A Related-Key Rectangle Attack on the Full KASUMI

Eli Biham[1], Orr Dunkelman*[1], Nathan Keller[2]

[1]Computer Science Department, Technion.
Haifa 32000, Israel
{biham,orrd}@cs.technion.ac.il
[2]Einstein Institute of Mathematics, Hebrew University.
Jerusalem 91904, Israel
nkeller@math.huji.ac.il

Abstract. KASUMI is an 8-round Feistel block cipher used in the confidentiality and the integrity algorithms of the 3GPP mobile communications. As more and more 3GPP networks are being deployed, more and more users use KASUMI to protect their privacy. Previously known attacks on KASUMI can break up to 6 out of the 8 rounds faster than exhaustive key search, and no attacks on the full KASUMI have been published.

In this paper we apply the recently introduced related-key boomerang and rectangle attacks to KASUMI, resulting in an attack that is faster than exhaustive search against the full cipher. We also present a related-key boomerang distinguisher for 6-round KASUMI using only 768 adaptively chosen plaintexts and ciphertexts encrypted or decrypted under four related keys.

Recently, it was shown that the security of the entire encryption system of the 3GPP networks cannot be proven using only the "ordinary" assumption that the underlying cipher (KASUMI) is a Pseudo-Random Permutation. It was also shown that if we assume that KASUMI is also secure with respect to differential-based related-key attacks then the security of the entire system can be proven. Our results show that theoretically, KASUMI is not secure with respect to differential-based related-key attacks, and thus, the security of the entire encryption system of the 3GPP cannot be proven at this time.

1 Introduction

KASUMI [31] is a 64-bit block cipher used in the confidentiality and the integrity algorithms of the 3GPP mobile communications. KASUMI was developed through the collaborative efforts of the 3GPP organizational partners. It is a slight modification of the known block cipher MISTY1 [27], optimized for implementation in hardware.

* The research presented in this paper was supported by the Clore scholarship programme.

The security of the entire 3GPP mobile network relies on the security of the underlying block cipher KASUMI. Initial examination of the modes of operation used in the 3GPP networks showed that if KASUMI is a Pseudo-Random Permutation (PRP), then the entire network is provably secure [20, 16]. However, it appeared that the proof was incorrect [17]. Moreover, it was shown that assuming only that the underlying cipher is a PRP, the security of the modes of operation cannot be proven [17]. In [18], Iwata and Kohno showed that if KASUMI is a PRP and is also secure with respect to differential-based related-key attacks, then the modes in which KASUMI is used can be proven secure. This result shows that the strength of KASUMI with respect to related-key attacks is crucial to the security of the entire mobile network.

KASUMI accepts 128-bit keys and consists of eight Feistel rounds. Previous results on KASUMI include an impossible differential attack on a 6-round version of the cipher presented by Kühn [25] and a related-key differential attack on a 6-round version of the cipher presented by Blunden and Escott [12]. There are no known attacks applicable to the full 8-round KASUMI.

In this paper we apply the recently introduced related-key boomerang and rectangle attacks to the full 8-round KASUMI and to reduced-round versions of the cipher.

The *boomerang attack* [33] is an adaptive chosen plaintext and ciphertext attack built over differential cryptanalysis [9]. The cipher is treated as a cascade of two sub-ciphers, and a short differential is used in each of these two sub-ciphers. These two differentials are combined in an elegant way to suggest some property of the entire cipher with high probability that can be detected using adaptive chosen plaintext and ciphertext queries.

The boomerang attack was further developed in [21] into a chosen plaintext attack called the *amplified boomerang attack*. The transformation uses birthday paradox techniques to eliminate the adaptive nature of the attack by encrypting large sets of plaintexts. After the encryption of the plaintexts, the attacker searches for quartets of plaintexts that behave as if they were constructed in the boomerang process. The transformation to a chosen plaintext attack (instead of an adaptive chosen plaintexts and ciphertexts attack) has price both in a much larger data complexity, and in a much more complicated algorithm for the identification of the right quartets. After its introduction, the amplified boomerang attack was further developed into the *rectangle attack* [6]. The rectangle attack utilizes a more careful analysis that shows that the probability of a right quartet is significantly higher than suggested by the amplified boomerang attack. Also an optimized algorithm for finding and identifying the right quartets was given in [7]. The boomerang and the rectangle attacks were used to attack several reduced-round versions of block ciphers, including the AES, Serpent, SHACAL-1, COCONUT98 (the full cipher), SC2000, Khufu and FEAL.

Related-key attacks were introduced by Biham [2] in 1993. This technique assumes that the attacker is able to request the encryptions of plaintexts under two related keys: an unknown key and a key (also unknown) that is related to it in some known way. Under this assumption, the attacker uses the relations between the keys and various weaknesses of the cipher to derive information about the two keys. In [2] a related-key attack was applied to a modified variant of DES [28], to LOKI [13] and to Lucifer [29]. In [22] Kelsey et al. combined the related-key technique with differential cryptanalysis [9]. In the related-key differential attack, the attacker requests the encryption of pairs of plaintexts with some chosen difference under the unknown key and under a related key such that the difference between the keys is chosen by the attacker. Related-key differential attacks were used to attack several full/reduced versions of block ciphers, including AES [14], KASUMI [31], and others (see the attacks of [19, 12, 22]).

The related-key boomerang and rectangle attacks were presented by Kim et al. [23, 24] and independently by Biham et al. [8]. These attacks are a combination of the boomerang/rectangle technique with the related-key differential technique. In the attack, the attacker examines quartets of plaintexts encrypted under four differentially related keys. The key differences are used to improve the two differentials used for the boomerang (or the rectangle) distinguisher. Related-key boomerang and rectangle attacks were used to attack reduced versions of AES [14], IDEA [26] and SHACAL-1 [15] and the full COCONUT98 [32].

In this paper we present a key recovery related-key rectangle attack on the entire 8-round version of KASUMI. The attack requires $2^{54.6}$ chosen plaintexts encrypted under four related keys and has time complexity of $2^{76.1}$ encryptions. We also present a related-key boomerang distinguisher of 6-round KASUMI. The distinguisher requires 768 adaptive chosen plaintexts and ciphertexts encrypted under four related keys and has a negligible time complexity. We summarize our results along with previously known results on KASUMI in Table 1.[1]

Our results do not practically compromise the security of the 3GPP mobile networks. However, our results show that KASUMI cannot be considered secure against differential-based related-key attacks. Therefore, the security of the entire mobile network cannot be proven at this stage.

This paper is organized as follows: In Section 2 we give a brief description of the structure of KASUMI. In Section 3 we describe the related-key boomerang and rectangle attacks. In Section 4 we present a related-key rectangle attack on the full KASUMI. Section 5 contains a related-key boomerang distinguisher of 6-round KASUMI. Finally, Section 6 summarizes the paper.

[1] We note that several generic attacks that apply to any block cipher with 64-bit block and 128-bit keys, such as exhaustive key search, key-collision, and time-memory-data tradeoffs, may be used to attack the cipher. For example, a key-collision attack on this cipher has time complexity of 2^{64} encryptions using 2^{64} known plaintexts, each encrypted under a different key [3]. For time-memory-data tradeoff attacks using four different keys as in our attack, the overall time complexity (including preprocessing) is very close to the time complexity of an exhaustive key search. A time-memory-data tradeoff attack using a fixed known plaintext encrypted under a large number of 2^{43} keys can be performed with on-line computation of 2^{84} encryptions and preprocessing of 2^{85} encryptions [11].

