Related-Key Boomerang and Rectangle Attacks[⋆,⋆⋆]

Jongsung Kim[1], Seokhie Hong[2], Bart Preneel[3], Eli Biham[4], Orr Dunkelman[3,5,7,⋆⋆⋆], and Nathan Keller[6,7,†]

[1] Division of e-Business,
Kyungnam University,
449, Wolyeong-dong, Masan, Kyungnam, 631-701, Korea,
jongsung.k@gmail.com

[2] Center for Information Security Technologies (CIST),
Korea University, Anam Dong, Sungbuk Gu, Seoul, Korea,
hsh@cist.korea.ac.kr

[3] Katholieke Universiteit Leuven,
Department of Electrical Engineering ESAT/SCD-COSIC,
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium,
bart.preneel@esat.kuleuven.be

[4] Computer Science Department,
Technion, Haifa 32000, Israel,
biham@cs.technion.ac.il

[5] École Normale Supérieure
Département d'Informatique,
45 rue d'Ulm, 75230 Paris, France.

[6] Einstein Institute of Mathematics, Hebrew University.
Jerusalem 91904, Israel

[7] Faculty of Mathematics and Computer Science
Weizmann Institute of Science
P.O. Box 26, Rehovot 76100, Israel
{orr.dunkelman,nathan.keller}@weizmann.ac.il

Abstract. This paper introduces the related-key boomerang and the related-key rectangle attacks. These new attacks can expand the cryptanalytic toolbox, and can be applied to many block ciphers. The main advantage of these new attacks, is the ability to exploit the related-key model twice. Hence, even ciphers which were considered resistant to either boomerang or related-key differential attacks may be broken using the new techniques.

[⋆] This paper is partially based on the papers [9, 10, 27] which appeared at EUROCRYPT 2005, ASIACRYPT 2005 and FSE 2007, respectively.

[⋆⋆] This paper was submitted to a Journal on October 2009.

[⋆⋆⋆] The fifth author was partially supported by the Clore scholarship programme and in part by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government and by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy).

[†] The sixth author was partially supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities.

In this paper we present a rigorous treatment of the related-key boomerang and the related-key rectangle distinguishers. Following this treatment, we devise optimal distinguishing algorithms using the LLR (Logarithmic Likelihood Ratio) statistics. We then analyze the success probability under reasonable independence assumptions, and verify the computation experimentally by implementing an actual attack on a 6-round variant of KASUMI. The paper ends with a demonstration of the strength of our new proposed techniques with attacks on 10-round AES-192 and the full KASUMI.

Keywords: Related-key Boomerang Attack, Related-Key Rectangle Attack, AES, KASUMI.

1 Introduction

Related-key differentials are an extension of differentials, where the adversary is allowed a control over the key difference along with control over the plaintext/ciphertext differences [23]. The additional control gives the adversary the possibility to cancel differences that enter the nonlinear parts of the cipher, and as a result, the probability of the differential is increased.

While the use of related-key differentials in differential attacks has been studied for more than a decade, the idea of using the related-key differentials in more complex attacks has not been as extensively studied. Although techniques like related-key impossible differential [22] and related-key differential-linear cryptanalysis [11] were used to attack specific ciphers, no systematic analysis was suggested.

In this paper we examine the applicability of related-key differentials in the boomerang and the rectangle attacks. We show that it is possible to change the differentials into related-key differentials, and allow the adversary to enjoy the related-key advantage twice, using two related-key differentials at the expense of using four keys for the attack.[8]

1.1 The Related-Key Model

The *related-key model* was introduced in [4, 28] and deals with attack scenarios where the adversary is given access to encryption under multiple unknown keys, such that the relation between them is known to (or even chosen by) the adversary. While this model might seem too strong, it has practical implications. Amongst the many issues, a block cipher which is not secure against related-key attacks might fail as a hash function (for example, a related-key attack on the block cipher TEA [40] used in Microsoft's Xbox architecture as a hash function, was used to hack the system).

Related-key attacks were intensively studied in the last decade, both from the theoretical [2] and the practical [11, 12, 19, 22, 43] points of view. Immunity to related-key attacks is considered one of the security goals in the design of modern block ciphers [16].

[8] We note that in some cases the number of keys remains two.

1.2 The Boomerang and the Rectangle Attacks

The boomerang attack [38] is a differential-based attack that uses two short differentials (of few rounds each) rather than one long differential (of many rounds). In an adaptive chosen plaintext and ciphertext process, the adversary constructs boomerang quartets by exploiting the short differentials.

The attack was later transformed into a chosen plaintext variant named the amplified boomerang attack [25] (and then renamed as the rectangle attack [7]). The transformation is done by a birthday-paradox argument, which leads to a higher data complexity, but still allows the use of two short differentials.

All of these attacks treat the distinguished part of the cipher E as a decomposition into two sub-ciphers, $E = E_1 \circ E_0$, where in each of these two sub-ciphers some (relatively) high probability differential exists. If the probability of the differential of E_0 is p and the probability of the differential of E_1 is q, then the data complexity of the corresponding boomerang distinguisher is $O((pq)^{-2})$ adaptively chosen plaintexts and ciphertexts and of the rectangle distinguisher is $O(2^{n/2} \cdot (pq)^{-1})$ chosen plaintexts, where n is the block size.

In the more complex variants of these attacks the use of multiple differentials is supported as long as they share the input difference in the differentials for E_0 (the first rounds) and share the output difference in the differentials for E_1 (the last rounds). This improvement reduces the data complexity of both attacks significantly.

1.3 Our Contributions

We consider the same conceptual division. Let the cipher E be a concatenation of two sub-ciphers, i.e., $E = E_1 \circ E_0$. Furthermore, assume that there exist high-probability related-key differentials in E_0 and in E_1 (not necessarily with the same key difference). We show that in this scenario it is possible to apply related-key boomerang and rectangle attacks. The basic related-key boomerang and rectangle distinguishers are summarized in the following theorem:

Theorem 1. *Let $E = E_1 \circ E_0 : \{0,1\}^n \to \{0,1\}^n$ be a block cipher. Consider encryptions with E under a secret key K and related-keys whose differences with K are chosen by the adversary. Let*

$$\hat{p} = \max_{\alpha \neq 0, \Delta K_0} \sqrt{\sum_{\beta} \left(\Pr_P \left[E_{0,K}(P) \oplus E_{0,K\oplus\Delta K_0}(P \oplus \alpha) = \beta \right] \right)^2},$$

$$\hat{q} = \max_{\delta \neq 0, \Delta K_1} \sqrt{\sum_{\gamma} \left(\Pr_C \left[E^{-1}_{1,K}(C) \oplus E^{-1}_{1,K\oplus\Delta K_1}(C \oplus \delta) = \gamma \right] \right)^2}$$

$$= \max_{\delta \neq 0, \Delta K_1} \sqrt{\sum_{\gamma} \left(\Pr_X \left[E_{1,K}(X) \oplus E_{1,K\oplus\Delta K_1}(X \oplus \gamma) = \delta \right] \right)^2}$$

where $E_{0,K}(P)$ denotes the partial encryption of P through E_0 under the key K and $E^{-1}_{1,K}(C)$ denotes the partial decryption of C through E_1 under the key K. Under independence assumptions between the differentials, for $c > 0$, given either

- $4c/(\hat{p}\hat{q})^2$ *unique adaptively chosen plaintexts and ciphertexts, or*
- $\sqrt{c} \cdot 2^{n/2+2}/\hat{p}\hat{q}$ *unique chosen plaintexts,*

encrypted under four related-keys of the form $K, K \oplus \Delta K_0, K \oplus \Delta K_1, K \oplus \Delta K_0 \oplus \Delta K_1$,[9] it is possible to distinguish E from a random permutation. The probability of success of the distinguisher is approximately $1 - e^{-c}/2$ (when $\hat{p}\hat{q}$ is sufficiently high).

We present a rigorous treatment of the related-key boomerang and rectangle distinguishers. We devise the optimal distinguishing algorithms using the LLR metric, and compute their success rate. Using this analysis, we prove Theorem 1, along with an easy lemma allowing to calculate lower bounds for $\hat{p}$ and $\hat{q}$ in practical ciphers.

As in other statistical attacks on block ciphers, the calculation of the success probability of our attack is based on some randomness assumptions. We state explicitly the assumptions we use and discuss their validity in various cases. To verify the validity of these assumptions we carried out computer experiments for the related-key boomerang attack on 6-round KASUMI [36].

We note that the analysis presented in our paper is also the first rigorous analysis of the boomerang/rectangle techniques themselves. Although these techniques were used many times in attacks, a rigorous analysis of them was not performed before.

After the theoretical treatment, we consider several improvements of the related-key boomerang and rectangle attacks:

1. **The Use of Structures of Keys:** We use structures of keys to overcome a wider range of key schedule algorithms. In ciphers with a nonlinear key schedule, a given key difference may cause many subkey differences, thus interfering with the construction of related-key differentials. Structures of keys can be used to reduce the effect of this event on the differentials.
2. **The Use of Other Relations between the Keys:** While XOR relations are common and inherent to the majority of differential-based related-key attacks, in some cases there are more suitable key relations (either due to the environment of the attack or in order to get higher probabilities of the differentials). We show that the proposed attacks are applicable when the XOR relations between the keys are replaced with different kinds of relations and discuss which relations induce feasible attacks.

[9] In some cases $\Delta K_0 = \Delta K_1$. In these cases, there are small changes in the analysis, most notably the use of only two related keys.

We then compare the new attacks with previously proposed related-key techniques. We explore the advantages of the new attacks, and show that in many cases the related-key boomerang and the related-key rectangle attacks are significantly more effective than other related-key techniques, even if in the single-key scenario the boomerang and the rectangle attacks are inferior to the non-related-key techniques.

Finally, out of the many ciphers for which related-key boomerang and rectangle attacks were applied to (to mention a few, IDEA, MISTY1, SHACAL-1, SHACAL-2, and XTEA), we present two cases that demonstrate the strength and the wide applicability of the new attacks. We chose to concentrate on KASUMI and AES, as these two ciphers demonstrate the advantages of using two pairs of related-keys to overcome complex round functions (KASUMI) and using structures of keys to bypass a nonlinear key schedule (AES).

An Attack on 10-round AES-192 The *Advanced Encryption Standard (AES)* [32] is a 128-bit block cipher with a variable key length (128, 192, and 256-bit keys are supported). Since its selection, AES gradually became one of the most widely used and analyzed block ciphers. The cipher has received a great deal of cryptanalytic attention, both during the AES process, and even more after its selection.

We present a related-key rectangle attack on 10-round AES-192 requiring $2^{119.2}$ chosen plaintexts encrypted under one of 64 related keys and time complexity of $2^{185.2}$ memory accesses. Our attack uses structures of 64 keys in order to overcome the nonlinearity of the AES key schedule. We summarize our results along with selected other results in Table 1.[10]

An Attack on the Full 8-Round KASUMI KASUMI is an 8-round Feistel block cipher used in the confidentiality and the integrity algorithms of some 3GPP mobile communications. Since the 3GPP mobile communications are used by millions of customers, KASUMI is one of the most widely used block ciphers.

We start by presenting a simple 6-round related-key boomerang attack on 6-round KASUMI, which has a practical data and time complexity. We follow to present a related-key rectangle attack on the full 8-round requiring $2^{54.6}$ chosen plaintexts and $2^{73.6}$ encryptions.

The cases of AES and KASUMI show the advantage of the related-key boomerang/rectangle attack over the other related-key attacks. While the other techniques can attack the same number of rounds as the best single-key attacks (8 rounds for AES-192 and 6 rounds for KASUMI), the related-key boomerang/rectangle attacks can attack either two more rounds (10 rounds for AES-192 and the full 8-round KASUMI), or the same number of rounds with a significantly lower

[10] We note that an independent related-key boomerang attack on 9-round AES-192 was presented recently in [20]. Also, related-key boomerang attacks on the full AES-192 and AES-256 were presented in [14] using a stronger model of related keys.

complexity. We summarize our results along with selected other results in Table 1.

Table 1. Comparison of our Attacks with Selected Previous Results

Cipher	Attack	Number of		Complexity	
		Rounds	Keys	Data	Time
KASUMI (8 rounds)	Imp. Diff. [29]	6	1	2^{55} CP	2^{100}
	RK Diff. [15]	6	2	$2^{18.6}$ RK-CP	$2^{113.6}$
	RK Boom. (Sect. 3.3)	$6^{\dagger}$	4	768 RK-ACPC	1
	RK Rect. (Sect. 3.6)	8	4	$2^{54.6}$ RK-CP	$2^{73.6}$
	RK Rect. (Sect. 3.7)	8	4	$2^{38.6}$ RK-CP	2^{104}
AES-192 (12 rounds)	Partial Sums [19]	8	1	$2^{128} - 2^{119}$ CP	2^{188}
	RK Diff.-Lin. [43]	8	2	2^{118} RK-CP	2^{165}
	RK Imp. Diff. [42]	8	2	$2^{64.5}$ RK-CP	2^{177}
		8	2	2^{88} RK-CP	2^{153}
		8	2	2^{112} RK-CP	2^{136}
	RS Rectangle [14]	12	4	2^{123} RS-CP	2^{176}
	RK Rect. (Sect. 4.4)	10	256	$2^{121.2}$ RK-CP	$2^{184.2}$ MA
	(Sect. 4.5)	10	64	$2^{119.2}$ RK-CP	$2^{185.2}$ MA

CP – chosen plaintexts, ACPC – adaptive chosen plaintexts and ciphertexts,
RK: Related-Key, RS: Related-Subkey
†: distinguishing attack, MA: Memory accesses
Time is measured in encryption units unless mentioned otherwise

1.4 The Organization of the Paper

The paper is organized as follows: In Section 2 we present the related-key boomerang and rectangle attacks and discuss them theoretically. In Section 3 we apply the attacks to the full KASUMI. In Section 4 we apply the attacks to reduced-round AES-192. Finally, Section 5 summarizes the paper.

2 The Related-Key Boomerang and Rectangle Attacks

In this section we introduce the related-key boomerang and the related-key rectangle attacks. We start with a brief description of the boomerang and the rectangle attacks in the single key model. We then introduce and analyze rigorously the related-key boomerang and rectangle attacks. We follow and examine the randomness assumptions used in the attacks. We conclude this section with several generalizations and comparisons of the newly proposed attacks.

2.1 Boomerang and Amplified Boomerang (Rectangle) Attacks

The main idea behind the boomerang attack [38] is to use two short differentials with high probabilities instead of one long differential with a low probability. We assume that a block cipher $E:\{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$ can be described as a cascade $E = E_1 \circ E_0$, such that for E_0 there exists a differential $\alpha \rightarrow \beta$ with probability p, and for E_1 there exists a differential $\gamma \rightarrow \delta$ with probability q.[11]

The distinguisher is based on the following boomerang process:

1. 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).
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).
3. Check whether $P_3 \oplus P_4 = \alpha$.

The boomerang attack uses the first differential ($\alpha \rightarrow \beta$) for E_0 with respect to the pairs (P_1, P_2) and (P_3, P_4), and the second differential ($\gamma \rightarrow \delta$) 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}, where n is the block size.[12] For E, the probability that the pair (P_1, P_2) is a right pair with respect to the first differential (i.e., the probability that the intermediate difference after E_0 equals β, as predicted by the differential) 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, $P_3 \oplus P_4 = \alpha$. Hence, the total probability of this quartet of plaintexts and ciphertexts to satisfy the condition $P_3 \oplus P_4 = \alpha$ is at least $(pq)^2$.

