNCS 963, pp. 1-14, Springer-Verlag, 1995).

[109] C. Rechberger and V. Rijmen, *Note on Distinguishing, Forgery, and Second Preimage Attacks on HMAC-SHA-1 and a Method to Reduce the Key Entropy of NMAC*, Crypto eprint archive, Report 2006/290, Available Online at *http://eprint.iacr.org/2006/290.pdf*.

[110] RIPE, *Integrity Primitives for Secure Information Systems*, Final Report of RACE Integrity Primitives Evaluation (RIPE-RACE 1040), LNCS 1007, 1995.

[111] R.L. Rivest, *The MD4 Message Digest Algorithm*, Advances in Cryptology – Proceedings of CRYPTO 1990, pp. 303-311, Springer-Verlag, 1991.

[112] R.L. Rivest, *The MD5 Message Digest Algorithm*, Request for Comments (RFC 1320), Internet Activities Board, Internet Privacy Task Force, 1992.

[113] R.L. Rivest, *The RC4 Encryption Algorithm*, RSA Data Security, Inc., March 12, 1992.

[114] R.L. Rivest, A. Shamir and L. Adleman, *A Method for Obtaining Digital Signatures and Public Key Cryptosystems*, Commuincations of the ACM, Vol. 21, No. 2, pp. 120-126, 1978.

[115] G. Rose, *A Stream Cipher Based on Linear Feedback over* $GF(2^8)$, Proceedings of ACISP 1998, LNCS 1438, pp. 135-146, Springer-Verlag, 1998.

[116] M.J.O. Saarinen, *Cryptanalysis of Block Ciphers Based on SHA-1 and MD5*, Proceedings of FSE 2003, LNCS 2887, pp. 36-44, Springer-Verlag, 2003.

[117] A.A. Selcuk and A. Bicak, *On Probability of Success in Linear and Differential Cryptanalysis*, Proceedings of SCN 2002, LNCS 2576, pp. 174-185, Springer-Verlag, 2002.

[118] C.E. Shannon, *Communication Theory of Secrecy Systems*, Bell System Technical Journal, Vol. 28, pp. 656-715, 1949.

[119] Y. Shin, J. Kim, G. Kim, S. Hong and S. Lee, *Differential-Linear Type Attacks on Reduced Rounds of SHACAL-2*, Proceedings of ACISP 2004, LNCS 3108, pp. 110-122, Springer-Verlag, 2004.

[120] *Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation: Methods and Techniques*, NIST, Available Online at *http://csrc.nist.gov/publications/fips/*.

[121] J. Sung, J. Kim, C. Lee and S. Hong, *Related-Cipher Attacks on Block Ciphers with Flexible Number of Rounds*, WEWoRC 2005 – Western European Workshop on Research in Cryptology, Lecture Notes in Informatics (LNI 74), pp. 64-75, 2005.

[122] J. Sung, S. Lee, J. Lim, S. Hong and S. Park, *Provable Security for the Skipjack-like Structure against Differential Cryptanalysis and Linear Cryptanalysis*, Advances in Cryptology – Proceedings of ASIACRYPT 2000, LNCS 1976, pp. 274-288, Springer-Verlag, 2000.

[123] D. Wagner, *The Boomerang Attack*, Proceedings of FSE 1999, LNCS 1636, pp. 156-170, Springer-Verlag, 1999.

[124] X. Wang, X. Lai, D. Feng, H. Chen and X. Yu, *Cryptanalysis of the Hash Functions MD4 and RIPEMD*, Advances in Cryptology – Proceedings of EUROCRYPT 2005, LNCS 3494, pp. 1-18, Springer-Verlag, 2005.

[125] X. Wang, Y.L. Yin and H. Yu, *Finding Collisions in the Full SHA-1*, Advances in Cryptology – Proceedings of CRYPTO 2005, LNCS 3621, pp. 17-36, Springer-Verlag, 2005.

[126] X. Wang and H. Yu, *How to Break MD5 and Other Hash Functions*, Advances in Cryptology – Proceedings of EUROCRYPT 2005, LNCS 3494, pp. 19-35, Springer-Verlag, 2005.

[127] X. Wang, H. Yu and Y.L. Yin, *Efficient Collision Search Attacks on SHA-0*, Advances in Cryptology – Proceedings of CRYPTO 2005, LNCS 3621, pp. 1-16, Springer-Verlag, 2005.

[128] H. Yoshida, A. Biryukov, C. De Cannière, J. Lano and B. Preneel, *Non-randomness of the Full 4 and 5-pass HAVAL*, Proceedings of SCN 2004, LNCS 3352, pp. 324-336, Springer-Verlag, 2005.

[129] H. Yu, G. Wang, G. Zhang and X. Wang, *The Second-Preimage Attack on MD4*, Proceedings of CANS 2005, LNCS 3810, pp. 1-12, Springer-Verlag, 2005.