Attack	Number of		Complexity		Source
	Rounds	Keys	Data	Time	
Higher-Order Differential	$4^{\dagger}$	1	$2^{10.5}$ CP	$2^{22.11}$	[30]
Related-Key Differential	6	1	$2^{18.6}$ RK-CP	$2^{113.6}$	[12]
Impossible Differential	6	1	2^{55} CP	2^{100}	[25]
Related-Key Boomerang Distinguisher	6	4	768 RK-ACPC	1	Section 5.2
Related-Key Boomerang Key Recovery	6	34	2^{13} RK-ACPC	2^{13}	Section 5.3
Basic Related-Key Rectangle	8	4	2^{53} RK-CP	2^{102}	Section 4.2
Improved Related-Key Rectangle	8	4	$2^{54.6}$ RK-CP	$2^{76.1}$	Section 4.4
Related-Key Boomerang	8	4	$2^{45.2}$ RK-ACPC	$2^{78.7}$	Section 4.4

RK – Related-key, CP – Chosen plaintext, ACPC – Adaptive chosen plaintext and ciphertext
Time complexity is measured in encryption units.
† – this attack is on a version of the cipher without the FL functions.

Table 1. Summary of the Attacks on KASUMI

2 The KASUMI cipher

KASUMI [31] is a 64-bit block cipher that has a key size of 128 bits. KASUMI was designed as a modification of MISTY1 [27], optimized for implementation in hardware. Therefore, most of the components of KASUMI are similar to the respective components of MISTY1.

KASUMI has a recursive structure. Each of its eight Feistel rounds is composed of an FO function which is a 3-round 32-bit Feistel construction, and of an FL function that mixes a 32-bit subkey with the data. The order of the two functions changes each round (in odd rounds the FL function is first, and in the even rounds the FO function is first).

The FO function also has a recursive structure. Each of the three rounds of the FO functions consists of a key mixing stage and of an application of the FI function, yet another three-round Feistel construction. The FI functions use two non-linear S-boxes $S7$ and $S9$ (where $S7$ is a 7-bit to 7-bit permutation and $S9$ is a 9-bit to 9-bit permutation) and accept an additional 16-bit subkey, which is mixed with the data. In total, a 96-bit subkey enters FO in each round — 48 subkey bits used in the FI functions and 48 subkey bits in the key mixing stages.

The FL function accepts 32-bit input and two 16-bit subkey words. One subkey word affects the data using the OR operation, while the second one affects the data using the AND operation. We outline the structure of KASUMI and its parts in Figure 1.

One of the major differences between KASUMI and MISTY1 is in the key schedule. In KASUMI, the subkeys are derived from the key in a linear way: The 128-bit key K is divided into eight 16-bit words: $K_1, K_2, \ldots, K_8$. Each K_i is used to compute $K_i' = K_i \oplus C_i$, where the C_i's are known and fixed constants. The constants C_i are interleaved with the key bits in order to avoid weak-key classes

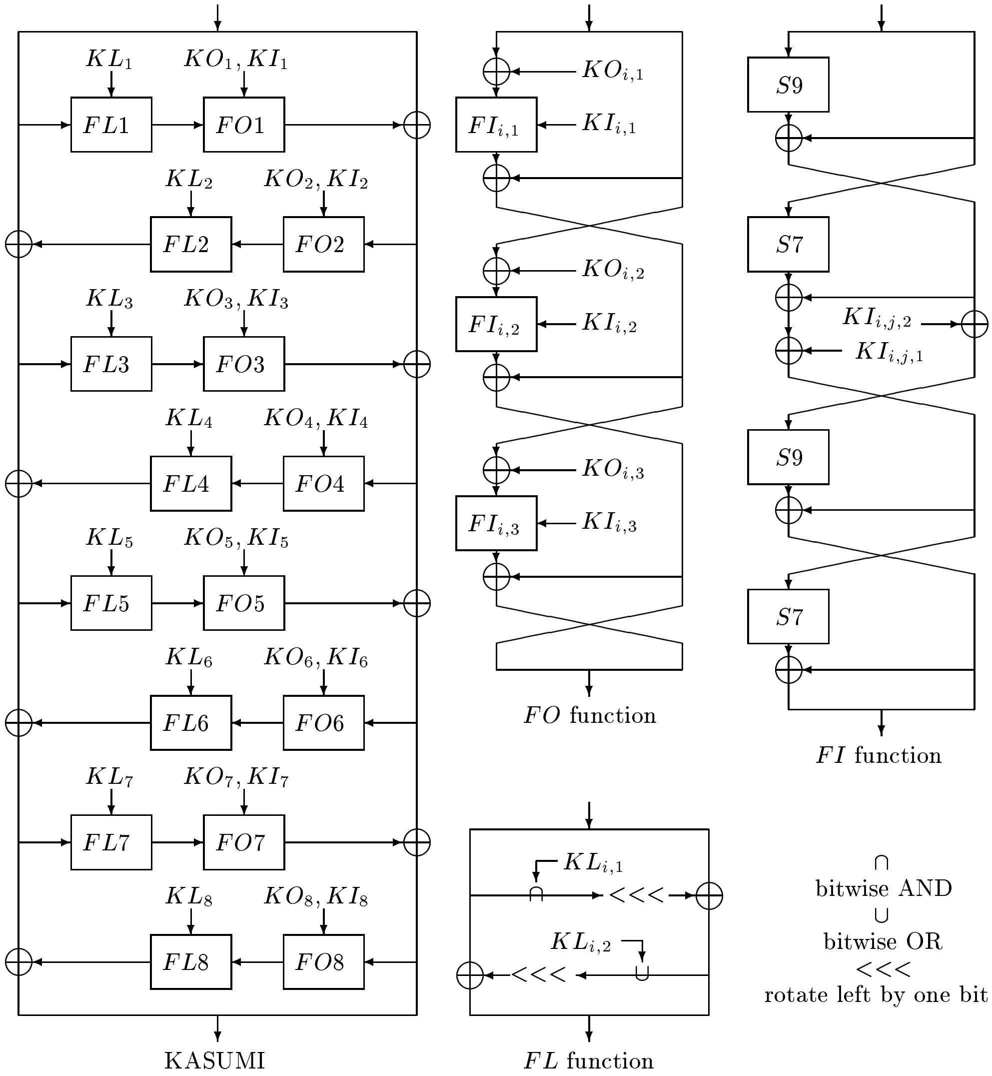

Fig. 1. Outline of KASUMI

based on fixing key bits to be zero. Such weak keys were found in IDEA [26] (see for example [10]) and in other ciphers as well.

In each round, eight words are used as the round subkey (up to some in-word rotations). Therefore, the 128-bit subkey of each round is a linearly dependent of the secret key in a very simple way. We give the exact key schedule of KASUMI in Table 2 and list the values of the constants in Table 3.