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 not less than $(\hat{p}\hat{q})^2$, where:

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

Using the boomerang process described above, the cipher E can be distinguished from a random permutation given $O((\hat{p}\hat{q})^{-2})$ adaptively chosen plaintexts and ciphertexts, provided that $\hat{p}\hat{q} \gg 2^{-n/2}$. The complete analysis is given in [7, 8, 38]. We omit the analysis here since it is essentially included in the analysis of the related-key boomerang attack presented in Section 2.2.

As the boomerang distinguisher requires adaptively chosen plaintexts and ciphertexts, it cannot be combined with many of the standard techniques for

[11] We note that in the attack, the differentials are used both in the forward (i.e., encryption), and in the backward (i.e., decryption) directions. As the considered differentials are not truncated differentials, the direction does not affect the probability of the differentials.

[12] For the analysis of E we rely on some independence assumptions, addressed in Section 2.4.

using distinguishers in key recovery attacks. This led to the introduction of a chosen plaintext variant of the boomerang attack called the *amplified boomerang attack* [25], and later renamed as the *rectangle attack* [7]. The transformation of the boomerang attack into a chosen plaintext attack relies on standard birthday-paradox arguments. The key idea behind the transformation is to encrypt many plaintext pairs with input difference α, and to look for quartets (i.e., pairs of pairs) that conform to the requirements of the boomerang process.

In the rectangle distinguisher, the adversary considers quartets of plaintexts of the form $((P_1, P_2 = P_1 \oplus \alpha), (P_3, P_4 = P_3 \oplus \alpha))$. A quartet is called a "right quartet" if the following conditions are satisfied:

1. $E_0(P_1) \oplus E_0(P_2) = \beta = E_0(P_3) \oplus E_0(P_4)$.
2. $E_0(P_1) \oplus E_0(P_3) = \gamma$ (which leads to $E_0(P_2) \oplus E_0(P_4) = \gamma$ if this condition holds along with the previous one).
3. $C_1 \oplus C_3 = \delta = C_2 \oplus C_4$.

The probability of a quartet to be a right quartet is a lower bound on the probability of the event

$$C_1 \oplus C_3 = \delta = C_2 \oplus C_4. \tag{1}$$

The usual assumption is that each of the above conditions is independent of the rest, and hence the probability that a given quartet $((P_1, P_2), (P_3, P_4))$ is a right quartet is $p^2 \cdot 2^{-n-1} \cdot q^2$. Since for a random permutation, the probability of Condition (1) is 2^{-2n}, the rectangle process can be used to distinguish E from a random permutation if $pq \gg 2^{-n/2}$ (like in the boomerang distinguisher).

The data complexity of the distinguisher is $O(2^{n/2}(pq)^{-1})$, which is much higher than the complexity of the boomerang distinguisher. The higher data complexity follows from the fact that the event $E_0(P_1) \oplus E_0(P_3) = \gamma$ occurs with a "random" probability of 2^{-n} (actually, this is the birthday-paradox argument used in the construction). The identification of right quartets is also more complicated than in the boomerang case, as instead of checking a condition on pairs, the adversary has to go over all the possible quartets. At the same time, the chosen plaintext nature allows using stronger key recovery techniques. An optimized method of finding the right rectangle quartets is presented in [8].

Like the boomerang attack, the rectangle attack can use all the possible β's and γ's simultaneously. This reduces the data complexity of the attack to $O(2^{n/2}(\hat{p}\hat{q})^{-1})$, where $\hat{p}$ and $\hat{q}$ are as defined above. The complete analysis of the rectangle attack is given in [7, 8].

2.2 The Related-Key Boomerang Attack

We now present the related-key boomerang distinguisher, and determine the conditions required for the distinguisher to succeed. Following a rigorous treatment we compute the optimal value of the threshold used in the distinguisher using the Logarithmic Likelihood Ratio method. Then we compute the success

rate of the distinguisher using a Poisson approximation. In order to keep this section readable, we refrain from presenting a detailed analysis of the key-recovery attack algorithm. The reader is referred to [8] for a generic key-recovery attack algorithm exploiting the boomerang distinguisher (which is easily adapted to the related-key model), and to the specific attack algorithms presented in Sections 3 and 4.
First, we recall the definition of related-key differentials and introduce a shorthand used throughout this paper to denote them:

Definition 1. *We say that a related-key differential $\alpha \to \beta$ with key difference ΔK holds for E with probability p, if*

$$\Pr_{P,K}\left[E_K(P) \oplus E_{K\oplus\Delta K}(P \oplus \alpha) = \beta\right] = p,$$

where $E_K(\cdot)$ denotes encryption through E with the key K. For the ease of exposition, we denote this event by $\Pr\left(\alpha \xrightarrow[\Delta K]{E} \beta\right) = p$. For sake of simplicity, we shall denote the related-key differential by $\left(\alpha \xrightarrow[\Delta K]{E} \beta\right)$ or when the cipher E is implicit from the text by $\left(\alpha \xrightarrow[\Delta K]{} \beta\right)$ (we alert the reader that this notation is not common).

In order to present the independence assumption used in the paper, we need another definition:

Definition 2. *For each related-key differential $\left(\alpha \xrightarrow[\Delta K]{E} \beta\right)$, we denote the set of right pairs with respect to the differential (for the given key K) by $G_K\left(\alpha \xrightarrow[\Delta K]{E} \beta\right)$. Formally, for a block cipher E and a given key K,*

$$G_K\left(\alpha \xrightarrow[\Delta K]{E} \beta\right) = \left\{P \middle| E_K(P) \oplus E_{K\oplus\Delta K}(P \oplus \alpha) = \beta\right\}.$$

Similarly, we define the set of good ciphertexts:

$$G_K^{-1}\left(\alpha \xrightarrow[\Delta K]{E} \beta\right) = \left\{E_K(P) \middle| P \in G\left(\alpha \xrightarrow[\Delta K]{E} \beta\right)\right\} = \left\{C \middle| E_K^{-1}(C) \oplus E_{K\oplus\Delta K}^{-1}(C \oplus \beta) = \alpha\right\}.$$

Our independence assumption asserts that the sets of the form $G\left(\alpha \xrightarrow[\Delta K]{E} \beta\right)$ are independent, in the following sense:

Assumption 1 *For the block cipher $E = E_1 \circ E_0$ under consideration, for any fixed key K, and for any set of differences $\alpha, \gamma_1, \delta, \Delta K_0$, and ΔK_1, we assume that the event $\left(X \in G_K\left(\gamma_1 \xrightarrow[\Delta K_1]{E_1} \delta\right)\right)$ is independent of any combination of these three events:*

1. $\left(X \oplus \beta_1 \in G_{K \oplus \Delta K_0}\left(\gamma_2 \xrightarrow[\Delta K_1]{E_1} \delta\right)\right)$ *for all* β_1, γ_2.

2. $\left(X \in G_K^{-1}\left(\alpha \xrightarrow[\Delta K_0]{E_0} \beta_1\right)\right)$ *for all* β_1.

3. $\left(X \oplus \gamma_1 \in G_{K \oplus \Delta K_1}^{-1}\left(\alpha \xrightarrow[\Delta K_0]{E_0} \beta_2\right)\right)$ *for all* β_2.

For example, our independence assumption asserts that

$$\Pr\left[X \in G_K\left(\gamma_1 \xrightarrow[\Delta K_1]{E_1} \delta\right) \middle| \left(X \oplus \beta_1 \in G_{K \oplus \Delta K_0}\left(\gamma_2 \xrightarrow[\Delta K_1]{E_1} \delta\right)\right) \bigwedge \right.$$

$$\left. \left(X \in G_K^{-1}\left(\alpha \xrightarrow[\Delta K_0]{E_0} \beta_1\right)\right) \bigwedge \left(X \oplus \gamma_1 \in G_{K \oplus \Delta K_1}^{-1}\left(\alpha \xrightarrow[\Delta K_0]{E_0} \beta_2\right)\right)\right]$$

$$= \Pr\left[X \in G_K\left(\gamma_1 \xrightarrow[\Delta K_1]{E_1} \delta\right)\right].$$

This assumption is used implicitly in all the statements in the sequel. We discuss the assumption and its relation to the independence assumptions used in other techniques, such as differential and linear cryptanalysis, in Section 2.4.

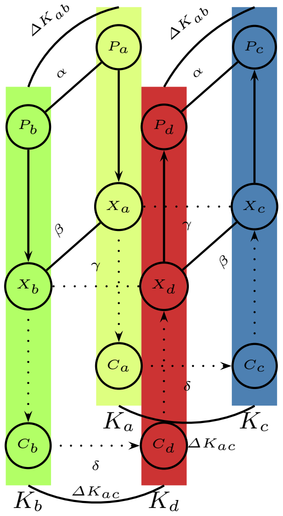
The Related-Key Boomerang Distinguisher Now we are ready to present the related-key boomerang distinguisher. Similarly to the boomerang attack, we treat the cipher E as a cascade of sub-ciphers: $E = E_1 \circ E_0$. The distinguisher involves four different unknown (but related) keys — K_a, $K_b = K_a \oplus \Delta K_{ab}$, $K_c = K_a \oplus \Delta K_{ac}$, and $K_d = K_a \oplus \Delta K_{ab} \oplus \Delta K_{ac}$. For fixed values α and δ, the attack algorithm is the following:

1. Choose M plaintexts at random, and set a counter C to zero. For each plaintext P_a, perform the following:
   (a) Compute $P_b = P_a \oplus \alpha$.
   (b) Ask for the ciphertexts $C_a = E_{K_a}(P_a)$ and $C_b = E_{K_b}(P_b)$.
   (c) Compute $C_c = C_a \oplus \delta$ and $C_d = C_b \oplus \delta$.
   (d) Ask for the plaintexts $P_c = E_{K_c}^{-1}(C_c)$ and $P_d = E_{K_d}^{-1}(C_d)$.
   (e) Check whether $P_c \oplus P_d = \alpha$. If yes, increase the value of the counter C by 1.
2. If $C > \mathit{Threshold}$, output "The cipher E". Otherwise, output "Random Permutation".

The value of $\mathit{Threshold}$ will be specified later in this section. See Figure 1 for an outline of a right related-key boomerang quartet.

It is easy to see that for a random permutation, the probability that the condition $P_c \oplus P_d = \alpha$ is satisfied is 2^{-n}. The probability that the condition is satisfied for E is given in the following proposition:

Fig. 1. A Related-Key Boomerang Quartet



Proposition 1. *Consider a quartet (P_a, P_b, P_c, P_d) constructed by the algorithm described above. We have*

$$\Pr[P_c \oplus P_d = \alpha] = \sum_{\beta_1 \oplus \beta_2 \oplus \gamma_1 \oplus \gamma_2 = 0} \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_1\right] \cdot \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_2\right] \cdot \Pr\left[\gamma_1 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right] \cdot \Pr\left[\gamma_2 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right]. \quad (2)$$

In particular,

$$\Pr[P_c \oplus P_d = \alpha] \geq (\hat{p}\hat{q})^2, \quad (3)$$

where

$$\hat{p} = \sqrt{\sum_{\beta'} \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta'\right]^2} \text{ and } \hat{q} = \sqrt{\sum_{\gamma'} \Pr\left[\gamma' \xrightarrow[\Delta K_{ac}]{E_1} \delta\right]^2}.$$

Proof. Consider a quartet (P_a, P_b, P_c, P_d) constructed by the algorithm. Denote the intermediate values $(E_0(P_a), E_0(P_b), E_0(P_c), E_0(P_d))$ (where the encryption is under the respective keys) by (X_a, X_b, X_c, X_d), respectively. For all $\beta_1, \gamma_1, \gamma_2$, we say that the event $S_{\beta_1,\gamma_1,\gamma_2}$ occurs, if the following conditions are satisfied:

$$X_a \oplus X_b = \beta_1, \qquad X_a \oplus X_c = \gamma_1, \qquad X_b \oplus X_d = \gamma_2.$$

Since the events $\{S_{\beta_1,\gamma_1,\gamma_2}\}$ for different values of $(\beta_1, \gamma_1, \gamma_2)$ are disjoint and their union is the entire space, we have

$$\Pr[P_c \oplus P_d = \alpha] = \sum_{\beta_1,\gamma_1,\gamma_2} \Pr[P_c \oplus P_d = \alpha | S_{\beta_1,\gamma_1,\gamma_2}] \cdot \Pr[S_{\beta_1,\gamma_1,\gamma_2}]. \quad (4)$$

If the event $S_{\beta_1,\gamma_1,\gamma_2}$ occurs, then

$$X_c \oplus X_d = (X_c \oplus X_a) \oplus (X_a \oplus X_b) \oplus (X_b \oplus X_d) = \gamma_1 \oplus \beta_1 \oplus \gamma_2.$$

Hence, by the independence assumptions,

$$\Pr\left[P_c \oplus P_d = \alpha \middle| S_{\beta_1,\gamma_1,\gamma_2}\right] = \Pr[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_2], \tag{5}$$

where $\beta_2 = \gamma_1 \oplus \beta_1 \oplus \gamma_2$. Similarly, the three conditions forming the event $S_{\beta_1,\gamma_1,\gamma_2}$ are independent, and hence

$$\Pr[S_{\beta_1,\gamma_1,\gamma_2}] = \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_1\right] \cdot \Pr\left[\gamma_1 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right] \cdot \Pr\left[\gamma_2 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right]. \tag{6}$$

Substituting Equations (5) and (6) into Equation (4) yields Equation (2).

$$\sum_{\beta_1 \oplus \beta_2 \oplus \gamma_1 \oplus \gamma_2 = 0} \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_1\right] \cdot \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_2\right] \cdot \Pr\left[\gamma_1 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right] \cdot \Pr\left[\gamma_2 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right] \cdot \geq$$

$$\geq \sum_{\substack{\beta_1 \oplus \beta_2 = 0, \\ \gamma_1 \oplus \gamma_2 = 0}} \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_1\right] \cdot \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_2\right] \cdot \Pr\left[\gamma_1 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right] \cdot \Pr\left[\gamma_2 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right] =$$

$$\sum_{\beta'} \left(\Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta'\right]^2\right) \sum_{\gamma'} \left(\Pr\left[\gamma' \xrightarrow[\Delta K_{ac}]{E_1} \delta\right]^2\right) = (\hat{p}\hat{q})^2,$$

and thus, Inequality (3) follows from Equation (2).

Proposition 1 shows that if $\hat{p}\hat{q} > 2^{-n/2}$, then the probability that the condition $P_c \oplus P_d = \alpha$ holds, is higher for E than for a random permutation, i.e., we expect more quartets in the case of E. We next compute the optimal choice of the value *Threshold* used in the distinguisher.