[130] Y. Zheng, J. Pieprzyk and J. Seberry, *HAVAL-A One-way Hashing Algorithm with Variable Length of Output*, Advances in Cryptology – Proceedings of AUSCRYPT 1992, LNCS 718, pp. 83-104, Springer-Verlag, 1993.

List of Publications

Lecture Notes in Computer Science

1. **Jongsung Kim**, Dukjae Moon, Wonil Lee, Seokhie Hong, Sangjin Lee and Seokwon Jung, *Amplified Boomerang Attack against Reduced-Round SHACAL*, Advances in Cryptology – Proceedings of ASIACRYPT 2002, LNCS 2501, pp. 243-253, Springer-Verlag, 2002.

2. **Jongsung Kim**, Seokhie Hong, Jaechul Sung, Sangjin Lee, Jongin Lim and Soohak Sung, *Impossible Differential Cryptanalysis for Block Cipher Structures*, Progress in Cryptology – Proceedings of INDOCRYPT 2003, LNCS 2904, pp. 82-96, Springer-Verlag, 2003.

3. Seokhie Hong, **Jongsung Kim**, Guil Kim, Jaechul Sung, Changhoon Lee and Sangjin Lee, *Impossible Differential Attack on 30-round SHACAL-2*, Progress in Cryptology – Proceedings of INDOCRYPT 2003, LNCS 2904, pp. 97-106, Springer-Verlag, 2003.

4. **Jongsung Kim**, Seokhie Hong, Sangjin Lee, Junghwan Song and Hyungjin Yang, *Truncated Differential Attacks on 8-Round CRYPTON*, Proceedings of ICISC 2003, LNCS 2971, pp. 446-456, Springer-Verlag, 2004.

5. **Jongsung Kim**, Guil Kim, Seokhie Hong, Sangjin Lee and Dowon Hong, *The Related-Key Rectangle Attack – Application to SHACAL-1*, Proceedings of ACISP 2004, LNCS 3108, pp. 123-136, Springer-Verlag, 2004.

6. Yongsup Shin, **Jongsung Kim**, Guil Kim, Seokhie Hong and Sangjin Lee, *Differential-Linear Type Attacks on Reduced Rounds of SHACAL-2*, Proceedings of ACISP 2004, LNCS 3108, pp. 110-122, Springer-Verlag, 2004.

7. **Jongsung Kim**, Guil Kim, Sangjin Lee, Jongin Lim and Junghwan Song, *Related-Key Attacks on Reduced Rounds of SHACAL-2*, Progress in Cryptology – Proceedings of INDOCRYPT 2004, LNCS 3348, pp. 175-189, Springer-Verlag, 2004.

8. Taekeon Lee, **Jongsung Kim**, Changhoon Lee, Jaechul Sung, Sangin Lee and Dowon Hong, *Padding Oracle Attacks on Multiple Modes of Operation*, Proceedings of ICISC 2004, LNCS 3506, pp. 343-351, Springer-Verlag, 2004.

9. Seokhie Hong, **Jongsung Kim**, Sangjin Lee and Bart Preneel, *Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192*, Proceedings of FSE 2005, LNCS 3557, pp. 368-383, Springer-Verlag, 2005.

10. Jaemin Choi, **Jongsung Kim**, Jaechul Sung, Sangjin Lee and Jongin Lim, *Related-Key and Meet-in-the-Middle Attacks on Triple-DES and DES-EXE*, Proceedings of ICCSA 2005, LNCS 3481, pp. 567-576, Springer-Verlag, 2005.

11. Changhoon Lee, **Jongsung Kim**, Seokhie Hong, Jaechul Sung and Sangjin Lee, *Related-Key Differential Attacks on Cobra-S128, Cobra-F64a, and Cobra-F64b*, Proceedings of Mycrypt 2005, LNCS 3715, pp. 244-262, Springer-Verlag, 2005.

12. Changhoon Lee, **Jongsung Kim**, Jaechul Sung, Seokhie Hong, Sangjin Lee and Dukjae Moon, *Related-Key Differential Attacks on Cobra-H64 and Cobra-H128*, Proceedings of CCC 2005, LNCS 3796, pp. 201-219, Springer-Verlag, 2005.

13. **Jongsung Kim**, Alex Biryukov, Bart Preneel and Sangjin Lee, *On the Security of Encryption Modes of MD4, MD5 and HAVAL*, Proceedings of ICICS 2005, LNCS 3783, pp. 147-158, Springer-Verlag, 2005.

14. Changhoon Lee, **Jongsung Kim**, Jaechul Sung, Seokhie Hong and Sangjin Lee, *Provable Security for an RC6-like Structure and a MISTY-FO-like Structure against Differential Cryptanalysis*, Proceedings of ICCSA 2006, LNCS 3982, pp. 446-455, Springer-Verlag, 2006.