Round	$KL_{i,1}$	$KL_{i,2}$	$KO_{i,1}$	$KO_{i,2}$	$KO_{i,3}$	$KI_{i,1}$	$KI_{i,2}$	$KI_{i,3}$
1	$K_1 \lll 1$	K'_3	$K_2 \lll 5$	$K_6 \lll 8$	$K_7 \lll 13$	K'_5	K'_4	K'_8
2	$K_2 \lll 1$	K'_4	$K_3 \lll 5$	$K_7 \lll 8$	$K_8 \lll 13$	K'_6	K'_5	K'_1
3	$K_3 \lll 1$	K'_5	$K_4 \lll 5$	$K_8 \lll 8$	$K_1 \lll 13$	K'_7	K'_6	K'_2
4	$K_4 \lll 1$	K'_6	$K_5 \lll 5$	$K_1 \lll 8$	$K_2 \lll 13$	K'_8	K'_7	K'_3
5	$K_5 \lll 1$	K'_7	$K_6 \lll 5$	$K_2 \lll 8$	$K_3 \lll 13$	K'_1	K'_8	K'_4
6	$K_6 \lll 1$	K'_8	$K_7 \lll 5$	$K_3 \lll 8$	$K_4 \lll 13$	K'_2	K'_1	K'_5
7	$K_7 \lll 1$	K'_1	$K_8 \lll 5$	$K_4 \lll 8$	$K_5 \lll 13$	K'_3	K'_2	K'_6
8	$K_8 \lll 1$	K'_2	$K_1 \lll 5$	$K_5 \lll 8$	$K_6 \lll 13$	K'_4	K'_3	K'_7

$X \lll i$ — X rotated to the left by i bits

Table 2. KASUMI's Key Schedule Algorithm

Round	1	2	3	4	5	6	7	8
Constant	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
Value	0123_x	4567_x	$89AB_x$	$CDEF_x$	$FEDC_x$	$BA98_x$	7654_x	3210_x

Table 3. KASUMI's Key Schedule Constants

3 Related-Key Boomerang and Related-Key Rectangle Attacks

In this section we describe the related-key boomerang and related-key rectangle attacks. First, we outline the boomerang/rectangle attacks and the related-key differential attacks separately. Then, we describe the combination that forms the related-key boomerang and related-key rectangle attacks.

3.1 The Boomerang and the Rectangle Attacks

The main idea behind the boomerang attack [33] is to use two short differentials with high probabilities instead of one long differential with a low probability. We assume that a block cipher $E : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ can be described as a cascade $E = E_1 \circ E_0$, such that for E_0 there exists a differential $\alpha \to \beta$ with probability p, and for E_1 there exists a differential $\gamma \to \delta$ with probability q. We note that the second differential $\gamma \to \delta$ for E_1 is actually used in the backward direction, i.e., decryption, but as we are dealing with differentials (and not truncated differentials), then this does not change the probability of the differential.

The distinguisher is based on the following boomerang process:

- Ask for the encryption of a pair of plaintexts (P_1, P_2) such that $P_1 \oplus P_2 = \alpha$ and denote the corresponding ciphertexts by (C_1, C_2).
- Calculate $C_3 = C_1 \oplus \delta$ and $C_4 = C_2 \oplus \delta$, and ask for the decryption of the pair (C_3, C_4). Denote the corresponding plaintexts by (P_3, P_4).
- Check whether $P_3 \oplus P_4 = \alpha$.

The boomerang attack uses the first characteristic ($\alpha \to \beta$) for E_0 with respect to the pairs (P_1, P_2) and (P_3, P_4), and uses the second characteristic ($\gamma \to \delta$) for E_1 with respect to the pairs (C_1, C_3) and (C_2, C_4).

For a random permutation the probability that the last condition is satisfied is 2^{-n}. For E, the probability that the pair (P_1, P_2) is a right pair with respect to the first differential ($\alpha \to \beta$) is p. The probability that both pairs (C_1, C_3) and (C_2, C_4) are right pairs with respect to the second differential is q^2. If all these are right pairs, then $E_1^{-1}(C_3) \oplus E_1^{-1}(C_4) = \beta = E_0(P_3) \oplus E_0(P_4)$. Thus, with probability p, $P_3 \oplus P_4 = \alpha$. The total probability of this quartet of plaintexts and ciphertexts to satisfy the boomerang conditions is $(pq)^2$.

The attack can be mounted for all possible β's and γ's simultaneously (as long as $\beta \neq \gamma$). Therefore, a right quartet for E is encountered with probability no less than $(\hat{p}\hat{q})^2$, where:

$$\hat{p} = \sqrt{\sum_{\beta} \Pr{}^2[\alpha \to \beta]}, \qquad \text{and} \qquad \hat{q} = \sqrt{\sum_{\gamma} \Pr{}^2[\gamma \to \delta]}.$$

The complete analysis is given in [33, 6, 7].

As the boomerang attack requires adaptive chosen plaintexts and ciphertexts, many of the techniques that were developed for using distinguishers in key recovery attacks cannot be applied. This led to the introduction of chosen plaintext variants of the boomerang attack called the *amplified boomerang attack* [21] and the *rectangle attack* [6]. The transformation of the boomerang attack into a chosen plaintext attack is quite standard, as it can be achieved by birthday-paradox arguments. The key idea behind the transformation is to encrypt many plaintext pairs with input difference α, and to look for quartets that conform to the requirements of the boomerang process.

Given the same decomposition of E as before, and the same basic differentials, the analysis in [6] shows that out of N plaintext pairs, the number of right quartets is expected to be $N^2 2^{-n} \hat{p}^2 \hat{q}^2$. We note, that the main reduction in the probability follows from the fact that unlike the boomerang attack, in the rectangle attack the event $E_0(P_1) \oplus E_0(P_3) = \gamma$ occurs with probability 2^{-n}.

3.2 Related-Key Differentials

Related-key differentials [22] were used for cryptanalysis several times in the past. Recall, that a regular differential deals with some plaintext difference ΔP and a ciphertext difference ΔC such that

$$\Pr{}_{P,K}[E_K(P) \oplus E_K(P \oplus \Delta P) = \Delta C]$$

is high enough (or zero [5]).

A related-key differential is a triplet of a plaintext difference ΔP, a ciphertext difference ΔC, and a key difference ΔK, such that

$$\Pr{}_{P,K}[E_K(P) \oplus E_{K \oplus \Delta K}(P \oplus \Delta P) = \Delta C]$$

is useful (high enough or zero).

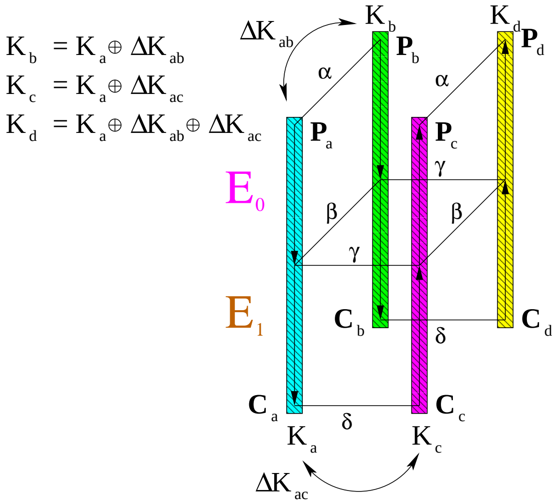

Fig. 2. A Related-Key Boomerang Quartet

3.3 Related-Key Boomerang Attacks

Let us assume that we have a related-key differential $\alpha \to \beta$ of E_0 under a key difference ΔK_{ab} with probability p. Assume also that we have another related-key differential $\gamma \to \delta$ for E_1 under a key difference ΔK_{ac} with probability q.

The related-key boomerang process involves four different unknown (but related) keys — K_a, $K_b = K_a \oplus \Delta K_{ab}$, $K_c = K_a \oplus \Delta K_{ac}$, and $K_d = K_a \oplus \Delta K_{ab} \oplus \Delta K_{ac}$. The attack is performed by the following algorithm:

- Choose a plaintext P_a at random, and compute $P_b = P_a \oplus \alpha$.
- Ask for the ciphertexts $C_a = E_{K_a}(P_a)$ and $C_b = E_{K_b}(P_b)$.
- Compute $C_c = C_a \oplus \delta$ and $C_d = C_b \oplus \delta$.
- Ask for the plaintexts $P_c = E^{-1}_{K_c}(C_c)$ and $P_d = E^{-1}_{K_d}(C_d)$.
- Check whether $P_c \oplus P_d = \alpha$.