The Optimal Choice of *Threshold* The optimal value of *Threshold* can be found using the Likelihood Ratio test for the distributions representing $\Pr[P_c \oplus P_d = \alpha]$ for E and for a random permutation. We use the following standard result:

Proposition 2 ([1], Proposition 1). *Consider two distributions D_0 and D_1 assuming values in a finite set Z, and a sample z^m of m independent elements of Z (represented as a vector in Z^m). The optimal test for deciding whether the sample is distributed according to D_0 or to D_1 is the test having acceptance region*

$$A_{D_0} = \{z^m \in Z^m : LLR(z^m) \geq 0\},$$

where

$$LLR(z^m) = \sum_{a \in Z} N(a|z^m) \log \frac{\Pr_{D_0}[a]}{\Pr_{D_1}[a]}$$

is the logarithmic likelihood ratio (with the convention that $\log(0/p) = -\infty$ *and* $\log(p/0) = \infty$*), and where* $N(a|z^m)$ *is the number of times* a *occurs in the sequence* z^m*.*

Denote $p_0 = \Pr[P_c \oplus P_d = \alpha]$ (where P_c and P_d are constructed by the boomerang process). We apply Proposition 2, where D_0 and D_1 are the distributions representing $\Pr[P_c \oplus P_d = \alpha]$ for E and for a random permutation, respectively. In this case, $Z = \{0, 1\}$, $m = M$, and both distributions represent Bernoulli random variables, where $D_0 = Ber(p_0)$ and $D_1 = Ber(2^{-n})$. Hence,

$$LLR(z^M) = N(0|z^M) \log \frac{1 - p_0}{1 - 2^{-n}} + N(1|z^M) \log \frac{p_0}{2^{-n}}. \tag{7}$$

Since in our distinguisher, the acceptance region of the test is $\{z^M \in Z^M : N(1|z^M) \geq Threshold\}$, the optimal value of $Threshold$ is $\min\{k : f(k) \geq 0\}$, where

$$f(k) = (M - k) \log \frac{1 - p_0}{1 - 2^{-n}} + k \log \frac{p_0}{2^{-n}}.$$

A simple computation shows that the optimal value is

$$Threshold = \left\lceil \frac{-\log \frac{1-p_0}{1-2^{-n}}}{\log \frac{p_0(1-2^{-n})}{(1-p_0)2^{-n}}} M \right\rceil. \tag{8}$$

The Success Probability of the Distinguisher We use the following standard definition of the success probability of a distinguisher (see, e.g., [1]):

Definition 3. *Let* A *be a distinguisher between distributions* D_0 *and* D_1*, such that for* $j = 0, 1$*, the statement* $[A(D) = j]$ *corresponds to "*D *is distributed like* D_j*". The probability of success of* A *is*

$$\Pr{}_s(A) = \frac{\Pr[A(D) = 0|D = D_0] + \Pr[A(D) = 1|D = D_1]}{2}.$$

Since the distinguisher counts the number of successes amongst M trials, it actually distinguishes between the Binomial distributions $Bin(M, p_0)$ and $Bin(M, 2^{-n})$. Hence, given the value $Threshold$ (as computed in Equation 8), the success probability of the distinguisher is given by the formula:

$$\Pr[Success] = \frac{1}{2}\Big[\Pr[Bin(M, p_0) \geq Threshold] + \Pr[Bin(M, 2^{-n}) < Threshold]\Big] =$$

$$= \frac{1}{2}\left[\sum_{k=Threshold}^{M}\binom{M}{k}p_0^k(1-p_0)^{M-k}\sum_{k=0}^{Threshold-1}\binom{M}{k}2^{-nk}(1-2^{-n})^{M-k}\right]. \tag{9}$$

For a large value of M (like the values usually used in attacks as M has to be at least $1/p_0$, and p_0 is in most cases very small), the Binomial distributions can be approximated by the Poisson distributions $Poi(p_0 M)$ and $Poi(2^{-n}M)$. Using this approximation, Equation (9) is simplified to:

$$\Pr[Success] \approx \frac{1}{2}\left[1 - e^{-p_0 M}\sum_{k=0}^{Threshold-1}\frac{(p_0 M)^k}{k!} + e^{-2^{-n}M}\sum_{k=0}^{Threshold-1}\frac{(2^{-n}M)^k}{k!}\right]. \tag{10}$$

Denote $c = Mp_0$, and $x = p_0/2^{-n}$. Equation (10) can be reformulated into:

$$\Pr[Success] \approx \frac{1}{2}\left[1 - \left(e^{-c}\cdot\sum_{k=0}^{Threshold-1}\frac{c^k}{k!}\right) + \left(e^{-c/x}\cdot\sum_{k=0}^{Threshold-1}\frac{(c/x)^k}{k!}\right)\right]. \tag{11}$$

We note that in actual attacks, c usually satisfies $1 \leq c \leq 100$, while the value x varies significantly between different attacks. In Table 2, we give the optimal threshold and success rate for several common values of c and x.

When x tends to infinity, Equation (11) can be simplified, as $e^{-c/x}$ tends to 1. In other words, when $x \gg 1$, given $M = c \cdot p_0^{-1}$ quartets, a threshold of 1 is sufficient to achieve the following success rate:

$$\Pr[success] \approx \frac{1}{2}\left(1 - e^{-c} + 1\right) = 1 - \frac{e^{-c}}{2}.$$

Table 2. Optimal Thresholds and Success Rates for Common Parameters

x	$c=1$	$c=2$	$c=3$	$c=4$	$c=6$	$c=8$	$c=16$
2	1 (61.9%)	2 (66.5%)	Imp	Imp	Imp	Imp	Imp
4	1 (70.5%)	2 (75.2%)	2 (81.4%)	3 (84.1%)	Imp	Imp	Imp
10	1 (76.8%)	1 (84.2%)	2 (88.2%)	2 (92.3%)	3 (95.7%)	4 (97.4%)	Imp
16	1 (78.6%)	1 (87.4%)	2 (89.3%)	2 (94.1%)	3 (96.6%)	3 (98.6%)	6 (99.9%)
100	1 (81.1%)	1 (92.2%)	1 (96.0%)	1 (97.1%)	2 (99.0%)	2 (99.7%)	4 (99.99%)
200	1 (81.4%)	1 (92.7%)	1 (96.8%)	1 (98.1%)	2 (99.1%)	2 (99.8%)	4 (99.995%)
1000	1 (81.6%)	1 (93.1%)	1 (97.4%)	1 (98.9%)	1 (99.6%)	2 (99.8%)	3 (99.999%)
10000	1 (81.6%)	1 (93.2%)	1 (97.5%)	1 (99.1%)	1 (99.8%)	1 (99.9%)	2 (99.9998%)

The entry $X(Y\%)$ means that the optimal threshold is X and the success rate is Y.
Imp — it is impossible to gather the amount of data required in this case.

We note that while for attacks based on linear cryptanalysis, the probability of success can be approximated using the Normal distribution (see, e.g., [1, 35]) in attacks based on differential cryptanalysis (like the attacks discussed in this paper) the Normal approximation may be inaccurate. The reason for the difference is that while in linear-based attacks, the value of the measured random variable is big (close to $M/2$), in differential-based attacks the value is usually very small (e.g., $1 \leq Threshold \leq 10$). For such small values, the approximation of a random variable assuming only integer values by a Normal distribution is inaccurate, and hence approximation using a Poisson random variable is preferable.[13]

Practical Lower Bounds for $\hat{p}$ and $\hat{q}$ In practical attacks, the probability of the related-key boomerang distinguisher (given by Equation 2) cannot be computed. Moreover, even the computation of the lower bound given by Inequality (3) is infeasible in most of the cases. Instead, the adversary finds high-probability differential characteristics $\left(\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta\right)$ and $\left(\gamma \xrightarrow[\Delta K_{ac}]{E_1} \delta\right)$. Then, the adversary computes lower bounds for $\hat{p}$ and $\hat{q}$ by considering only part of the possible β' and γ' values. For example, she can take into consideration all the characteristics $\left(\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta'\right)$ that coincide with the characteristic $\left(\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta\right)$ in all the rounds except for the last one, and take all possible values in the output difference of the last round.

In certain cases, especially when a good differential cannot be found, the following simple proposition is useful as a generic lower bound for $\hat{p}$ and $\hat{q}$.

Proposition 3. *Consider related-key differentials through E_0 with input difference α and key difference ΔK. If there exist only m differences β' such that $\Pr\left[\alpha \xrightarrow[\Delta K]{E_0} \beta'\right] > 0$, then $\hat{p} \geq \sqrt{1/m}$. Moreover, equality holds if and only if all the m differentials $(\alpha \xrightarrow[\Delta K]{} \beta')$ with non-zero probability have probability $1/m$ each.*

Proof. Recall that the Cauchy-Schwarz inequality asserts that for any two sequences $\{a_1, a_2, \ldots, a_m\}$ and $\{b_1, b_2, \ldots, b_m\}$ of non-negative numbers, we have

$$\sum_{i=1}^{m} a_i \cdot b_i \leq \sqrt{\sum_{i=1}^{m} a_i^2} \cdot \sqrt{\sum_{i=1}^{m} b_i^2}.$$

[13] In [35] the success probabilities of both a linear attack and a differential attack are approximated using the Normal distribution. The experiments presented in [35] show that the approximation is much more accurate in the case of linear cryptanalysis. It is possible that using a Poisson approximation would result in a better accuracy in the differential case, as explained above.

Denote the probabilities of the differentials of the form $\left(\alpha \xrightarrow[\Delta K]{} \beta'\right)$ by $p_1, p_2, \ldots, p_m$ (ignoring the differentials with zero probability). Clearly, we have

$$\sum_{i=1}^{m} p_i = 1, \qquad \text{and} \qquad \hat{p} = \sqrt{\sum_{i=1}^{m} p_i^2}.$$

We apply the Cauchy-Schwarz inequality for the sequences $\{p_1, p_2, \ldots, p_m\}$ and $\{1, 1, \ldots, 1\}$ and get

$$1 = \sum_{i=1}^{m} p_i \cdot 1 \leq \sqrt{\sum_{i=1}^{m} p_i^2} \cdot \sqrt{\sum_{i=1}^{m} 1} = \hat{p}\sqrt{m},$$

and hence $\hat{p} \geq \sqrt{1/m}$, as asserted. Furthermore, since equality in the Cauchy-Schwarz inequality holds if and only if the sequences $\{a_i\}_{i=1}^{m}$ and $\{b_i\}_{i=1}^{m}$ are proportional (i.e., there exists c such that $a_i = c \cdot b_i$ for all i), in our case equality holds if and only if all the p_i's are equal.

The generic lower bound given by Proposition 3 can be combined with a "good" differential for part of the rounds.

Proposition 4. *Consider related-key differentials through E_0 with input difference α and key difference ΔK. Assume that there exists a decomposition $E_0 = E_{01} \circ E_{00}$, and a difference α', such that:*

1. $\Pr\left[\alpha \xrightarrow[\Delta K]{E_{00}} \alpha'\right] = p'$, *and*
2. *There exist only m differences β' such that* $\Pr[\alpha' \xrightarrow[\Delta K]{E_{01}} \beta'] > 0$.

Then $\hat{p} \geq p'\sqrt{1/m}$.

Proof. We compute a lower bound on $\hat{p}$ by considering only the characteristics $\left(\alpha \xrightarrow[\Delta K]{E_0} \beta'\right)$ for E_0 whose restriction to E_{00} is $\left(\alpha \xrightarrow[\Delta K]{E_{00}} \alpha'\right)$. By the assumptions, there are only m such differentials (ignoring differentials with probability zero), and the sum of their probabilities is p'. The assertion follows from the Cauchy-Schwarz inequality by the same argument as used in the proof of Proposition 3.

Clearly, the same arguments apply also for the computation of $\hat{q}$. Propositions 3 and 4 are used in our attacks on KASUMI and AES-192, presented in Sections 3 and 4.

2.3 The Related-Key Rectangle Attack

The transformation of the related-key boomerang attack into the related-key rectangle attack is similar to the transformation of the boomerang attack to the rectangle attack in the single-key model. The related-key rectangle distinguisher 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 $K_d = K_a \oplus \Delta K_{ab} \oplus \Delta K_{ac}$. For fixed values α and δ, the algorithm of the distinguisher is as follows:

1. Choose M plaintexts P_a, and compute $P_b = P_a \oplus \alpha$. Ask for the ciphertexts $C_a = E_{K_a}(P_a)$ and $C_b = E_{K_b}(P_b)$.
2. Choose M plaintexts P_c, and compute $P_d = P_c \oplus \alpha$. Ask for the ciphertexts $C_c = E_{K_c}(P_c)$ and $C_d = E_{K_d}(P_d)$.
3. Set a counter C to zero.
4. For each of the M^2 choices for (P_a, P_c) (and the corresponding (P_b, P_d)):
   (a) Check whether both conditions $C_a \oplus C_c = \delta$ and $C_b \oplus C_d = \delta$ are satisfied. If yes, increase the value of the counter C by 1.
5. If $C > \mathit{Threshold}$, output "The cipher E". Otherwise, output "Random Permutation".

The value of $\mathit{Threshold}$ will be specified later in this section.

It is easy to see that for a random permutation, the probability that both conditions $C_a \oplus C_c = \delta$ and $C_b \oplus C_d = \delta$ are satisfied is 2^{-2n}. The probability that the conditions are satisfied for E is given in the following proposition:

Proposition 5. *Consider a quartet of plaintexts and their corresponding ciphertexts $((P_a, C_a), (P_b, C_b), (P_c, C_c), (P_d, C_d))$ constructed by the algorithm described above. We have*

$$\Pr\Big[(C_a \oplus C_c = \delta) \wedge (C_b \oplus C_d = \delta)\Big] \approx$$

$$\approx 2^{-n} \sum_{\beta_1 \oplus \beta_2 \oplus \gamma_1 \oplus \gamma_2 = 0} \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_1\right] \cdot \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_2\right] \cdot \Pr\left[\gamma_1 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right] \cdot \Pr\left[\gamma_2 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right]. \tag{12}$$

In particular,

$$\Pr\left[\Big(C_a \oplus C_c = \delta\Big) \wedge \Big(C_b \oplus C_d = \delta\Big)\right] \geq 2^{-n}(\hat{p}\hat{q})^2, \tag{13}$$

where

$$\hat{p} = \sqrt{\sum_{\beta'} \Pr[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta']^2}, \text{ and } \hat{q} = \sqrt{\sum_{\gamma'} \Pr[\gamma' \xrightarrow[\Delta K_{ac}]{E_1} \delta]^2}.$$

Proof. The proof is similar to the proof of Proposition 1. Consider a quartet $((P_a, C_a), (P_b, C_b), (P_c, C_c), (P_d, C_d))$ constructed by the algorithm. Denote the intermediate values $(E_0(P_a), E_0(P_b), E_0(P_c), E_0(P_d))$ (where the encryption is under the respective keys) by (X_a, X_b, X_c, X_d). For all $\beta_1, \beta_2, \gamma_1$, we say that the event $S_{\beta_1,\beta_2,\gamma_1}$ occurs, if the following conditions are satisfied:

$$X_a \oplus X_b = \beta_1, \qquad X_c \oplus X_d = \beta_2, \qquad X_a \oplus X_c = \gamma_1.$$

Since the events $\{S_{\beta_1,\beta_2,\gamma_1}\}$ for different values of $(\beta_1, \beta_2, \gamma_1)$ are disjoint and their union is the entire space, we have

$$\Pr\left[\Big(C_a \oplus C_c = \delta\Big) \wedge \Big(C_b \oplus C_d = \delta\Big)\right] =$$

$$= \sum_{\beta_1,\beta_2,\gamma_1} \Pr\left[\Big(C_a \oplus C_c = \delta\Big) \wedge \Big(C_b \oplus C_d = \delta\Big) \Big| S_{\beta_1,\beta_2,\gamma_1}\right] \cdot \Pr[S_{\beta_1,\beta_2,\gamma_1}]. \quad (14)$$

If the event $S_{\beta_1,\beta_2,\gamma_1}$ occurs, then

$$X_b \oplus X_d = (X_b \oplus X_a) \oplus (X_a \oplus X_c) \oplus (X_c \oplus X_d) =$$

$$\beta_1 \oplus \gamma_1 \oplus \beta_2.$$

Hence, by the independence assumption,

$$\Pr\left[\Big(C_a \oplus C_c = \delta\Big) \wedge \Big(C_b \oplus C_d = \delta\Big) \Big| S_{\beta_1,\beta_2,\gamma_1}\right] = \Pr\left[\gamma_1 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right] \cdot \Pr\left[\gamma_2 \xrightarrow[\Delta K_{ac}]{E_1} \delta\right], \quad (15)$$

where $\gamma_2 = \beta_1 \oplus \gamma_1 \oplus \beta_2$. Applying again the independence assumption, we have

$$\Pr[S_{\beta_1,\beta_2,\gamma_1}] = \Pr\left[X_a \in G_{K_a}^{-1}\left(\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_1\right) \Big| \left(X_c \in G_{K_c}^{-1}\left(\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_2\right)\right) \bigwedge \Big(X_a \oplus X_c = \gamma_1\Big)\right] \cdot$$

$$\cdot \Pr\left[X_c \in G_{K_c}^{-1}\left(\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_2\right) \Big| X_a \oplus X_c = \gamma_1\right] \cdot \Pr\left[X_a \oplus X_c = \gamma_1\right] =$$

$$= \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_1\right] \cdot \Pr\left[\alpha \xrightarrow[\Delta K_{ab}]{E_0} \beta_2\right] \cdot \Pr(X_a \oplus X_c = \gamma_1). \quad (16)$$

Since P_a and P_c are chosen independently, then

$$\Pr[X_a \oplus X_c = \gamma_1] \approx 2^{-n}. \quad (17)$$

Note that for any fixed value of $P_a \oplus P_c$ and γ_1, this approximation is rather inaccurate. For an ideal cipher, it is expected that for a fraction $e^{-1/2}$ of the possible values of γ_1, we have $\Pr[X_a \oplus X_c = \gamma_1] = 0$, and for the other values, the probability is at least 2^{-n+1}. However, when the probability is averaged over many different pairs (P_a, P_c), the approximation becomes reasonable.