15. Changhoon Lee, **Jongsung Kim**, Jaechul Sung, Seokhie Hong and Sangjin Lee, *Forgery and Key Recovery Attacks on PMAC and Mitchell's TMAC Variant*, Proceedings of ACISP 2006, LNCS 4058, pp. 421-431, Springer-Verlag, 2006.

16. Jiqiang Lu, **Jongsung Kim**, Nathan Keller and Orr Dunkelman, *Related-Key Rectangle Attack on 42-Round SHACAL-2*, Proceedings of ISC 2006, LNCS 4176, pp. 85-100, Springer-Verlag, 2006.

17. **Jongsung Kim**, Alex Biryukov, Bart Preneel and Seokhie Hong, *On the Security of HMAC and NMAC Based on HAVAL, MD4, MD5, SHA-0 and SHA-1*, Proceedings of SCN 2006, LNCS 4116, pp. 242-256, Springer-Verlag, 2006.

18. Jiqiang Lu, Changhoon Lee and **Jongsung Kim**, *Related-Key Attacks on the Full-Round Cobra-F64a and Cobra-F64b*, Proceedings of SCN 2006, LNCS 4116, pp. 95-110, Springer-Verlag, 2006.

19. 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*, Proceedings of CHES 2006, LNCS 4249, pp. 46-59, Springer-Verlag, 2006.

20. Orr Dunkelman, Nathan Keller and **Jongsung Kim**, *Related-Key Rectangle Attack on the Full SHACAL-1*, Proceedings of SAC 2006, to appear.

21. Jiqiang Lu, **Jongsung Kim**, Nathan Keller and Orr Dunkelman, *Differential and Rectangle Attacks on Reduced-Round SHACAL-1*, Proceedings of INDOCRYPT 2006, to appear.

International Conferences/Workshops

1. **Jongsung Kim**, Dukjae Moon, Wonil Lee, Seokhie Hong, Sangjin Lee and Seokwon Jung, *Amplified Boomerang Attack against Reduced-Round SHACAL*, Third NESSIE Workshop, 2002.

2. Jaechul Sung, **Jongsung Kim**, Changhoon Lee and Seokhie Hong, *Related-Cipher Attacks on Block Ciphers with Flexible Number of Rounds*, WEWoRC 2005 – Western European Workshop on Research in Cryptology, Lecture Notes in Informatics (LNI 74), pp. 64-75, 2005.

Domestic Conferences/Workshops

1. Guil Kim, **Jongsung Kim**, Seokhie Hong and Sangjin Lee, *Cryptanalysis of SHACAL-1*, 14th Joint Conference on Communications and Information (JCCI), 2004.

2. Guil Kim, **Jongsung Kim**, Jaechul Sung, Sangjin Lee and Jongin Lim, *Provable Security for Feistel-variant Structures against Differential Cryptanalysis and Linear Cryptanalysis*, Proceedings of CISC 2004, Conference on Information Security and Cryptology, Vol. 14, No. 1, pp. 341-345, 2004.

3. Taewoong Kim, Taekeon Lee, Changhoon Lee, **Jongsung Kim**, Jaechul Sung and Sangjin Lee, *Padding Oracle Attacks on the CBC Modes of Operation for the Block Cipher SEED*, Proceedings of CISC 2004, Conference on Information Security and Cryptology, Vol. 14, No. 1, pp. 346-351, 2004.

4. Hungsu Im, Guil Kim, **Jongsung Kim**, Jaechul Sung and Sangjin Lee, *Related-Key Differential-Linear Attack on DES*, Proceedings of CISC 2004, Conference on Information Security and Cryptology, Vol. 14, No. 1, pp. 511-514, 2004.

5. Taewoong Kim, Guil Kim, **Jongsung Kim**, Jaechul Sung and Sangjin Lee, *Collision Attacks on CRYPTON*, 15th Workshop on Information Security and Cryptology (WISC 2004), 2004.

CV

Jongsung Kim was born on Feb. 4, 1978 in South Korea. He obtained his bachelor and master degrees in mathematics from Korea University in 2000 and 2002, respectively. His master thesis is on *Provable Security for Block Cipher Structures against Differential Cryptanalysis, Linear Cryptanalysis and Impossible Differential Cryptanalysis*. His thesis supervisor was Prof. Sangjin Lee. In September 2002 and February 2005, he started working toward the Ph.D. degree in Engineering of Information Security of Korea University (Korea) and in the research group COSIC (COmputer Security and Industrial Cryptography) at the department of Electrical Engineering (ESAT) of the K.U.Leuven (Belgium), respectively. His Ph.D. supervisors are Prof. Bart Preneel and Prof. Sangjin Lee.