See Figure 2 for an outline of such a quartet.

It is easy to see that for a random permutation, the probability that the last condition is satisfied is 2^{-n}. For E the probability that this condition is satisfied is p^2q^2. Hence, the related-key boomerang attack can be used for distinguishing and key recovery attacks for this cipher.

The attack can use many differentials for E_0 and E_1 simultaneously (just like in a regular boomerang attack), as long as all related-key differentials used in E_0 have the same key difference ΔK_{ab} and the same input difference α, and that all related-key differentials used in E_1 have the same key difference ΔK_{ac} and the same output difference δ. Thus, the probability of a quartet to be a right one is $\hat{p}^2\hat{q}^2$.

In the case of KASUMI, the key schedule algorithm is linear. Therefore, given a key difference, all subkey differences are known, and can be easily used in the related-key model.

3.4 Related-Key Rectangle Attack

The transformation of the related-key boomerang attack into a related-key rectangle attack is similar to the transformation of the boomerang attack to the rectangle attack. The related-key rectangle distinguisher is as follows:

- Choose N plaintext pairs $(P_a, P_b = P_a \oplus \alpha)$ at random and ask for the encryption of P_a under K_a and of P_b under K_b. Denote the set of these pairs by S.
- Choose N plaintext pairs $(P_c, P_d = P_c \oplus \alpha)$ at random and ask for the encryption of P_c under K_c and P_d under K_d. Denote the set of these pairs by T.
- Search a pair of plaintexts $(P_a, P_b) \in S$ and a pair of plaintexts $(P_c, P_d) \in T$, and their corresponding ciphertexts (C_a, C_b) and (C_c, C_d), respectively, satisfying:
  - $P_a \oplus P_b = P_c \oplus P_d = \alpha$
  - $C_a \oplus C_c = C_b \oplus C_d = \delta$

The analysis of the related-key rectangle attack is similar to the one of the rectangle attack (with the same modifications that were presented at the related-key boomerang attack). Starting with N plaintext pairs in S and N plaintext pairs in T, we expect to find $N^2 2^{-n} (\hat{p}\hat{q})^2$ right quartets. For a random permutation the number of "right quartets" is about $N^2 2^{-2n}$, so as long as $\hat{p}\hat{q} > 2^{-n/2}$ we can use the rectangle attack to distinguish between a random permutation and the attacked cipher. This distinguisher can be later used for a key recovery attack.

4 Related-Key Rectangle Attack on KASUMI

In this section we devise a related-key rectangle attack on the entire KASUMI. We start with a short description of the related-key differentials used in this attack, then describe a basic attack without full optimization, and its analysis. Finally, we describe the optimizations that reduce the complexities to our final results.

4.1 Related-Key Differentials of KASUMI

As mentioned earlier, KASUMI's round function is composed of two main functions: the FO function and the FL function. A non-zero input difference to the FO function can become almost any output difference, with approximately the same probability. However, non-zero differences to the FL-function propagate with much higher probabilities.

For the rectangle attack we use two related-key differentials. The first related-key differential is for rounds 1–4, while the second is used in rounds 5–7.

4.1.1 A 4-Round Related-Key Differential for Rounds 1–4

This 4-round related-key differential is an extension by one round of the related-key differential presented in [12]. The key difference is $\Delta K_{ab} = (0,0,1,0,0,0,0,0)$, i.e., only the third key word has a non-zero difference $K_3' = 0001_x$. The plaintext difference of the differential is $\alpha = (0_x, 0020\ 0000_x)$. It was shown in [12] that with probability 1/4, the difference after three rounds is equal to α as well. The input difference of the FO function in the fourth round is non-zero ($0020\ 0000_x$). The key difference of the fourth round is introduced only at the end of the FO function (precisely, in $FI_{4,3}$). Hence, the non-zero difference propagates through all the parts of FO, and the output difference of the round function is distributed almost uniformly. Therefore, we shall use the differentials $\alpha = (0_x, 0020\ 0000_x) \to (y, 0020\ 0000_x)$ for all the possible values of y. In the worst case, all the y values are equiprobable. Thus, when using all the 2^{32} possible values, each of them is expected to occur with probability 2^{-32}. Hence, each differential of the form $\alpha = (0_x, 0020\ 0000_x) \to (y, 0020\ 0000_x)$ has probability 2^{-34}. The effective probability of the differentials when using all these differentials simultaneously is $\hat{p} = \sqrt{2^{32} \cdot (2^{-34})^2} = \sqrt{2^{-36}} = 2^{-18}$. If the y values are not equiprobable, then the value of $\hat{p}$ is higher.

As observed in [12], the attacker can select two bits of the plaintext in order to double the probability of the differential: The attacker assigns one bit of the plaintext to be one (thus fixing one bit of the output of the OR operation in $FL1$) and one bit of the plaintext to be zero (thus fixing one bit of the output of the AND operation in $FL1$). More precisely, let $P = (P_{LL}, P_{LR}, P_{RL}, P_{RR})$, where P_{LL} is the 16 plaintext bits that enter the AND operation of the FL function in the first round, and P_{LR} are the remaining bits of the left half of the plaintext. The attacker sets the least significant bit of P_{LL} and the second least significant bit of P_{LR} to $P_{LL}^0 = 0$ and $P_{LR}^1 = 1$, where the superscript $x \in \{0,1\}$ denotes the x'th bit of that quarter of the plaintext. This selection ensures that the characteristic holds with probability 1 in the first round (instead of 1/2), despite of the key difference. Therefore, the probability of the differential $\alpha = (0_x, 0020\ 0000_x) \to (y, 0020\ 0000_x)$ is increased from 2^{-34} to 2^{-33}, and the effective probability of the first part of the rectangle is increased to $\hat{p} = \sqrt{2^{32} \cdot (2^{-33})^2} = \sqrt{2^{-34}} = 2^{-17}$.

It is possible to rotate all the words of the key difference ΔK_{ab} and the characteristic by the same number of bits, without changing the probability of the characteristic. Hence, the above characteristic can be replaced by 15 other characteristics.

4.1.2 A 3-Round Related-Key Differential for Rounds 5–7

The 3-round related-key differential used in rounds 5–7 is the 3-round differential of [12] shifted by four rounds. The key difference is $\Delta K_{ac} = (0,0,0,0,0,0,1,0)$. Again, it is possible to rotate the difference in K_7' and the corresponding values in the characteristic, to obtain a new characteristic with the same probability.

The differential is $\gamma = (0_x, 0020\ 0000_x) \to (0_x, 0020\ 0000_x) = \delta$ with probability $q = \hat{q} = 1/4$.

4.2 The Basic Related-Key Rectangle Attack on KASUMI

The attack on KASUMI treats the cipher as a cascade of three parts: E_0 consists of the first four rounds, E_1 consists of rounds 5–7, and E_f the round after the distinguisher (round 8), which is used for analysis. Let K_a, $K_b = K_a \oplus \Delta K_{ab}$, $K_c = K_a \oplus \Delta K_{ac}$, and $K_d = K_c \oplus \Delta K_{ab}$ be the unknown related keys that we want to retrieve.