Substituting Equations (15), (16), and (17) into Equation (14) yields Equation (12). The proof of Equation (13) given Equation (12) is identical to the derivation of Equation (3) from Equation (2) in the proof of Proposition 1.

Proposition 5 shows that if $\hat{p}\hat{q} > 2^{-n/2}$, then the probability that the conditions $(C_a \oplus C_c = \delta)$ and $(C_b \oplus C_d = \delta)$ hold simultaneously, is higher for E than for a random permutation, and hence Step 2 of the distinguisher makes sense.

The optimal choice of $Threshold$ and the computation of the success probability of the distinguisher given the probability

$$p_0 = \Pr\left[\Big(C_a \oplus C_c = \delta\Big) \wedge \Big(C_b \oplus C_d = \delta\Big)\right]$$

are very similar to the respective steps for the related-key boomerang distinguisher presented in Section 2.2, and hence are omitted here. A key recovery attack based on the related-key rectangle distinguisher is more complicated than the respective related-key boomerang attack, due to the abundance of quartets the adversary has to examine. We do not describe the key-recovery algorithm here, and refer the reader to the algorithm of the rectangle key-recovery attack in [8], that can be easily adopted to the related-key model. We note that Table 2 can also be applied to the case of the rectangle attack, with a different value for p_0, c and x: For the related-key rectangle attack $p_0 = 2^{-n}(\hat{p}\hat{q})^2$, $x = p_0/2^{-2n}$, and c is the number of expected quartets (i.e., given $M = \sqrt{c/p_0}$ pairs).

2.4 The Independence Assumptions

All statistical cryptanalytic techniques require various randomness assumptions. For example, the construction of differential characteristics uses the assumption that the cipher is a *Markov cipher* (see [6]), which implies that the characteristics of single rounds are independent of each other and can be combined. Linear cryptanalysis is based on Matsui's Piling up Lemma [31], which essentially asserts that linear approximations of single rounds are independent. These randomness assumptions allow a rigorous treatment of the techniques, as well as better applicability (since the search of differentials and linear approximations can be done for each round separately). It is easy to construct artificial examples of ciphers that do not satisfy the randomness assumptions, which would result in failure of the differential or the linear attacks. However, based on many experimental results, it is reasonable to assume that most of the ciphers satisfy the randomness assumptions. Moreover, if some cipher does not satisfy these assumptions, then this non-randomness is probably exploitable in some other attack, e.g., impossible differential attack. Nevertheless, it is important to verify the attacks experimentally whenever possible in order to assure that the assumptions indeed hold in the specific case of interest.

The randomness assumption used in the related-key boomerang and rectangle attacks (i.e., Assumption 1) has two parts. The second part of the assumption, that essentially asserts that differentials of different parts of the cipher are independent, is similar to the standard assumption that the cipher is Markovian, which is used in differential cryptanalysis. However, the first part of Assumption 1 is relatively stronger than the assumptions used in differential cryptanalysis.

Differential cryptanalysis is based on the assumption that for any fixed key K and any (related-key) differential $(\alpha \to \beta)$, the set $G_K(\alpha \to \beta)$ is distributed randomly and uniformly in the plaintext space.[14] In the related-key boomerang and rectangle attacks, the assumption deals with the distribution of *pairs of sets* of the class $G_K\left(\alpha \xrightarrow[\Delta K_0]{E_0} \beta\right)$. We assume that any two pairs of such sets are independent, i.e., the events $X \in G_K\left(\gamma_1 \xrightarrow[\Delta K_1]{E_1} \delta\right)$ and $X \oplus \beta_1 \in G_{K \oplus \Delta K_0}\left(\gamma_2 \xrightarrow[\Delta K_1]{E_1} \delta\right)$ are independent, for any value of $\gamma_1, \gamma_2, \beta_1, \delta$, and K.

To show the problem that may exist in the independence assumptions, we give the following simple artificial example, which uses high probability differentials.

Assume that for given K, α, and β, for which $MSB(\beta) = 0$ (i.e., the most significant bit of β is 0), we have $G_K^{-1}(\alpha \xrightarrow[\Delta K_0]{E_0} \beta) = \{X | MSB(X) = 1\}$ and $G_{K \oplus \Delta K_1}^{-1}(\alpha \xrightarrow[\Delta K_0]{E_0} \beta) = \{X | MSB(X) = 0\}$ (in particular, it follows that $\Pr[\alpha \xrightarrow[\Delta K_0]{E_0} \beta] = 1/2$). Further assume that for some γ such that $MSB(\gamma) = 0$ and for some δ, $\Pr[\gamma \xrightarrow[\Delta K_1]{E_1} \delta] = 1/2$. By the independence assumptions, it is expected that the probability in a related-key boomerang distinguisher based on the differentials $\left(\alpha \xrightarrow[\Delta K_0]{E_0} \beta\right)$ and $\left(\gamma \xrightarrow[\Delta K_1]{E_1} \delta\right)$ is at least $(1/4)^2 = 1/16$. However, consider a right quartet with respect to this distinguisher and denote the intermediate encryption values by (X_a, X_b, X_c, X_d). Since $X_a \in G_K^{-1}(\alpha \xrightarrow[\Delta K_0]{E_0} \beta)$, we have $MSB(X_a) = 1$, and thus, since $MSB(\gamma) = 0$, necessarily $MSB(X_c) = 1$. This implies that $X_c \notin G_{K \oplus \Delta K_1}^{-1}(\alpha \xrightarrow[\Delta K_0]{E_0} \beta)$, and thus, the actual probability of the distinguisher is zero![15]

This example demonstrates failure of the first part of Assumption 1 (independence inside the same sub-cipher). Similarly, the second part of the assumption fails if we assume that for some K, α, β, γ and δ, we have $G_K^{-1}\left(\alpha \xrightarrow[\Delta K_0]{E_0} \beta\right) = \{X | MSB(X) = 1\}$ and $G_K\left(\gamma \xrightarrow[\Delta K_1]{E_1} \delta\right) = \{X | MSB(X) = 0\}$, since in this

[14] There are cases in which this cannot be satisfied even in a regular cipher as shown in [17], where the behavior of differential characteristics with probability lower than 2^{-n} is shown to be dependent on the key. This is also the case for weak key classes, i.e., classes of keys which behave significantly different than random.

[15] Actually, the probability of the distinguisher may be higher due to differentials of the form $(\alpha \xrightarrow[\Delta K_0]{E_0} \beta')$ for $\beta' \neq \beta$. However, if there are no high-probability differentials of this form, the probability of the distinguisher is still significantly lower than the predicted value 1/16.

case X_a cannot be element in both $G_K^{-1}\left(\alpha \xrightarrow[\Delta K_0]{E_0} \beta\right)$ and $G_K\left(\gamma \xrightarrow[\Delta K_1]{E_1} \delta\right)$ simultaneously.

We note that in several specific cases, deviations from the prediction of the independence assumptions were detected in "real" ciphers. Such an example is the *ladder switch* described in [14], where *higher* probability for the related-key boomerang distinguisher is obtained using dependencies.

Luckily, in the related-key boomerang and rectangle attacks, there are several mechanisms which may overcome dependence problems. The first is the fact that in the attack we count over many differentials (all $\beta_1, \beta_2, \gamma_1, \gamma_2$ such that $\beta_1 \oplus \beta_2 \oplus \gamma_1 \oplus \gamma_2 = 0$), which ensures that even if there is a problem in some combination of differentials, it is expected that other combinations still succeed. The second one is the fact that four different keys are used (in the case $\Delta K_0 \neq \Delta K_1$),and thus, even if there is a dependence between the differentials, it is slightly countered by the different keys.

Experimental Verification of the Randomness Assumption As follows from the discussion above, in the related-key boomerang and rectangle attacks it is very important to verify the independence assumption practically in any specific case. Unfortunately, in many of the cases it is impossible due to the high complexity of the attack. Moreover, for the rectangle attack such verification is inherently impossible: the data complexity of the attack is lower-bounded by $2^{n/2}$ and infeasible for any block cipher with block size of 128 bits or more (e.g., AES). For the boomerang attack, it is sometimes possible to challenge the independence assumptions for a reduced-round variant of the attack, e.g., for a variant containing one or two rounds in each sub-cipher. However, this verification is not fully sufficient, since in the full attack, the overall probability of the distinguisher is an average taken over many possible differentials, while in the reduced-round variant only a small subset of the differentials is considered. It is possible that while the reduced-round attack does not satisfy the independence assumption, the full attack does satisfy it, since the deviations from independence for different characteristics cancel each other.

As an example of the validity of Assumption 1, we experimentally verified the related-key boomerang distinguisher on 6-round KASUMI, presented in Section 3.3. The predicted rate of experiments with at least one quartet of 86.5% was met with an 87% of the experiments showing one such quartet after 10,000 experiments, proving the validity of the analysis for 6-round KASUMI. For more details, we refer the reader to Section 3.3.

2.5 Generalizations of the Related-Key Boomerang and Rectangle Attacks

In this section we briefly present two generalizations of the basic related-key boomerang and rectangle attacks.

Using Structures of Keys The related-key differentials used in the attack are usually based on fixed *subkey* differences. If the key schedule of the attacked cipher is linear, such differences can be achieved by choosing the appropriate key difference. However, if the key schedule is nonlinear, a fixed key difference does not ensure fixed subkey differences. Instead, the adversary can apply differential cryptanalysis to the key schedule. By studying the differential properties of the key schedule, the adversary can find a key difference that leads to the required subkey differences with a relatively high probability. Then, the adversary can repeat the attack for many pairs of related-keys and expect that in one of the pairs, the required subkey differences are satisfied and the basic related-key boomerang/rectangle attack can be applied.

Furthermore, we observe that the number of keys used in the attack can be reduced by using *structures of keys*. Instead of finding a single key difference leading to the required subkey differences with a high probability, the adversary can find many such key differences (possibly with lower probabilities). Then, the adversary can use structures of keys such that each structure contains many pairs of keys corresponding to different "key characteristics", and thus reduce the number of keys required for the attack.

A concrete example of this improvement can be found in the attack on AES-192 in Section 4. The improvement uses 127 key characteristics in parallel, and succeeds to reduce the number of keys required for the attack from 256 to 64.

Generalizing the Key Relation While XOR relations are common and inherent to the majority of differential-based related-key attacks, in some cases other key relations are more suitable (either due to the environment of the attack or in order to obtain higher probabilities of the differentials). The related-key boomerang and rectangle attacks can be applied almost without a change when the XOR key relations are replaced by any relation satisfying a condition specified below.

Denote the relation between the keys K and K' by $R(K, K')$. We note that R can be any relation which is symmetric, and covers all keys. At the same time, we recall the fact that the more complex the relation R is, the applicability of the related-key attack may be affected. For example, in the basic related-key boomerang and rectangle attacks we can set $R(K, K') = K \oplus K'$.

The boomerang and rectangle attacks can be applied whenever the key relation satisfies the following condition:

$$\forall(K_a, K_b, K_c, K_d), \Big(R(K_a, K_c) = R(K_b, K_d)\Big) \Longrightarrow \Big(R(K_a, K_b) = R(K_c, K_d)\Big). \tag{18}$$

Condition (18) ensures that in each of the sub-ciphers, the same key relation is used in both differentials. For example, for XOR differences

$$(K_a \oplus K_c = K_b \oplus K_d) \Longrightarrow (K_a \oplus K_b = K_c \oplus K_d),$$

and hence the condition holds.

Condition (18) is satisfied for a wide variety of key relations, including additive differences (e.g., $R(K, K') = (K - K') \bmod 2^n$) and rotations. On the other hand, the condition does not hold if the relation used in the first sub-cipher (i.e., between (K_a, K_b) and (K_c, K_d)) and the relation used in the second sub-cipher (i.e., between (K_a, K_c) and (K_b, K_d)) are of different classes (e.g., XOR differences in the first sub-cipher and modular differences in the second sub-cipher).

We note that the basic related-key boomerang and rectangle attacks can be extended to use different values α, α' in the related-key differentials of E_0, and δ, δ' in the related-key differentials of E_1. Similarly, the attack can use different key differences $\Delta K_0, \Delta K'_0$ and $\Delta K_1, \Delta K'_1$ in the differentials of E_0 and E_1, respectively. This allows to extend Condition (18) to the following:

Proposition 6. *The related-key boomerang attack can be applied with two key relations R_1, R_2, as long as for every quadruple (K_a, K_b, K_c, K_d) the relations $R_1(K_a, K_b)$, and $R_2(K_a, K_c), R_2(K_b, K_d)$ imply the relation $R_1(K_c, K_d)$. The related-key rectangle attack can be applied if the relations $R_1(K_a, K_b), R_1(K_c, K_d)$ and $R_2(K_a, K_c)$ imply the relation $R_2(K_b, K_d)$.*

Finally, even if the condition of Proposition 6 is not satisfied, in some cases the attack can be still applied using structures of keys, as described earlier.

2.6 Comparison With Other Related-Key Attacks

For any new technique constructed as a combination of existing techniques, a natural question to ask is whether there are cases in which the combined attack is better than each of its components taken separately. In this section we briefly describe several important cases in which the related-key boomerang and rectangle attacks are expected to outperform each of their components. Concrete examples of the advantage of related-key boomerang and rectangle attacks over other attack techniques are given in Sections 3 and 4.