For E_0 we use the 4-round differential with $\hat{p} = 2^{-17}$ presented earlier, whose key difference is $\Delta K_{ab} = (0, 0, 1, 0, 0, 0, 0, 0)$ and whose input difference is $\alpha = (0_x, 0020\ 0000_x)$. For E_1 we use the 3-round differential with $\hat{q} = 2^{-2}$ presented earlier, whose key difference is $\Delta K_{ac} = (0, 0, 0, 0, 0, 1, 0, 0)$ and whose output difference is $\delta = (0_x, 0020\ 0000_x)$.

If we encrypt $N = 2^{51}$ pairs of plaintexts under K_a and K_b, and the same number of pairs under K_c and K_d, we expect to find $N^2 = 2^{102}$ quartets, of which about $N^2 \cdot 2^{-64} \cdot 2^{-34} \cdot 2^{-4} = 2^{102} \cdot 2^{-102} = 1$ are right rectangle quartets.

In the attack we identify the right quartets out of all possible quartets, and then analyze them to retrieve the subkey of round 8. This analysis is performed in the following way:

1. **Data Collection Phase:**
   (a) Ask for the encryption of 2^{51} pairs of plaintexts (P_a, P_b), where $P_b = P_a \oplus \alpha$, $P^0_{a_{LL}} = 0$, and $P^1_{a_{LR}} = 1$, and where P_a is encrypted under K_a and P_b is encrypted under K_b. Insert each pair into a hash table indexed by the 64-bit value of $(C_{a_{RL}}, C_{a_{RR}}, C_{b_{RL}}, C_{b_{RR}})$.
   (b) Ask for the encryption of 2^{51} pairs of plaintexts (P_c, P_d), where $P_d = P_c \oplus \alpha$, $P^0_{c_{LL}} = 0$, and $P^1_{c_{LR}} = 1$, and where P_c is encrypted under K_c and P_d is encrypted under K_d. For each pair, access the hash table in the entry corresponding to the value $(C_{c_{RL}} \oplus 0020_x, C_{c_{RR}}, C_{d_{RL}} \oplus 0020_x, C_{d_{RR}})$. For each pair (P_a, P_b) found in this entry, apply Step 2 on the quartet (P_a, P_b, P_c, P_d).

The $(2^{51})^2$ possible quartets are filtered according to a condition on 64 bits on the difference of the ciphertexts, leading to about 2^{38} quartets that enter Step 2. In the following step, we treat all remaining quartets as right quartets. The analysis of a quartet is done by guessing 32 bits of the key $(KO_{8,1}, KI_{8,1})$, and trying to deduce $KL_{8,2}$. In most cases there is a contradiction, e.g., one of the pairs suggests something which is impossible, or the two pairs disagree on some key bit.

2. **Analyzing Quartets:**
   (a) For each quartet (C_a, C_b, C_c, C_d), guess the 32-bit value of $KO_{8,1}$ and $KI_{8,1}$. Assume that this is a right quartet. For the two pairs (C_a, C_c) and (C_b, C_d) use the value of the guessed key to compute the input and output differences of the OR operation in the last round of both pairs. For each bit of this 16-bit OR operation of $FL8$, the possible values of the corresponding bit of $KL_{8,2}$ are given in Table 4. On average $(8/16)^{16} = 2^{-16}$ values of $KL_{8,2}$ are suggested by each quartet and guess of $KO_{8,1}$ and $KI_{8,1}$.

OR — $KL_{8,2}$				
	(X'_2, Y'_2)			
(X'_1, Y'_1)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	{0,1}	—	1	0
(0,1)	—	—	—	—
(1,0)	1	—	1	—
(1,1)	0	—	—	0

AND — $KL_{8,1}$				
	(X'_2, Y'_2)			
(X'_1, Y'_1)	(0,0)	(0,1)	(1,0)	(1,1)
(0,0)	{0,1}	—	0	1
(0,1)	—	—	—	—
(1,0)	0	—	0	—
(1,1)	1	—	—	1

* The two bits of the differences are denotes by (input difference, output difference): (X'_1, Y'_1) for one pair and (X'_2, Y'_2) for the other.

Table 4. Possible Values of $KL_{8,2}$ and $KL_{8,1}$

(b) For each quartet and values of $KO_{8,1}, KI_{8,1}$ and $KL_{8,2}$ suggested in Step 2(a), guess the 32-bit value of $KO_{8,3}$ and $KI_{8,3}$, and use this information to compute the input and output differences of the AND operation in both pairs. For each bit of the 16-bit AND operation of $FL8$, the possible values of the corresponding bit of $KL_{8,1}$ are given in Table 4. On average $(8/16)^{16} = 2^{-16}$ values of $KL_{8,1}$ are suggested by each quartet and guess of $KO_{8,1}, KI_{8,1}, KO_{8,3}$, and $KI_{8,3}$ and the computed value of $KL_{8,2}$.

3. **Finding the Right Key:** For each quartet and value of $KO_{8,1}, KI_{8,1}, KO_{8,3}, KI_{8,3}$ and the value of $KL_{8,1}$ and $KL_{8,2}$ suggested in Step 2, guess the remaining 32 bits of the key, and perform a trial encryption.

4.3 Analysis of the Attack

We first analyze Step 2(a), and show that given the input and output differences of the OR operation in the two pairs of the quartet, the expected number of suggestions for the key $KL_{8,2}$ is 2^{-16}. This means that the $2^{38} \cdot 2^{32} = 2^{70}$ (quartet, subkey guesses) tuples suggest $2^{70} \cdot 2^{-16} = 2^{54}$ subkey guesses for 48-bit value.

Let us examine a difference in some bit j. There are four combinations of input difference and output difference for this bit for each pair. Table 4 lists the key bits that the two pairs suggest for the respective key bit.

There are nine entries that contain no value. For example, a difference 0 may never cause a difference 1 by any function. Another possible contradiction happens when one pair suggests that the key bit is 0, while the second pair suggests that the key bit is 1. The total number of possible key bits is 8 out of 16 entries. Thus, on average there is 1/2 a possibility for each bit. In total, for the 16 bits there are $(1/2)^{16} = 2^{-16}$ possibilities on average. A similar analysis can be applied to Step 2(b).

As noted earlier, the expected number of (quartet, subkey guesses) tuples that enter Step 2(b) is 2^{54}. For each of these tuples, we guess 32 additional bits, resulting in $2^{54} \cdot 2^{32} = 2^{86}$ (quartet, subkey guesses) tuples. As step 2(b) is similar to Step 2(a), then after its execution, the expected number of (quartet, subkey guesses) tuples is $2^{86} \cdot 2^{-16} = 2^{70}$, while the guessed subkey has 96 bits in total.

Step 2(a) can be implemented using only a few logical operations. The test whether a pair suggests a contradiction (a zero difference in the input with corresponding non-zero difference in the output) can be performed as follows: Let X' be the word of input differences and let Y' be the word of output differences. Compute $Z = \overline{X'} \wedge Y'$, where $\overline{X'}$ is the bitwise complement of X'. If Z is non-zero then there is some bit in X' which is zero, while the corresponding bit in Y' is 1. Thus, we can check using two logical operation whether one of the pairs suggests a contradiction of this kind.

We can also find which bits of the key a key suggests. For the OR operation, the bits that a pair suggests is the bits for which X' has 1, and the value of $KL_{8,2}$ in these bits is the same as in $\overline{Y'}$. To check whether the two pairs suggest contradicting values for the key, it suffices to check whether $(X'_1 \wedge X'_2) \wedge (Y'_1 \oplus Y'_2) \neq 0$. A similar method can be used on Step 2(b) (after updating the relevant expression to take into consideration the AND operation). Further optimizations of the generation of the list of possible values of $KL_{8,2}$ and $KL_{8,1}$ can be made using table lookups.

Step 3 goes over all 2^{70} suggestions for the 96 bits of the key, and tries to complete the remaining 32 bits by an exhaustive search. This can easily be performed due to the linear key schedule of KASUMI. The time complexity of this step is 2^{102} trial encryptions.

As the complexity of Step 3 is dominant, the total complexity of this attack is 2^{102} trial encryptions. This complexity is further reduced in the next subsection.

4.4 Improvements of the Attack

Step 3 can be improved by using counting techniques. In case we encrypt three times the data ($2^{52.6}$ plaintexts encrypted under four different keys), we expect to have nine right quartets. Instead of completing the missing key bits by an exhaustive key search, we count how many (quartet, subkey guesses) tuples suggest each value of the 96 bits of $KO_{8,1}$, $KI_{8,1}$, $KO_{8,3}$, $KI_{8,3}$, $KL_{8,1}$ and $KL_{8,2}$. Only few possible wrong key values are expected to get more than five suggestions. On the other hand, the right key has probability 88.4% to have at least this number of suggestions. Therefore, we identify which 96-bit values have more than five suggestions, and exhaustively search over the remaining bits of these cases. The time complexity of this attack is dominated by Step 2(b). The data complexity of the attack is $2^{54.6}$ related-key chosen plaintexts and the time complexity of the attack is equivalent to $2^{86.2}$ full KASUMI encryptions.

Another improvement of the attack is based on the observation that Step 2(b) can be implemented in two substeps. In the first one, we guess $KO_{8,3}$ and the 9 bits of $KI_{8,3,2}$, and find the value of only 9 bits of $KL_{8,1}$. Hence, we generate $9 \cdot 2^{54} \cdot 2^{25} = 2^{82.2}$ (quartet, subkey guesses) where the subkey guess is of 73 bits. As this improvement first deals only with 9 bits of $KL_{8,1}$, the expected number of remaining (quartet, subkey guesses) values is $2^{73.2}$. Then, we perform the second substep on the 7 remaining bits of $KI_{8,3,1}$ and of $KL_{8,1}$. The time complexity of the attack is now dominated by the first substep of Step 2(b), whose complexity is equivalent to about $2^{79.2}$ KASUMI encryptions.

Our last improvement uses the fact that Step 2(b) (and even its first substep) partially depends on Step 2(a). After Step 2(a) there are 2^{54} tuples of the form (quartet, subkey guesses), where the subkey guess is of 48 bits. However, Step 2(b) uses only 32 bits of the guessed subkey, i.e., the value of $KO_{8,1}$ and $KI_{8,1}$. As mentioned earlier, a given quartet suggests about 2^{16} values for the 48 bits of $KO_{8,1}, KI_{8,1}, KL_{8,2}$. However, it suggests about $2^{12.9}$ values for 32 bits of $KO_{8,1}, KI_{8,1}$.

This observation is used to reduce the complexity of the attack: The purpose of Step 2(a) is now to find the list of about $2^{12.9}$ values for $KO_{8,1}, KI_{8,1}$ that a quartet suggests, and then Step 2(b) finds the list of about $2^{12.9}$ values for $KO_{8,3}, KI_{8,3}$. Only then, in Step 3, we take into consideration the possible values of $KL_{8,1}$ and $KL_{8,2}$. This reduces the time complexity of the attack to $2^{76.1}$ KASUMI encryptions.

The attack can also be transformed into a related-key boomerang attack that requires $2^{43.2}$ adaptive chosen plaintexts and ciphertexts (encrypted under four different keys). The attack is performed starting at the decryption direction, and thus it is a chosen ciphertext attack with adaptively chosen plaintexts. The time complexity of this related-key boomerang attack is $2^{78.1}$ encryptions.

5 The Related-Key Boomerang Attack on 6-Round KASUMI

In this section we present a related-key boomerang attack on 6-round KASUMI. The attack is on the first six rounds (rounds 1–6). It finds 16 bits of the key using only 768 adaptive chosen plaintexts and ciphertexts.

5.1 Another 3-Round Differential of KASUMI

In this subsection we present four related-key conditional characteristics [1] for rounds 4–6 of KASUMI. We describe the conditional characteristics in the backward direction as this is the direction in which we use them. These characteristics can be easily adapted to hold for any three consecutive rounds starting with an even round, either in the forward or in the backward direction.

The key difference of all these conditional characteristics is $\Delta K_{ac} = (0, 0, 0, 0, 0, 1, 0, 0)$. Unlike regular characteristics, conditional characteristics depend on the value of some key bit. The four conditional characteristics we use depend on the same key bit. Two of them assume that the value of this key bit is 0, while the two other assume that the value is 1. Let $\delta_0 = (0020\ 0000_x, 0_x)$, $\delta_1 = (0020\ 0040_x, 0_x)$, and $\delta' = (0001\ 0000_x, 0_x)$. The two conditional characteristics that depend on the value zero are $\delta_0 \to \delta_0$ and $\delta_0 \oplus \delta' \to \delta_0$. The two conditional characteristics that depend on the value one are $\delta_1 \to \delta_1$ and $\delta_1 \oplus \delta' \to \delta_1$ (the index of the subscript of δ denotes the value of the key bit). All these conditional characteristics have probability 1/4.

Given a pair with a ciphertext difference of the conditional characteristic, then during the decryption the zero input difference is preserved in round 6 by the $FO6$, and with probability 1/2 it is also the output difference of $FL6$ (there is a subkey difference in one bit that is canceled with probability 1/2). In round 5, we hope to achieve a difference of $0020\ 0000_x$ after $FL5$, which is then canceled with the key difference in $KO_{5,1}$. This is where the conditional property of the characteristics is used. In order to achieve the desired output difference of $FL5$, the conditional characteristic depends on the value of the key bit that is ANDed in $FL5$. There is an active bit in the data, and if the value of the key bit is 1, then this difference is preserved. Otherwise, if the value is 0, then the AND operation has a zero output difference. Thus, for a given value of this key bit, exactly two out of the four characteristics yield a difference $0020\ 0000_x$ after $FL5$ (this part of the conditional characteristic has probability 1), whereas for the other two characteristics this difference is impossible. Therefore, in our attack we use all four characteristics in parallel, and know that two of them pass round 5 with a zero output difference with probability 1.

In round 4, the zero difference is preserved by the $FO4$ function. Again, it has probability 1/2 to be preserved also by $FL4$, and probability 1/2 of not being preserved. Thus, the input difference of the characteristic is either the output difference (δ_1 or δ_2), or the output difference XORed with δ'.

Hence, either each of the first two conditional characteristics have probability 1/4, or the other two have probability 1/4. For each such case the effective probability based on the two characteristics is $\hat{q} = \sqrt{(1/4)^2 + (1/4)^2} = 1/\sqrt{8}$. The successful conditional characteristics are determined by the value of the fifth bit of K_5 (i.e., K_5^4).

We note that all these conditional characteristics can be rotated along with the key difference, to produce 15 similar sets of characteristics with the same effective probability.

5.2 A Related-Key Boomerang Distinguisher on 6-Round KASUMI

In this subsection we present a related-key boomerang distinguisher of 6-round KASUMI. The distinguisher is mounted on rounds 1–6 of KASUMI, but it can be easily adapted to rounds 2–7 or to rounds 3–8 as well.