The main advantage of the related-key differential attacks over ordinary differential attacks is the ability of the adversary to use the subkey differences to cancel the plaintext difference in the input of one (or more) of the non-linear parts of the cipher. As a result, the adversary obtains one (or more) rounds in the differential that hold with probability 1, allowing the extension the differential by one (or more) rounds.

In the related-key boomerang and rectangle attacks, the adversary can enjoy this advantage twice, once in each of the sub-ciphers. As a result, the overall distinguisher can be extended by two (and in some cases even more) rounds. This is a significant advantage of the related-key boomerang/rectangle attack over all other differential-based related-key attacks (e.g., related-key differential attack, related-key impossible differential attack and related-key differential-linear attack) that can enjoy the advantage of the related-key model only once.

The advantage of gaining a single additional round (or two rounds) to the distinguisher is significant in ciphers in which the number of rounds is small and each round function is relatively strong. Hence, the gain of the related-key boomerang/rectangle attack is expected to be significant in ciphers like AES [32] and KASUMI [36].

Another property of the cipher required for the success of related-key boomerang and rectangle attacks is simplicity of the key schedule. The basic version of the attack is applicable only to ciphers with a linear key schedule, but using structures of keys the attack can be applied to ciphers with a nonlinear key schedule as well. However, if the key schedule of the cipher is complex enough and does not have "good" differential properties, then the number of keys required for the attack becomes infeasibly big.

Summarizing the discussion above, the related-key boomerang and rectangle attacks are expected to be successful if the attacked cipher has the following properties:

- A small number of relatively strong rounds.
- A relatively simple key schedule.

The class of ciphers satisfying these properties includes widely used ciphers such as AES [32], KASUMI [36], and IDEA [30]. These three ciphers can be indeed attacked efficiently using the related-key boomerang/rectangle attack technique.

3 Related-Key Boomerang/Rectangle Attacks on KASUMI

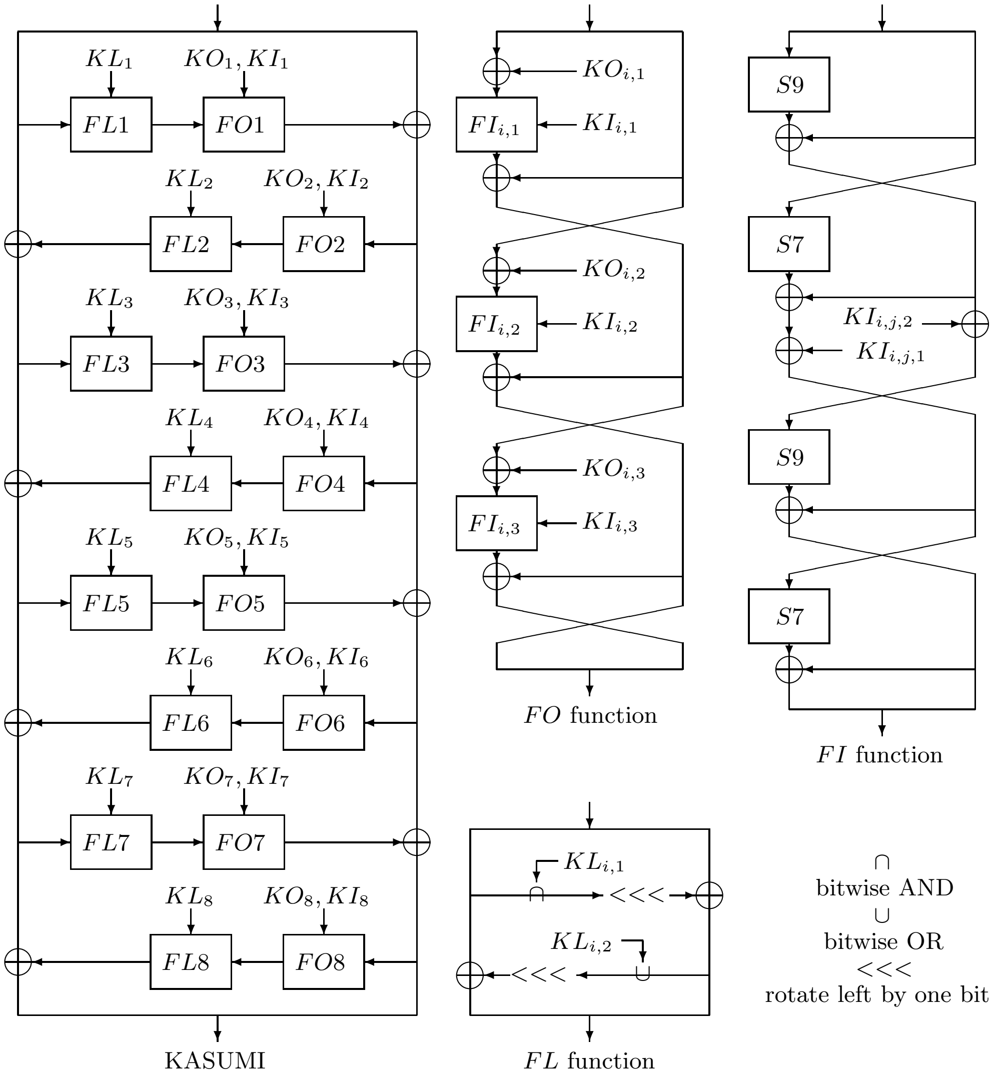
3.1 The KASUMI Block Cipher

KASUMI [36] is a 64-bit block cipher with 128-bit keys, with a recursive Feistel structure, following its ancestor, MISTY1. The cipher has eight Feistel rounds, where each round is composed of two functions: the FO function which is in itself a 3-round 32-bit Feistel construction, and the FL function that mixes a 32-bit subkey with the data in a linear way. The order of the two functions depends on the round number: in the even rounds the FL function is applied first, and in the odd rounds the FO function is applied first.

The FO function also has a recursive structure: its F-function, called FI, is a four-round Feistel construction. The FI function uses 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 accepts an additional 16-bit subkey, that is mixed with the data. In total, a 96-bit subkey enters FO in each round — 48 subkey bits are used in the FI functions and 48 subkey bits are used in the key mixing stages.

The FL function accepts a 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 Fig. 2.

Fig. 2. Outline of KASUMI



KASUMI

FO function

FI function

FL function

Table 3. KASUMI's Key Schedule Algorithm

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

The key schedule of KASUMI is very simple and the subkeys are derived from the key linearly. 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 fixed constants (we omit these from the paper as they have no effect on our results). We denote the bits of the subkeys by $K_i = (K_i^{15}, K_i^{14}, \ldots, K_i^0)$, where K_i^{15} is the most significant bit.

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 linearly dependent on the secret key in a very simple way. We give the key schedule algorithm of KASUMI in Table 3.

3.2 Related-Key Differentials of KASUMI

In our attacks we use three related-key differentials: a 4-round differential for rounds 1–4, and 3-round differentials for rounds 4–6 and rounds 5–7. Note that the change in the order between FO and FL requires to use two 3-round differentials.

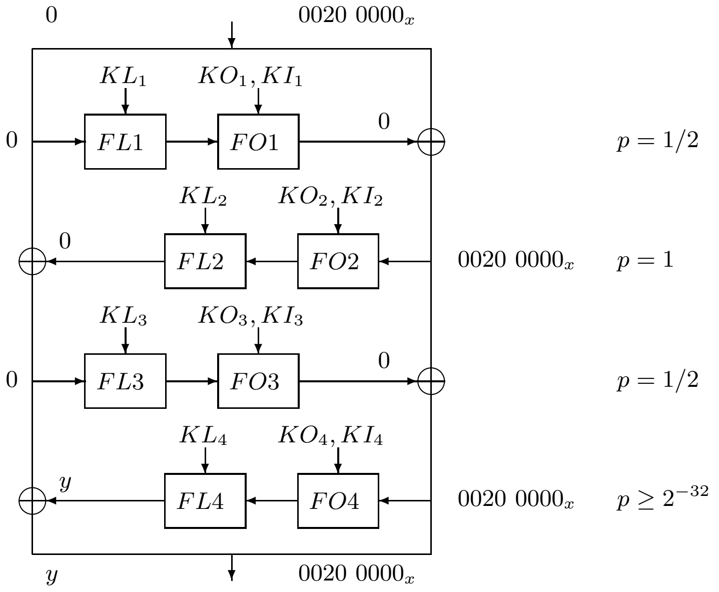
A 4-Round Related-Key Differential for Rounds 1–4 Our attack on the full KASUMI uses a related-key differential of rounds 1–4 of KASUMI which is an extension by one round of the related-key differential presented in [15]. The input difference of this differential is $\alpha = (0_x, 0020\ 0000_x)$, where 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 $\Delta K_3 = 0001_x$. The first three rounds of the characteristic have probability 1/4, and due to the Feistel structure, the α difference can propagate to at most 2^{32} differences after round 4. Hence, by Proposition 4, we have

$$\hat{p} \geq \frac{1}{4} \cdot \sqrt{2^{-32}} = 2^{-18}.$$

We outline the differential in Figure 3.

It was further observed in [15] that the probability of the differential can be increased by controlling two plaintext bits. If the adversary assigns one bit of

Fig. 3. 4-Round Related-Key Differential Characteristic of KASUMI



the plaintexts to be one (thus fixing one bit of the output of the OR operation in $FL1$) and one bit of the plaintexts to be zero (thus fixing one bit of the output of the AND operation in $FL1$), then the probability of the differential described in [15] is increased to 1/2. As a result, for our 4-round differential we have $\hat{p} \geq 2^{-17}$.[16]

We note that it is possible to rotate all the words of the key difference ΔK_{ab} and of the differential by the same number of bits, without changing the probability of the differential. Hence, the above differential can be replaced by 15 equivalent differentials.

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 [15] shifted by four rounds and rotated by one position to the right. The key difference is $\Delta K_{ac} = (0, 0, 0, 0, 0, 0, 8000_x, 0)$, and the data differences are $\gamma = (0_x, 0010\ 0000_x) \rightarrow (0_x, 0010\ 0000_x) = \delta$. Since we use a single differential (and not count over other possibilities), we have

$$\hat{q} = q = 1/4.$$

As before, it is possible to obtain 15 equivalent differentials by rotating the key difference in ΔK_7 and rotating the data differences correspondingly.

[16] Note that if the differential is used in the backward direction, the lower bound remains 2^{-18}.

3-Round Related-Key Differential for Rounds 4–6. In rounds 4–6 we use conditional related-key differential characteristics [3], i.e., characteristics that depend on some unknown key bit.

Let $\delta_0 = (0010\ 0000_x, 0_x)$, $\delta_1 = (0010\ 0040_x, 0_x)$, and $\delta' = (0001\ 0000_x, 0_x)$. If $K_5^4 = 0$ (i.e., the fifth least significant bit of K_5 equals zero), we use the two differentials $\delta_0 \rightarrow \delta_0$ and $\delta_0 \oplus \delta' \rightarrow \delta_0$. If $K_5^4 = 1$, we use the differentials $\delta_1 \rightarrow \delta_1$ and $\delta_1 \oplus \delta' \rightarrow \delta_1$. The key difference of all the characteristics is $\Delta K_{ac} = (0, 0, 0, 0, 0, 1, 0, 0)$. Each of the four characteristics has probability 1/4, if K_5^4 has the corresponding value.

For example, we describe the difference propagation in the backward direction of the characteristics $\delta_0 \rightarrow \delta_0$ and $\delta_0 \oplus \delta' \rightarrow \delta_0$. Consider a pair with ciphertext difference $\delta_0 = (0010\ 0000_x, 0_x)$. In round 6 the zero difference is preserved with probability 1/2 (i.e., the key difference is cancelled with probability 1/2). In round 5, we need a difference of $0010\ 0000_x$ after $FL5$, which is then cancelled with the key difference in $KO_{5,1}$. If $K_5^4 = 0$, then this is indeed the case with probability 1. In round 4, the zero difference is preserved by the $FO4$ function. As in round 6, it has probability 1/2 to be preserved also by $FL4$, and probability 1/2 to evolve into $\delta_0 \oplus \delta'$. Thus, the input difference of the differential characteristic is either δ_0 or $\delta_0 \oplus \delta'$, with probability 1/4 each.

In the attack, we apply the distinguisher twice, once with each pair of characteristics, and expect that in one of the applications, both differentials hold with probability 1/4.[17] For that application, we have

$$\hat{q} = \sqrt{(1/4)^2 + (1/4)^2} = 1/\sqrt{8}.$$

We note that the four conditional differential characteristics we use can be rotated along with the key difference, to produce 15 similar sets of differential characteristics with the same probabilities.

3.3 Related-Key Boomerang Distinguisher on 6-Round KASUMI

In this section we present a related-key boomerang distinguisher for 6-round KASUMI. The distinguisher we present applies to rounds 1–6 of KASUMI, but it can be easily adapted to rounds 2–7 or 3–8, as well. Let E_0 be rounds 1–3, and let E_1 be rounds 4–6. In E_0 we use the differential $\alpha = (0_x, 0020\ 0000_x) \rightarrow (0_x, 0020\ 0000_x)$ with key difference $\Delta K_{ab} = (0, 0, 1, 0, 0, 0, 0, 0)$. As shown in Section 3.2, the probability of the differential in the forward direction is 1/2 (after adding constraints on the plaintexts), and the probability in the backward direction is 1/4. In E_1 we use the two pairs of differentials $(\delta_0 \rightarrow \delta_0, \delta_0 \oplus \delta' \rightarrow \delta_0)$, and $(\delta_1 \rightarrow \delta_1, \delta_1 \oplus \delta' \rightarrow \delta_1)$, both with key difference $\Delta K_{ac} = (0, 0, 0, 0, 0, 1, 0, 0)$. As shown in Section 3.2, one of the pairs of differentials yields overall probability of $\hat{q} = 1/\sqrt{8}$ (where the "successful" pair depends on the value of the key bit K_5^4).

[17] We note that the knowledge of the "successful" pair of characteristics reveals the value of the key bit K_5^4.

The attack essentially performs two standard related-key boomerang distinguishers, one for each possible value of the key bit K_5^4. To reduce the data complexity of the attack, we share some of the chosen plaintexts between the two distinguishers. The attack algorithm requires four keys:

$$K_a;\ K_b = K_a \oplus \Delta K_{ab};\ K_c = K_a \oplus \Delta K_{ac};\ K_d = K_b \oplus \Delta K_{ac}.$$

The algorithm of the distinguisher is as follows:

1. Choose M pairs of plaintexts $(P_{a,i}, P_{b,i})$ (for $1 \leq i \leq M$) such that $P_{a,i} \oplus P_{b,i} = \alpha$. For each pair, ask for the encryption of $P_{a,i}$ and $P_{b,i}$ under the keys K_a and K_b, respectively, and denote the corresponding ciphertexts by $C_{a,i}$ and $C_{b,i}$.
2. For $1 \leq i \leq M$, calculate $C_{c,i} = C_{a,i} \oplus \delta_0$ and $C_{d,i} = C_{b,i} \oplus \delta_0$. For all i, ask for the decryption of $C_{c,i}$ and $C_{d,i}$ under the keys K_c and K_d, respectively, and denote the corresponding plaintexts by $P_{c,i}$ and $P_{d,i}$.
3. For $1 \leq i \leq M$, calculate $C_{e,i} = C_{a,i} \oplus \delta_1$ and $C_{f,i} = C_{b,i} \oplus \delta_1$. For all i, ask for the decryption of $C_{e,i}$ and $C_{f,i}$ under the keys K_c and K_d, respectively, and denote the corresponding plaintexts by $P_{e,i}$ and $P_{f,i}$.
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[18] $(1/2) \cdot (1/4) \cdot (1/\sqrt{8})^2 = 1/64$, either for quartets counted in Step 4 or for quartets counted in Step 5. Therefore, for $M = 128$ we expect to find two right quartets in Step 4 or in Step 5 (either for the quartets $(P_{a,i}, P_{b,i}, P_{c,i}, P_{d,i})$ or for the quartets $(P_{a,i}, P_{b,i}, P_{e,i}, P_{f,i})$). The right quartets can be detected effectively as for a random cipher 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}.