Denote by E a reduced version of KASUMI consisting of the first six rounds of the cipher. We describe E as a cascade $E = E_1 \circ E_0$, where E_0 corresponds to rounds 1–3 and E_1 corresponds to rounds 4–6. The attack exploits the characteristic $\alpha = (0_x, 0020\ 0000_x) \rightarrow (0_x, 0020\ 0000_x)$ of E_0 with probability 1/4, as well as the four characteristics $\delta_0 \rightarrow \delta_0$, $\delta_0 \oplus \delta' \rightarrow \delta_0$, $\delta_1 \rightarrow \delta_1$, and $\delta_1 \oplus \delta' \rightarrow \delta_1$ of E_1 with probability 1/4. The key difference used in E_0 is $\Delta K_{ab} = (0,0,1,0,0,0,0,0)$, and the key difference of all the characteristics of E_1 is $\Delta K_{ac} = (0,0,0,0,0,1,0,0)$.

The attack essentially performs two standard related-key boomerang distinguishers, one for each possible value of the key bit K_5^4. A small improvement that we use, is to save some of the data by reusing some of the plaintexts generated in the attack. The attack algorithm is as follows:

1. Choose M pairs of plaintexts $(P_{a,i}, P_{b,i})$ (for $1 \leq i \leq M$) such that $P_{a,i} \oplus P_{b,i} = \alpha$. Ask for the encryption of the pairs such that in each pair, $P_{a,i}$ is encrypted under K_a and $P_{b,i}$ is encrypted under the related-key $K_b = K_a \oplus \Delta K_{ab}$. Denote the corresponding ciphertexts by $(C_{a,i}, C_{b,i})$.
2. For $1 \leq i \leq M$, calculate $C_{c,i} = C_{a,i} \oplus \delta_0$ and $C_{d,i} = C_{b,i} \oplus \delta_0$. Ask for the decryption of the pairs $(C_{c,i}, C_{d,i})$ such that in each pair, $C_{c,i}$ is decrypted under $K_c = K_a \oplus \Delta K_{ac}$ and $C_{d,i}$ is decrypted under $K_d = K_a \oplus \Delta K_{ab} \oplus \Delta K_{ac}$. Denote the corresponding plaintexts by $(P_{c,i}, P_{d,i})$.
3. For $1 \leq i \leq M$, calculate $C_{e,i} = C_{a,i} \oplus \delta_1$ and $C_{f,i} = C_{b,i} \oplus \delta_1$. Ask for the decryption of the pairs $(C_{e,i}, C_{f,i})$ such that in each pair $C_{e,i}$ is decrypted under K_c and $C_{f,i}$ is decrypted under K_d. Denote the corresponding plaintexts by $(P_{e,i}, P_{f,i})$.
4. Check whether $P_{c,i} \oplus P_{d,i} = \alpha$ and count the number of such occurrences.
5. Check whether $P_{e,i} \oplus P_{f,i} = \alpha$ and count the number of such occurrences.
6. If one of the two counters from Steps 4 and 5 is greater than zero, then output "6-Round KASUMI". Otherwise, output "Not 6-Round KASUMI".

The total probability of the boomerang process of this distinguisher is $(1/4)^2 \cdot (1/\sqrt{8})^2 = 1/128$, either for quartets counted in Step 4 or for quartets counted in Step 5. Therefore, for $M = 256$ we expect to find two right quartets in Step 4 or Step 5 (either for the quartets $(P_{a,i}, P_{b,i}, P_{c,i}, P_{d,i})$ or for the quartets $(P_{a,i}, P_{b,i}, P_{e,i}, P_{f,i})$). Filtering of these pairs is expected to be very effective as for a random permutation the probability of the event $P_{c,i} \oplus P_{d,i} = \alpha$ (or the event $P_{e,i} \oplus P_{f,i} = \alpha$) is 2^{-64}.

The boomerang distinguisher can be improved using the following observation: Just like in the rectangle attack, by fixing two plaintext bits ($P^0_{a_{LL}} = 0, P^1_{a_{LR}} = 1$), the probability of the first characteristic in the encryption direction is $1/2$ (instead of $1/4$)[2]. Therefore, if we choose all the $(P_{a,i}, P_{b,i})$ according to this additional requirement, the probability of the characteristic in rounds 1–3 in the forward direction doubles.

The overall probability of this boomerang process in this case is doubled to $1/64$. Thus, $M = 128$ suffices for a success rate of about 86%. Hence, our distinguisher requires a total of $3 \cdot 128 \cdot 2 = 768$ adaptively chosen plaintexts and ciphertexts such that 256 chosen plaintexts are encrypted and 512 adaptively chosen ciphertexts are decrypted. The time complexity of the attack is negligible.

5.3 Related-Key Boomerang Key Recovery Attack on 6-Round KASUMI

We note that the boomerang distinguisher can be also used for a key recovery attack. As mentioned earlier, the set of characteristics (of E_1) for which the attack succeeds depends on the value of a single key bit of K_5. Thus, the value of this key bit can be detected by observing which one of the sets of characteristics

[2] The actual probability is slightly higher, i.e., 5/8, and the probability of the first characteristic in the decryption direction is 5/16.

of E_1 is successful. Similar attacks can be mounted by taking other single bits of K_6 to have key difference in E_1. That way, all 16 bits of K_5 can be retrieved by performing the attack 16 times, each time with another key difference. The rest of the key can be retrieved using auxiliary techniques.

This variant of the attack requires 256 chosen plaintexts encrypted under two keys (K_a and K_b), and sixteen times the decryption of 512 adaptive chosen ciphertexts decrypted under two related keys. The total data complexity of the attack is 2^{13} adaptive chosen plaintexts and ciphertexts encrypted under 34 keys. The time complexity of the attack is less than 2^{13} KASUMI encryptions.

6 Summary and Conclusions

In this paper we apply the related-key boomerang and related-key rectangle attacks to the KASUMI block cipher. Our attacks are first attacks on the full cipher. The related-key rectangle attack requires $2^{54.6}$ chosen plaintexts encrypted under four keys ($2^{52.6}$ plaintexts encrypted under each key). The time complexity is equivalent to $2^{76.1}$ KASUMI encryptions.

We also present an efficient related-key boomerang distinguisher on 6-round KASUMI requires 768 adaptive chosen plaintexts and ciphertexts, using four related keys.[3] This attack can be converted to a key recovery attack that requires 2^{13} adaptive chosen plaintexts and ciphertexts encrypted under 34 related keys, and finds 16 key bits with time complexity of less than 2^{13} KASUMI encryptions.

Previous works show that the security of the KASUMI block cipher with respect to related-key attacks is significant for proving that the modes of operations used in the 3GPP networks are secure. Our results show that KASUMI cannot be considered secure with respect to differential-based related-key attacks. Therefore, the currently existing security proofs of the protocols of the 3GPP network should be revised to reflect this situation.

Acknowledgments

It is a pleasure to thank Elad Barkan for useful references, and to Tetsu Iwata for providing us with a clear understanding of the model and requirements of the security proofs. The valuable comments made by the anonymous referees are also appreciated.

References

1. Ishai Ben-Aroya, Eli Biham, *Differential Cryptanalysis of Lucifer*, Advances in Cryptology, proceedings of EUROCRYPT '93, Lecture Notes in Computer Science 773, pp. 187–199, Springer-Verlag, 1994.

[3] We expect to be able to reduce this complexity even further, but decided to save some of our time.