We note that if two right quartets are expected, then the probability that none is found is about 14%, i.e., the success rate of the attack is about $(86\% + 100\%)/2 = 93\%$. The data complexity is $3 \cdot 128 \cdot 2 = 768$ 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.

We verified the distinguishing attack experimentally. We sampled 10,000 random keys, and ran the above distinguisher with $M = 128$. By the analysis presented above, we expected that in 86.5% of the experiments there will be at least one right quartet. Our experiments revealed that in 87% there was at least one such quartet. We outline in Table 4 the number of quartets suggested in each experiments and compare it with the expected number based on Poisson distribution with a mean of 2. As can be seen from the table, the figures seem to be highly correlated.

[18] Recall that the first differential has probability 1/2 for the pair (P_a, P_b) due to fixing the plaintexts correctly.

Table 4. The Number of Found Quartets in 10,000 Experiments

Quartets	0	1	2	3	4	5	6	7	8	9	10
Experiments	1302	2695	2692	1879	907	348	127	27	9	4	0
Poisson (mean = 2)	1353	2707	2707	1804	902	361	120	34	9	2	< 1

As noted in [10], this distinguisher can be transformed into a key recovery attack. The key recover attack has a total data and time complexities of 2^{13}. The number of keys used in the attack is 34.

3.4 The Basic Related-Key Rectangle Attack on the Full KASUMI

Our attack on the full KASUMI applies the related-key rectangle distinguisher in rounds 1–7 and retrieves subkey material in round 8. Let $\Delta K_{ab} = (0,0,1,0,0,0,0,0)$ and $\Delta K_{ac} = (0,0,0,0,0,8000_x,0,0)$, and 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 we want to retrieve. In rounds 1–4 we use the related-key differential presented in Section 3.2 that has an input difference $\alpha = (0_x, 0020\,0000_x)$, a key difference ΔK_{ab} and for which $\hat{p} = 2^{-17}$.[19] In rounds 5–7 we use the related-key differential presented in Section 3.2 that has an output difference $\delta = (0_x, 0010\,0000_x)$, a key difference ΔK_{ac} and for which $\hat{q} = 2^{-2}$.[20]

We start with $N = 2^{51}$ pairs of plaintexts encrypted under K_a and K_b, and the same number of plaintext pairs encrypted under K_c and K_d. This data set contains $N^2 = 2^{102}$ quartets, of which about $N^2 \cdot 2^{-64} \cdot 2^{-34} \cdot 2^{-4} = 2^{102} \cdot 2^{-102} = 1$ are expected to be right rectangle quartets. In the attack we identify the candidate quartets out of all possible quartets, and then analyze them to retrieve the subkey of round 8.

Denote the 64-bit plaintext P by (P_L, P_R), where each 32-bit half is composed of two 16-bit halves, i.e., $P = ((P_{LL}, P_{LR}), (P_{RL}, P_{RR}))$. The attack algorithm is as follows:

1. **Data Collection Phase:**
   (a) Choose a structure of 2^{51} pairs of plaintexts (P_a, P_b), where $P_b = P_a \oplus \alpha$, $P^0_{a_{LL}} = 0$ (i.e., the least significant bit of $P_{a_{LL}}$ is fixed to zero for all the plaintexts in the structure), and $P^1_{a_{LR}} = 1$.[21] For each pair, ask for the encryption of P_a and P_b under the keys K_a and K_b, respectively, and insert each pair of ciphertexts into a hash table indexed by the 64-bit value of $(C_{a_{RL}}, C_{a_{RR}}, C_{b_{RL}}, C_{b_{RR}})$.

[19] In this rectangle attack the differentials are used only in the forward direction and hence $\hat{p}$ is 2^{-17} rather than 2^{-18}, as shown in Section 3.2.

[20] We note that we picked a slightly rotated version of δ_0 to ensure maximal independence between the two sub-ciphers, thus validating the independence assumptions.

[21] Fixing a bit in P_a also fixes the corresponding bit in its counterpart P_b, due to the difference α.

Table 5. Possible Values of $KL_{8,2}$ and $KL_{8,1}$

OR — $KL_{8,2}$

	(X'_{bd}, Y'_{bd})			
(X'_{ac}, Y'_{ac})	(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'_{bd}, Y'_{bd})			
(X'_{ac}, Y'_{ac})	(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 denoted by (input difference, output difference): (X'_1, Y'_1) for one pair and (X'_2, Y'_2) for the other pair.

(b) Choose a structure of 2^{51} pairs of plaintexts (P_c, P_d), where $P_d = P_c \oplus \alpha$, $P^0_{c_{LL}} = 0$, and $P^1_{c_{LR}} = 1$. For each pair, ask for the encryption of P_c and P_d under the keys K_c and K_d, respectively. Then, access the hash table in the entry corresponding to the value $(C_{c_{RL}} \oplus 0010_x, C_{c_{RR}}, C_{d_{RL}} \oplus 0010_x, C_{d_{RR}})$. 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).

In the first step described above, the $(2^{51})^2 = 2^{102}$ possible quartets are filtered according to a condition on the 64 bits of difference which are known (due to the output difference δ), which leaves about 2^{38} quartets to Step 2. In this step, we treat all the remaining quartets as right quartets. Under this assumption, we know not only the actual inputs to round 8, but also the output differences. We guess 32 bits of the key $(KO_{8,1}, KI_{8,1})$, and try to deduce $KL_{8,2}$.

2. **Analyzing Quartets:**
   (a) For each remaining quartet (C_a, C_b, C_c, C_d) guess the 32-bit value of $KO_{8,1}$ and $KI_{8,1}$. 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 5. On average $(8/16)^{16} = 2^{-16}$ values of $KL_{8,2}$ are suggested by each quartet and guess of $KO_{8,1}$ and $KI_{8,1}$.
   (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 5. On average $(8/16)^{16} = 2^{-16}$ values of $KL_{8,1}$ are suggested by each quartet and guess of $KO_{8,1}, KI_{8,1}, KO_{8,3}$, $KI_{8,3}$, and the computed value of $KL_{8,2}$.
3. **Finding the Right Key:** For each quartet and value of ($KO_{8,1}$, $KI_{8,1}$, $KO_{8,3}$, $KI_{8,3}$, $KL_{8,1}$,$KL_{8,2}$) suggested in Step 2, guess the remaining 32 bits of the key, and perform a trial encryption.

3.5 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}.

Let us examine a difference in some bit j. For each pair, there are four combinations of input difference and output difference in this bit. Table 5 lists the values that the two pairs suggest for the respective key bit.

In the table there are nine entries that contain no value, which means a contradiction. For example, a difference 0 can never lead to a difference 1 by any linear function. Another possible contradiction occurs 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 suggestions for the key bit is 8. Since the table has 16 entries, the average number of suggested values for the key bit is 1/2. In total, for the 16 bits there are $(1/2)^{16} = 2^{-16}$ key suggestions on average. A similar analysis can be applied to Step 2(b).

We note that the identification of suggested values (or of the found contradictions) can be done efficiently in a bit-sliced manner. Hence, we conclude that this step can be implemented efficiently (for each quartet and initial subkey guess).

Step 2 starts with 2^{38} quartets. In Step 2(a), the $2^{38} \cdot 2^{32} = 2^{70}$ (quartet, subkey guesses) tuples suggest $2^{70} \cdot 2^{-16} = 2^{54}$ values for the 48 subkey bits $(KO_{8,1}, KI_{8,1}, KL_{8,2})$. Similarly, after the additional guess of $KO_{8,3}$ and $KI_{8,3}$ in Step 2(b), we get $2^{54} \cdot 2^{32} \cdot 2^{-16} = 2^{70}$ suggestions for the 96 subkey bits $(KO_{8,1}, KI_{8,1}, KL_{8,2}, KO_{8,3}, KI_{8,3}, KL_{8,1})$.

Step 3 goes over all 2^{70} suggestions for the 96 key bits, and tries to complete the remaining 32 key bits by an exhaustive search. This can be performed easily 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.

3.6 Improvements of the Attack

In this section we present several improvements of the attack that allow to decrease its time complexity considerably.

Improvement of Step 3 Step 3 can be improved by using key ranking techniques. Taking $2^{52.6}$ plaintexts encrypted under four different keys (i.e., three times the data as before), we expect nine right quartets. Instead of completing the missing key bits by an exhaustive key search, we count how many (quartet, subkey guess) 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 a 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. After this modification, the time complexity of

Step 3 becomes negligible compared to that of Step 2(b). This reduces the time complexity of the attack to $2^{86.2}$ full KASUMI encryptions, while increasing the data complexity to $2^{54.6}$ related-key chosen plaintexts.

First Improvement of Step 2(b) Another improvement of the attack is based on the observation that Step 2(b) can be implemented in two sub-steps. In the first one, we guess $KO_{8,3}$ and the 9-bit subkey $KI_{8,3,2}$, 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 guess) tuples where the subkey guess is of 73 bits. As this improvement deals only with 9 bits of $KL_{8,1}$, the expected number of remaining (quartet, subkey guess) values is $2^{73.2}$. Then, in the second sub-step we guess the 7 bits of $KI_{8,3,1}$ and find the value of the 7 remaining bits of $KL_{8,1}$. The time complexity of the attack is now dominated by the first sub-step of Step 2(b), whose complexity is equivalent to about $2^{79.2}$ KASUMI encryptions.

Second Improvement of Step 2(b) Our next improvement uses the fact that Step 2(b) depends only partially on Step 2(a). After Step 2(a) there are 2^{54} tuples of the form (quartet, subkey guess), where the subkey guess is of 48 bits. However, Step 2(b) uses only 32 bits of the guessed subkey, namely, 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 only $2^{12.9}$ values for the 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.

Third Improvement of Step 2(b) Finally, we offer another improvement that is based on a more delicate attack procedure. We recall that for a random input/output difference to an S-box, there is on average one pair of actual values which fit these differences.[22]

Denote the input and output of $FL8$ by (Y_0, X_0) and (Y_1, X_1), respectively. The relation between the input and the output of $FL8$ is given by

$$X_1 = X_0 \oplus ((Y_0 \wedge KL_{8,1}) \lll 1); \; Y_1 = Y_0 \oplus ((X_1 \vee KL_{8,2}) \lll 1); \qquad (19)$$

Let (Y_0', X_0') and (Y_1', X_1') be the differences in the input and the output of $FL8$ for the considered pair. Note that the output difference is known to the adversary, and after the guess of $KO_{8,1}$ and $KI_{8,1}$, the adversary can also compute Y_0'.

[22] Actually, the number of such ordered pairs is always even. On the other hand, the probability that no pair satisfies the input/output difference constraint is at least 1/2.

In the modified variant of the attack, in Step 2(b) the adversary guesses only $KO_{8,3}$, and not $KI_{8,3}$. Like in the first improvement of Step 2(b), the step is divided into two sub-steps. The first sub-step collects suggestions for the 9-bit subkey $KI_{8,3,1}$ and the second sub-step collects suggestions for the 7-bit subkey $KI_{8,3,2}$.

The First Sub-Step: The knowledge of $KO_{8,3}$ allows the adversary to compute the input difference to the second $S9$ S-box of the function $FI_{8,3}$. Moreover, the output difference of that S-box is given by the 9 corresponding bits of X'_0. During this sub-step, we abuse the notation and refer by X'_0 and Y'_0 to these 9 bits.

When Y'_0 has $9-i$ zeros, the AND operation with the subkey $KL_{8,1}$ may affect the difference only in the i active bits, and hence can result in at most 2^i differences. According to Equation (19) each such difference Y'_0 suggests at most 2^i differences in X'_0. Each such X'_0, combined with the input difference to the second $S9$ S-box of $FI_{8,3}$, translates to one suggestion on average for the actual inputs to that S-box, that in turn translates to one candidate on average for the subkey $KI_{8,3,1}$.

Hence, assuming that the differences Y'_0 are distributed uniformly, the expected number of candidates a pair suggests for $KI_{8,3,1}$ is

$$\sum_{i=0}^{9} \binom{9}{i} \cdot 2^{-9} \cdot 2^{i} = 2^{-9} \sum_{i=0}^{9} \binom{9}{i} \cdot 2^{i} \cdot 1^{9-i} = 2^{-9} \cdot (1+2)^9$$

$$= \frac{3^9}{2^9} \approx 2^{5.3}.$$

We note that this is also the average time complexity associated with this procedure (for the first pair). Then, for the analysis of the second pair, one can either repeat the procedure, and compute the intersection of the two lists, or just try the keys offered by the first procedure.

Of course, a more efficient approach is to start with the pair that is going to suggest less keys (i.e., the one for which Y'_0 has more 0's). Repeating the analysis presented above, and taking into consideration this fact, it is expected that the pair with lower hamming weight requires $2^{4.2}$ trials on average to find the list of $2^{4.2}$ candidate subkeys for $KI_{8,3,1}$, and an equivalent time to challenge them for consistency with the second pair. At the end of the process, for a given quartet, guess of $KO_{8,1}, KI_{8,1}$ and $KO_{8,3}$, about 2.88 candidates to $KI_{8,3,1}$ are expected to remain. Of course, if no candidates remain, then it is possible to discard this (quartet, $KO_{8,3}, KI_{8,3}$) combination. We note that about 39.7% of the combinations are indeed discarded at this stage. The time complexity of this sub-step is $9 \cdot 2^{54} \cdot 2^{12.9} \cdot 2 \cdot 2^{4.2} = 2^{75.3}$ one-round encryptions (when combined with the previous improvements).

The Second Sub-Step: In this sub-step, for each of the remaining candidates, the adversary obtains suggestions for $KI_{8,3,2}$, by examining the 7 remaining bits of X'_0 and Y'_0. This time, the running time for the pair with lower hamming weight is expected to be $2^{3.2}$ evaluations on average. Hence, this step takes $9 \cdot 2^{54} \cdot (2.88 \cdot 2^{12.9}) \cdot 2 \cdot 2^{3.2} = 2^{75.9}$ one-round encryptions. The expected number of suggested candidates for the subkey $KI_{8,3,2}$ is 2.25.

After these two sub-steps, the adversary is left with $9 \cdot 2^{51.9} \cdot 2^{16} \cdot 2.88 \cdot 2.25 = 2^{72.8}$ suggestions for combinations of $(quartet, KO_{8,1}, KI_{8,1}, KO_{8,3}, KI_{8,3})$. At this point, the adversary can use the obtained information to get suggestions for the subkeys $KL_{8,2}$ and $KL_{8,1}$, along with an additional filtering. The resulting number of suggestions for $(quartet, KO_{8,1}, KI_{8,1}, KO_{8,3}, KI_{8,3}, KL_{8,1}, KL_{8,2})$ is $2^{73.2}$, like in the basic attack (actually, the number in the basic attack is 2^{70} but in the modified attack we examine 9 times more quartets). The remaining key bits can be retrieved efficiently, as was shown earlier. The time complexity of this attack, is thus dominated by the analysis of the S-boxes $S9$ and $S7$ of the function $FI_{7,3}$, which takes time of about $2^{75.9} + 2^{75.3} = 2^{76.6}$ one-round encryptions, or a total of $2^{73.6}$ full KASUMI encryptions.

3.7 A Different 8-Round Rectangle Attack

It is also possible to apply a slightly different attack algorithm, which is based on a 6-round related-key rectangle distinguisher. The distinguisher is used in rounds 2–7, where the first related-key differential (used in rounds 2–4) is the differential of rounds 4–6 presented in Section 3.2 (shifted two rounds backwards), and the second related-key differential is used in rounds 5–7. The associated probabilities are $\hat{p} = 1/\sqrt{8}$, and $\hat{q} = 1/4$, respectively. Hence, given 2^{72} quartets with the right input differences, we expect about $2^{72} \cdot (1/\sqrt{8})^2 \cdot (1/4)^2 \cdot 2^{-64} = 2$ right quartets.

The attack uses 16 structures, each composed of three sets of 2^{32} plaintexts each. The structures are of the form $P = \{A, x\}$ for a fixed A, and all possible x's which are encrypted under K_a and K_c, and $P_0 = \{A \oplus \delta_0, x\}$ and $P_1 = \{A \oplus \delta_1, x\}$, each encrypted under K_b and K_d. For sake of clarity, we describe the attack for $K_2^4 = 0$ (which means that the differentials in use are based on δ_0). The other case, follows immediately.

We search for quartets composed of $((P_a, P_b), (P_c, P_d))$ with input difference δ_0 after the first round (in the pairs (P_a, P_b) and (P_c, P_d)), and difference α before the last round (in the pairs (P_a, P_c) and (P_b, P_d)). To do so, for each pair of structures, the adversary finds all the candidate quartets (out of possible 2^{128} quartets, only 2^{64} satisfy the known ciphertext differences). As there are $16^2 = 2^8$ pairs of structures, the adversary analyzes 2^{72} quartets, in a manner very similar to the basic attack:

1. For each candidate quartet, guess $KI_{8,1}, KO_{8,1}$ and retrieve $KL_{8,2}$ (similarly to Step 2(a)).
2. For each candidate (quartet, subkey) tuple, guess $KI_{1,1}$ and use the known value of $KO_{1,1}$ (which is known from $KL_{8,2}$) to obtain candidate $KL_{1,2}$.
3. For each candidate (quartet, subkey) tuple, guess $KO_{1,3}$ and from the knowledge of $KL_{1,1}$ (which is known from $KO_{8,1}$), find candidates to $KI_{1,3}$ (apply Step 2(b) in a slightly different order).
4. Exhaustively search over all remaining keys.

The first step of the attack takes $2^{72} \cdot 2^{32} = 2^{104}$ operations, and results in $2^{72} \cdot 2^{32} \cdot 2^{-16} = 2^{88}$ tuples of (quartets, 48-bit key guess). In the second step, 16 more key bits are guessed, each resulting in a consistent suggestion for $KL_{1,2}$ with probability 2^{-16}. Hence, this step takes $2^{88} \cdot 2^{16} = 2^{104}$ operations, and offers $2^{88} \cdot 2^{16} \cdot 2^{-16} = 2^{88}$ tuples of (quartets, 80-bit key guess). In Step 3, we apply an analysis step which is similar to Step 2(b) (using the fact that we know the inputs to $FI_{1,3}$ and its output difference), which allows finding a consistent suggestion for $KI_{1,3}$ with probability 2^{-16} for each $KO_{1,3}$. Hence, also this step takes $2^{88} \cdot 2^{16} = 2^{104}$ operations, and results in $2^{88} \cdot 2^{16} \cdot 2^{-16} = 2^{88}$ tuples of the form (quartet, 112-bit subkey guess). At this point, exhaustive search takes $2^{88} \cdot 2^{16} = 2^{104}$ trial encryptions, which is what the adversary does.

We note that the attack has to be repeated twice, once with δ_0 and once with δ_1. However, when using δ_0 we are assuming that $K_3^4 = 0$, which means that there is no need to guess its value, and by reversing the order of Steps 1 and 2, reduce the total running time to 2^{104} trial encryptions. The data complexity is unchanged, i.e., $2 \cdot 16 \cdot 3 \cdot 2^{32} = 2^{38.6}$ chosen plaintexts in total.

4 Related-Key Rectangle Attacks on AES-192

4.1 The AES Block Cipher

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 Nr, i.e., $Nr = 10$, 12 and 14 when used with 128, 192 and 256-bit keys, respectively. The rounds are numbered $0, \cdots, Nr-1$. The round function of AES consists of the following four basic operations:

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

Each round of AES applies the SB, SR, MC and ARK operations in that order. Before the first round, an additional ARK operation is performed (using the whitening key), and in the last round, the MC operation is omitted. For more details of the above four transformations, we refer the reader to [16].

AES uses different key scheduling algorithms according to the length of the key. As we deal only with AES-192 (i.e., AES with 192-bit keys), we describe the key schedule for AES-192: Let the supplied key be 6 words of 32-bits $(W[0], W[1], \cdots, W[5])$. To generate 13 subkeys of 128 bits (which compose 52 words of 32 bits), the following algorithm is used:

- For $i = 6$ till $i = 51$ do the following:
  - If $i \equiv 0 \bmod 6$, then $W[i] = W[i-6] \oplus \mathrm{SB}(\mathrm{RotByte}(W[i-1])) \oplus \mathrm{Rcon}[i/6]$,
  - else $W[i] = W[i-1] \oplus W[i-6]$,

where RotByte represents one byte rotation to the left and *Rcon* denotes an array of fixed constants.

The 128-bit block of AES is represented by a 4×4 byte matrix. Throughout the paper we treat the internal state as bytes ((0,1,2,3),(4,5,6,7),(8,9,10,11),(12,13,14,15)) (see Figure 4 for a graphical representation).

Fig. 4. Byte coordinates of a 128-bit block of AES (Ri: Row i, Ci: Column i, X_i: Byte i)

	$C0$	$C1$	$C2$	$C3$
$R0$	X_0	X_4	X_8	X_{12}
$R1$	X_1	X_5	X_9	X_{13}
$R2$	X_2	X_6	X_{10}	X_{14}
$R3$	X_3	X_7	X_{11}	X_{15}

4.2 Preliminaries

We first define some notation which is used in our attacks on AES. Denote the 10 first 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 (starting from the AddRoundKey operation of round 0), E^1 is rounds 5–8 and E^f is round 9. In our 10-round AES-192 attack, we use the related-key differential for E^0 depicted in Fig. 5 and the related-key differential for E^1 depicted in Fig. 6. We use these related-key differentials for constructing a related-key rectangle distinguisher for $E^1 \circ E^0$, that allows us to recover some portion of the keys in E^b and E^f.

Let K_a, K_b, K_c, K_d be a quartet of keys satisfying the subkey differences required for the related-key differential (or that we conjecture that they satisfy these subkey differences). Then K_x^w is the whitening key derived from K_x and K_x^i is the ith round subkey derived from K_x. We use the notation P_x to denote a plaintext encrypted under K_x. The intermediate value in the encryption of P_x is denoted by I_x^i (the input to round i).

Besides the key differences ΔK_{ab} and ΔK_{ac}, we use $\Delta I_{ab}^i = I_a^i \oplus I_b^i = I_c^i \oplus I_d^i$ and $\Delta I_{ac}^i = I_a^i \oplus I_c^i = I_b^i \oplus I_d^i$. Finally, $HW_b(X)$ denotes the hamming weight in bytes of X, x denotes an 8-bit difference, y, z denote (not necessarily different) 8-bit differences such that x can evolve to y or z through the SubBytes operation, and $*$ denotes an unknown byte difference.

Fig. 5. The related-key differential for E_0 (ARK of round 0 and rounds 1-4), and the preceding E_b (the whitening ARK and most of round 0). ΔP_{ab} denotes a set of differences.

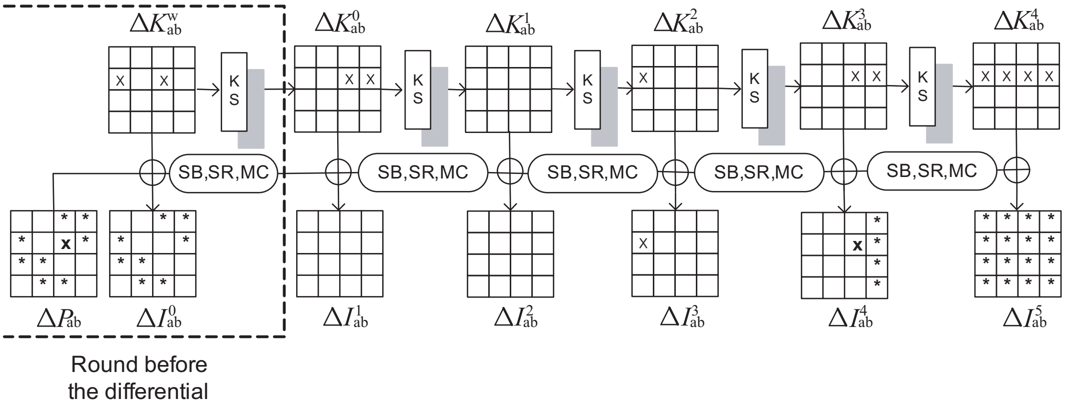
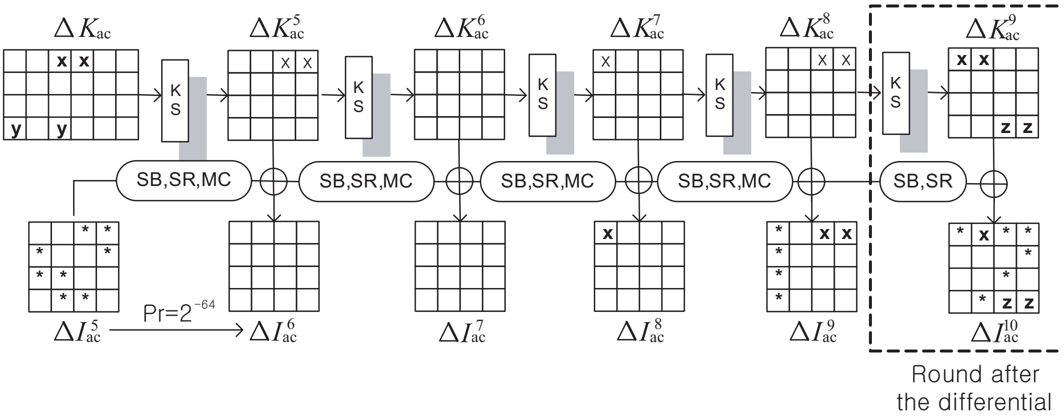


Fig. 6. The related-key differential for rounds 5-8 (E^1) and the following round (E^f).



4.3 8-Round Related-Key Rectangle Distinguisher

Our related-key differentials exploit the slow difference propagation of the key schedule of AES-192, that allows three consecutive rounds for which the Hamming weight in bytes of the key differences is 2,0,1, respectively, as shown in Figure 6.

The differential used for E^0 is depicted in Figure 5. Its key difference ΔK_{ab} equals x in bytes 1 of $W[0]$ and $W[2]$, and is zero in all the other bytes (see Figure 6). The input difference α equals x in bytes $X_{9,13}$ and zero in the rest of the bytes, such that it cancels with the subkey difference, and the input difference to round 1 becomes $\Delta I_{ab}^1 = 0$. Since there are at most 2^{39} possible output differences, it follows from Proposition 3 that $\hat{p} \geq \sqrt{2^{-39}} = 2^{-19.5}$.

The second differential, depicted in Figure 6, is a truncated differential. Its output difference set ΔI_{ac}^9 consists of 127 possible output differences that share all but the first column. The first column of ΔI_{ac}^9 can accept any of the values

$$\begin{aligned}\mathcal{B} = \{\mathrm{MC}(j,0,0,0) \mid j = \mathrm{SB}(i) \oplus \mathrm{SB}(x \oplus i), \\ i = 0,1,2,\cdots,255\}.\end{aligned}$$

As for the key difference, it appears that the required subkey differences (presented in Figure 6) cannot be assured by any fixed key difference. The subkey difference pattern requires some cancellation (in byte 11 of ΔK_{ac}^3), that occurs with probability 2^{-7}. Hence, this differential can be interpreted as a weak key class or a conditional differential. Fortunately, the relation can be assured with a small set of keys.

In order to compute $\hat{q}$, we take all the possible input differences ΔI_{ac}^5 that can lead after SubBytes, ShiftRows, and MixColumns operations to a state with difference x in bytes $X_{8,12}$ and zero difference in the rest of the bytes. Such difference is then canceled with the key difference ΔK_{ac}^5 and leads to zero difference ΔI_{ac}^6. There are 127^8 such input differences, one of them with probability $(2^{-6})^8 = 2^{-48}$, $8 \cdot 126$ of them with probability $(2^{-6})^7 \cdot 2^{-7} = 2^{-49}$, and so forth until $(126)^8$ of them with probability $(2^{-7})^8 = 2^{-56}$. Summing over all of them yields $\hat{q} \geq 2^{-27.9}$.

Therefore, the overall probability of the rectangle distinguisher (i.e., $\Pr[I_a^9 \oplus I_c^9, I_b^9 \oplus I_d^9 \in \Delta I_{ac}^9]$) is

$$2^{-128} \cdot (2^{-19.5})^2 \cdot (2^{-27.9})^2 = 2^{-222.8}.$$

We note that since ΔI_{ac}^9 consists of 127 differences, the probability that the condition $[I_a^9 \oplus I_c^9, I_b^9 \oplus I_d^9 \in \Delta I_{ac}^9]$ holds for a random permutation is approximately 2^{-242}.

4.4 Key Recovery Attack on 10-Round AES-192 with 256 Related Keys

Before we present the attack, we address two points arising from the related-key nature of the attack.

Table 6. The development of the subkey differences

ΔK_{ab}^{w}				ΔK_{ab}^{0}				ΔK_{ab}^{1}				ΔK_{ab}^{2}				ΔK_{ab}^{3}				ΔK_{ab}^{4}			
x		x				x	x					x						x	x	x	x	x	x

a	a	a	a	a	a	a		a		a		c	c	d	d	c	c	c	
x		x		x		x	x			x	x	x				x		f	f
												e	e	e	e	e	e	e	
						b	b	b	b	b	b	b		b		b		g	g
ΔK_{ab}^{5}				ΔK_{ab}^{6}				ΔK_{ab}^{7}				ΔK_{ab}^{8}				ΔK_{ab}^{9}			

The development of the subkey differences for ΔK_{ab}

ΔK_{ac}^{w}				ΔK_{ac}^{0}				ΔK_{ac}^{1}				ΔK_{ac}^{2}				ΔK_{ac}^{3}				ΔK_{ac}^{4}			
		x	x					x						x	x	x	x			x		x	
y		y				y	y					y											

		x	x					x						x	x	x	x		
																		z	z
ΔK_{ac}^{5}				ΔK_{ac}^{6}				ΔK_{ac}^{7}				ΔK_{ac}^{8}				ΔK_{ac}^{9}			

The development of the subkey differences for ΔK_{ac}

Multiple Keys. As noted before, the attack requires a quartet of keys satisfying the subkey differences depicted in Figure 6. Since no single key difference can assure these subkey differences, the simplest solution is to repeat the attack for all 127 possible byte values y such that difference x can evolve to y through the SubBytes operation. For each such y, the attack is performed with the same keys K_a and K_b, and with different keys K_c and K_d that correspond to y. Thus, the total number of required related keys is 256 (a single pair (K_a, K_b), and 127 pairs $(\widetilde{K}_c, \widetilde{K}_d)$). Note that since we try all possible values for y, one of the quartets we consider necessarily satisfies the subkey differences.