2. Eli Biham, *New Types of Cryptanalytic Attacks Using Related Keys (Extended Abstract)*, Journal of Cryptology, Vol. 7, No. 4, pp. 229–246, Springer-Verlag, 1994.
3. Eli Biham, *How to decrypt or even substitute DES-encrypted messages in* 2^{28} *steps*, Information Processing Letters, Vol. 84, No. 3, pp. 117–124, Elsevier, 2002.
4. Eli Biham, Alex Biryukov, Adi Shamir, *Miss in the Middle Attacks on IDEA and Khufu*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 124–138, Springer-Verlag, 1999.
5. Eli Biham, Alex Biryukov, Adi Shamir, *Cryptanalysis of Skipjack Reduced to 31 Rounds*, Advances in Cryptology, proceedings of EUROCRYPT '99, Lecture Notes in Computer Science 1592, pp. 12–23, Springer-Verlag, 1999.
6. Eli Biham, Orr Dunkelman, Nathan Keller, *The Rectangle Attack – Rectangling the Serpent*, Advances in Cryptology, proceedings of EUROCRYPT '01, Lecture Notes in Computer Science 2045, pp. 340–357, Springer-Verlag, 2001.
7. Eli Biham, Orr Dunkelman, Nathan Keller, *New Results on Boomerang and Rectangle Attacks*, proceedings of Fast Software Encryption 9, Lecture Notes in Computer Science 2365, pp. 1–16, Springer-Verlag, 2002.
8. Eli Biham, Orr Dunkelman, Nathan Keller, *Related-Key Boomerang and Rectangle Attacks*, Advances in Cryptology, proceedings of EUROCRYPT '05, Lecture Notes in Computer Science 3494, pp. 507–525, Springer-Verlag, 2005.
9. Eli Biham, Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
10. Alex Biryukov, Jorge Nakahara J., Bart Preneel, Joos Vandewalle, *New Weak-Key Class of IDEA*, proceedings of Information and Communications Security 4, Lecture Notes in Computer Science 2513, pp. 315–326, Springer-Verlag, 2002.
11. Alex Biryukov, Sourav Mukhopadhyay, Palash Sarkar, *Improved Time-Memory Trade-offs with Multiple Data* preproceedings of Selected Areas in Cryptography 2005, pp. 113-131, 2005, to appear in LNCS.
12. Mark Blunden, Adrian Escott, *Related Key Attacks on Reduced Round KASUMI*, proceedings of Fast Software Encryption 8, Lecture Notes in Computer Science 2355, pp. 277–285, Springer-Verlag, 2002.
13. Lawrence Brown, Josef Pieprzyk, Jennifer Seberry, *LOKI — A Cryptographic Primitive for Authentication and Secrecy Applications*, Advances in Cryptology, proceedings of AUSCRYPT '90, Lecture Notes in Computer Science 453, pp. 229–236, Springer-Verlag, 1990.
14. Joan Daemen, Vincent Rijmen, *The design of Rijndael: AES — the Advanced Encryption Standard*, Springer-Verlag, 2002.
15. Helena Handschuh, David Naccache, *SHACAL*, preproceedings of NESSIE first workshop, Leuven, 2000.
16. Dowon Hong, Ju-Sung Kang, Bart Preneel, Heuisu Riu, *A Concrete Security Analysis for 3GPP-MAC*, proceedings of Fast Software Encryption 10, Lecture Notes in Computer Science 2887, pp. 154–169, Springer-Verlag, 2003.
17. Tetsu Iwata, Kaoru Kurosawa, *On the Correctness of Security Proofs for the 3GPP Confidentiality and Integrity Algorithms*, proceedings of Cryptography and Coding — 9th IMA International Conference, Lecture Notes in Computer Science 2898, pp. 306–318, Springer-Verlag, 2003.
18. Tetsu Iwata, Tadayoshi Kohno, *New Security Proofs for the 3GPP Confidentiality and Integrity Algorithms*, proceedings of Fast Software Encryption 11, Lecture Notes in Computer Science 3017, pp. 427–445, Springer-Verlag, 2004.
19. Goce Jakimoski, Yvo Desmedt, *Related-Key Differential Cryptanalysis of 192-bit Key AES Variants*, proceedings of Selected Areas in Cryptography 2003, Lecture Notes in Computer Science 3006, pp. 208–221, Springer-Verlag, 2004.

20. Ju-Sung Kang, Sang Uk Shin, Dowon Hong, Okyeon Yi, *Provable Security of KASUMI and 3GPP encryption mode*, Advances in Cryptology, proceedings of ASIACRYPT '01, Lecture Notes in Computer Science 2248, pp. 255–271, Springer-Verlag, 2001.
21. John Kelsey, Tadayoshi Kohno, Bruce Schneier, *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent*, proceedings of Fast Software Encryption 7, Lecture Notes in Computer Science 1978, pp. 75–93, Springer-Verlag, 2000.
22. John Kelsey, Bruce Schneier, David Wagner, *Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA*, proceedings of Information and Communication Security 1997, Lecture Notes in Computer Science 1334, pp. 233–246, Springer-Verlag, 1997.
23. Jongsung Kim, Guil Kim, Seokhie Hong, Sangjin Lee, Dowon Hong, *The Related-Key Rectangle Attack — Application to SHACAL-1*, proceedings of ACISP 2004, Lecture Notes in Computer Science 3108, pp. 123–136, Springer-Verlag, 2004.
24. Seokhie Hong, Jongsung Kim, Guil Kim, Sangjin Lee, Bart Preneel, *Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192*, proceedings of Fast Software Encryption 12, Lecture Notes in Computer Science 3557, pp. 368–383, Springer-Verlag, 2005.
25. Ulrich Kühn, *Cryptanalysis of Reduced-Round MISTY*, Advances in Cryptology, proceedings of EUROCRYPT '01, Lecture Notes in Computer Science 2045, pp. 325–339, Springer-Verlag, 2001.
26. Xuejia Lai, James L. Massey, *A Proposal for a New Block Cipher Encryption Standard*, Advances in Cryptology, proceeding of EUROCRYPT '90, Lecture Notes in Computer Science 473, pp. 389–404, Springer-Verlag, 1991.
27. Mitsuru Matsui, *Block encryption algorithm MISTY*, proceedings of Fast Software Encryption 4, Lecture Notes in Computer Science 1267, pp. 64–74, Springer-Verlag, 1997.
28. US National Bureau of Standards, *Data Encryption Standard*, Federal Information Processing Standards Publications No. 46, 1977.
29. Arthur Sorkin, *Lucifer, a Cryptographic Algorithm*, Cryptologia, Vol. 8, No. 1, pp. 22–41, 1984.
30. Hidema Tanaka, Chikashi Ishii, Toshinobu Kaneko, *On the Strength of KASUMI without FL Functions against Higher Order Differential Attack*, proceedings of Information Security and Cryptology 3, Lecture Notes in Computer Science 2015, pp. 14–21, Springer-Verlag, 2001.
31. 3rd Generation Partnership Project, Technical Specification Group Services and System Aspects, 3G Security, *Specification of the 3GPP Confidentiality and Integrity Algorithms; Document 2: KASUMI Specification*, V.3.1.1, 2001.
32. Serge Vaudenay, *Provable Security for Block Ciphers by Decorrelation*, proceedings of Annual Symposium on Theoretical Aspects of Computer Science '98, Lecture Notes in Computer Science 1373, pp. 249–275, Springer-Verlag, 1998.
33. David Wagner, *The Boomerang Attack*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 156–170, 1999.