The Amount of Guessed Subkey Material. The attack recovers bytes 1, 2, 6, 7, 8, 11, 12, 13 of the whitening key quartet $(K_a^w, K_b^w, K_c^w, K_d^w)$, and bytes 0,4,6,7,8,10,12,13 of the subkey quartet $(K_a^9, K_b^9, K_c^9, K_d^9)$. While there are exactly $(2^8)^8 = 2^{64}$ possible values for the eight guessed bytes in the whitening keys, the number of possible values of the six bytes of K^9 is much higher than $(2^8)^8 = 2^{64}$. This occurs as the subkey difference ΔK_{ab}^9 in four of these bytes is unknown to the adversary (these are bytes 0,8,10,13, whose differences are denoted in Figure 6 by c, c, e, f, respectively). As a result, the adversary has to learn not only six bytes of K_a^9, but also the values c, e, f. Due to properties of the key schedule, given the value of e, there are only 127 possible values of f. Hence, the number of possible guesses in K^9 is $(2^8)^8 \cdot (2^8)^2 \cdot 2^7 = 2^{87}$.

The attack algorithm goes as follows:

Data Collection Phase (Steps 1–3):

1. Choose $2^{49.2}$ structures $S_a^1,\ S_a^2, \cdots,\ S_a^{2^{49.2}}$ of 2^{64} plaintexts each, where in each structure the values of bytes 0, 3, 4, 5, 9, 10, 14, 15 are fixed and the remaining eight bytes assume all the possible values, and ask for their encryption under the key K_a.
2. Compute $2^{49.2}$ structures $S_b^1,\ S_b^2, \cdots,\ S_b^{2^{49.2}}$ of 2^{64} plaintexts each by XORing byte 9 of all the plaintexts in $S_a^1,\ S_a^2, \cdots,\ S_a^{2^{49.2}}$ with x. Ask for the encryption of these structures under the key K_b.
3. Guess a candidate for the difference y and compute $\Delta\widetilde{K}_{ac}$. For the key difference $\Delta\widetilde{K}_{ac}$, do the following:
   (a) Choose $2^{49.2}$ structures $S_c^1,\ S_c^2, \cdots,\ S_c^{2^{49.2}}$ of 2^{64} plaintexts each, constructed similarly to S_a^j (possibly with different constant values), and ask for their encryption under the key $\widetilde{K}_c = K_a \oplus \Delta\widetilde{K}_{ac}$.
   (b) Compute $2^{49.2}$ structures $S_d^1,\ S_d^2, \cdots,\ S_d^{2^{49.2}}$ of 2^{64} plaintexts each by XORing byte 9 of all the plaintexts in $S_c^1,\ S_c^2, \cdots,\ S_c^{2^{49.2}}$ with x. Ask for the encryption of these structures under the key $\widetilde{K}_d = K_b \oplus \Delta\widetilde{K}_{ac}$.

Analyzing E^b and Finding Candidate Quartets (Step 4):

4. Guess the 64 bits of bytes 1, 2, 6, 7, 8, 11, 12, 13 of K_a^w and derive from them the corresponding bytes in K_b^w, K_c^w, K_d^w. For each such guess:

(a) Partially encrypt all the plaintexts through the 8 active S-boxes of round 0, and find all pairs (P_a, P_b) with difference[23] α just before the AddRoundKey of round 0 (where P_x is encrypted under K_x). Denote the corresponding ciphertexts by (C_a, C_b) respectively. Each pair of structures (S_a^j, S_b^j) is expected to contain 2^{64} such pairs (C_a, C_b), and hence the total number of pairs at this stage is $2^{49.2} \cdot 2^{64} = 2^{113.2}$.

(b) Insert all pairs (C_a, C_b) into a hash table (indexed by bytes 1, 2, 3, 4, 5, 6, 9, 14 of each of the ciphertexts).

(c) Similarly, find all pairs (P_c, P_d) with difference α just before the AddRoundKey of round 0 (P_x is encrypted under $\widetilde{K}_x$). Denote their corresponding ciphertexts by (C_c, C_d), respectively. As before, the expected number of pairs (C_c, C_d) at this stage is $2^{113.2}$.

(d) For every pair (C_c, C_d) check whether there exists a pair (C_a, C_b) such that $C_a \oplus C_c$ and $C_b \oplus C_d$ are zero in bytes 1,2,3,5,6,9,14, and x in byte 4. For each such quartet, check that the difference $C_a \oplus C_c$ is the same in bytes 11 and 15 (denoted by z_{ac}), and check the same for $C_b \oplus C_d$ (where the difference is denoted by z_{bd}). Check that x input difference to the S-box may cause z_{ac} and z_{bd} output differences (otherwise, discard the quartet). Starting with $(2^{113.2})^2 = 2^{226.4}$ quartets, about $2^{226.4} \cdot 2^{-112} = 2^{114.4}$ quartets satisfy the zero differences in bytes 1,2,3,5,6,9,14, of which $2^{114.4} \cdot 2^{-16} = 2^{98.4}$ satisfy the x differences in byte 4, and amongst them $2^{98.4} \cdot (2^{-8})^2 \cdot (127/256)^2 = 2^{80.4}$ quartets remain and are further analyzed.

Analyzing E^f (Steps 5–8):

5. Using the difference in byte 12 of ΔI_{ac}^{10} and ΔI_{bd}^{10}, deduce the subkey suggested by the pairs (C_a, C_c) and (C_b, C_d) for byte 12 of $K_a^9, K_b^9, K_c^9, K_d^9$ (due to the subkey differences, the value is the same for these four subkeys).[24] If the values disagree, discard the quartet. Of the $2^{80.4}$ quartets entering Step 5, we expect $2^{80.4} \cdot 2^{-8} = 2^{72.4}$ quartets to remain, each suggesting one value for subkey byte 12.
6. For each remaining quartet, consider the pairs (C_a, C_c) and (C_b, C_d) separately and use the differences z_{ac}, z_{bd} to deduce the value of byte 4 in K_a^9, K_b^9, K_c^9, K_d^9.[25] Deduce the value of c from byte 4 of the difference $K_a^9 \oplus K_b^9$.

[23] The difference α is defined in Section 4.3.

[24] Recall that a $(\Delta_{IN}, \Delta_{OUT})$ pair for the SubBytes operation yields on average one suggestion for the actual inputs/outputs. In our case, the output difference is known, and we assume that the input difference is x. This gives us one suggestion on average for the actual outputs of the SubBytes operation, that in turn yields a single suggestion on average for byte 12 of the last subkey.

[25] For each pair, we consider the generation of byte 11 in K^9 in the key schedule algorithm. At this stage, the input difference to the SubBytes operation in the key schedule algorithm is x, and the output difference is z_{ac} (or z_{bd} for the second pair). This suggests one value on average for the input of the SubBytes operation, i.e., to byte 4 of K^9.

7. For each quartet, consider the pairs (C_a, C_c) and (C_b, C_d) separately, and for each of them use the difference in byte 8 of $\Delta(I^{10})$ to deduce the values suggested by the pairs (C_a, C_c) and (C_b, C_d) for byte 8 of $K^9_a, K^9_b, K^9_c, K^9_d$. Use the value of byte 8 of the difference $K^9_a \oplus K^9_b$ to get a suggestion for c. If the suggestion disagrees with the value of c obtained for the quartet in Step 6, discard the quartet. Consistent values are expected with probability 2^{-8}, and about $2^{72.4} \cdot 2^{-8} = 2^{64.4}$ quartets remain. Each of the remaining quartets suggests on average one value for bytes $4, 8, 12$ of $K^9_a, K^9_b, K^9_c, K^9_d$, and for c.
8. For each remaining (quartet, subkey guess):
   (a) Guess byte 0 of K^9_a (that along with the knowledge of c is sufficient to compute byte 0 of K^9_b, K^9_c, K^9_d), and partially decrypt the two pairs to find the value of byte 0 in the input of round 9.
   (b) For each pair, use the knowledge of byte 0 in ΔI^9_{ac} (or ΔI^9_{bd}) to retrieve the entire difference ΔI^9_{ac} (or ΔI^9_{bd}, respectively). This is possible since the differences in the other bytes of the first column depend linearly on the difference in byte 0.
   (c) For each pair, use the input and output difference of the SubBytes operation in byte 3 of round 9 to find the value of byte 7 in K^9. If the pairs of the quartet disagree on that value, discard the (quartet,subkey guess). For each guess of byte 0, 2^{-8} of the quartets are expected to suggest consistent values, and hence on average, each quartet suggests a single value for bytes 0,7 of K^9.
   (d) For each pair, use the input and output differences of the SubBytes operation in bytes 1,2 of round 9 to find the value of bytes 10,13 in K^9. Use the difference in bytes 10,13 of $K^9_a \oplus K^9_b$ to retrieve the value of e, f. If e input difference to the S-box cannot cause f output difference, discard the (quartet, subkey guess).

   At this stage, each of the $2^{64.4}$ remaining quartets suggests one value on average for bytes $0, 4, 7, 8, 10, 12, 13$ of $K^9_a, K^9_b, K^9_c, K^9_d$, and for c, e, f, and half of them are expected to offer consistent suggestion for e and f.

- **Finding the Right Key (Step 9):**

9. If a subkey combination is suggested by six quartets or more, assume it is correct, and try to deduce the correct key using exhaustive search of the remaining bytes.

Analysis of the Attack In Steps 1,2,3 the adversary encrypts $2^{121.2}$ chosen plaintexts ($2^{113.2}$ plaintexts under each of the keys $K_a, K_b, \widetilde{K}_c$, and $\widetilde{K}_d$). Hence, the data complexity of this attack is about $2^{121.2}$ related-key chosen plaintexts. Steps 4(a) may look as if they take $2^{121.2} \cdot 2^{64} = 2^{185.2}$ partial encryptions each. However, as the values of the plaintexts in the bytes which are fixed and have no effect on the actual "pairing", then it is sufficient to partially encrypt a set of 2^{64} values under 2^{64} subkeys (and repeat this for each value of y). Hence, the total time complexity of an optimized implementation of each of these two steps is $2^{64} \cdot 2^{64} \cdot 2^{7} = 2^{135}$ partial encryptions (which are about 2^{132} encryptions in total). Steps 4(b) is composed of inserting the pairs found in Step 2(a) into tables. For each subkey guess there are $2^{113.2}$ pairs which are put into the table, and thus, this step takes $2^{64} \cdot 2^{113.2} \cdot 2^{7} = 2^{184.2}$ memory accesses.

Step 4(c) is similar to Step 4(a), and takes the same time. The same is true to Step 4(d) and Step 4(b), respectively. We note that Step 4(b) has to be performed only once and not 2^7 times like the other steps (as it is independent of $\widetilde{\Delta K_{ac}}$), and thus, Step 4 takes in total $2^{184.2}$ memory accesses.

Steps 5–8 each analyzes a small number of quartets, in a relatively efficient manner. Finally, Step 9, exhaustively searches over 2^{128} keys for each key candidate. Hence, we conclude that the running time of the attack is dominated by Steps 4(c) and 4(d), and is approximately $2^{184.2}$ memory accesses.

We can calculate the success rate of the attack by using the Poisson distribution. At the end of Step 8(d), we expect $2^{63.4}$ (quartet,subkey guess) combinations to remain. Given that there are 2^{79} possible values at this point (c, e, f suggest values for the related-keys K_b, K_d), we expect a wrong subkey to be suggested by $2^{-15.6}$ quartets on average, while the right subkey is expected to be suggested by $2^{226.4} \cdot 2^{-222.8} = 2^{3.6} = 12$ quartets. Hence, the probability that any given wrong tuple of subkeys (of the $2^{79} \cdot 2^{64} = 2^{143}$ subkeys) is suggested more than five times is about $2^{-103.1}$. Hence, we expect $2^{143} \cdot 2^{-103.1} = 2^{39.9}$ wrong subkeys to be analyzed in Step 9, which means that Step 9 has time complexity of $2^{167.9}$ trial encryptions.

The right subkey is expected to be suggested by more than five quartets with probability about 98.0%. When this is the case, the right key is found, and thus, the success rate of the attack is about 98.0%.

4.5 Reducing the Number of Related Keys from 256 to 64

The number of related keys used in our attack can be reduced from 256 to 64 using key structures. The following 64 related keys are used in our optimized attack:

- 16 keys K_a^i ($i = 0, 1, \cdots, 15$) such that $K_a^i \oplus K_a^j$ is zero in all bytes, besides bytes 3,8,11, and 12, and the difference in bytes 3,11 is fixed to some w and the difference in bytes 8,12 is fixed to some r (not necessarily $w \to r$).
- 16 keys K_b^i, each computed as $K_a^i \oplus \Delta K_{ab}$ for a specific value of x.
- 16 keys K_c^i ($i = 0, 1, \cdots, 15$) such that $K_c^i \oplus K_c^j$ is zero in all bytes, besides bytes 3,8,11, and 12, and the difference in bytes 3,11 is fixed to some w' and the difference in bytes 8,12 is fixed to some r' (not necessarily $w' \to r'$).
- 16 keys K_d^i, each computed as $K_c^i \oplus \Delta K_{ab}$ for the same value of x used to generate K_b^i.

Using these delicately chosen key relationships, we generate 256 key quartets $(K_a^i, K_b^i, K_c^j, K_d^j)$ of which one is expected to satisfy the subkey difference requirement of the attack (the attack can be applied where the value of x in the first and the second differentials are replaced by x_1 and x_2, respectively).

As we generate the data set for each possible key only once, the data complexity of the attack is reduced to $2^{119.2}$ related-key chosen plaintexts. On the other hand, the attack is now to be run 256 times rather than 127 times, resulting in a total running time of $2^{185.2}$ memory accesses.

5 Conclusions

In this paper we introduced the related-key boomerang and the related-key rectangle attacks. The attacks use weaknesses of the key schedule algorithms to achieve significant advantage over other attacks techniques. We presented a rigorous treatment of the new techniques, thus devising optimal distinguishers.

Both the related-key boomerang attack and the related-key rectangle attack enjoy the use of key differences twice. Hence, in exchange for an attack model with 4 related-keys, the adversary is able to attack a significantly larger amount of rounds than in the standard single-key model or a standard related-key differential attack.

Apart from the immediate attacks, another outcome of our results is a better understanding of the importance of well designed key schedule algorithms for the security of block ciphers. While it is commonly believed that a linear key schedule (or one close to it), is of no security concern to a well designed block cipher, the related-key boomerang and rectangle attacks along with the concept of structures of keys (for nonlinear key schedule algorithms) show that this belief is dangerous and at times may be faulty.

References

1. Thomas Baignères, Pascal Junod, Serge Vaudenay, *How Far Can We Go Beyond Linear Cryptanalysis*, Advances in Cryptology, proceedings of ASIACRYPT 2004, Lecture Notes in Computer Science 3329, pp. 432–450, Springer-Verlag, 2004.
2. Mihir Bellare, Tadayoshi Kohno, *A Theoretical Treatment of Related-Key Attacks: RKA-PRPs, RKA-PRFs, and Applications*, Advances in Cryptology, proceedings of EUROCRYPT 2003, Lecture Notes in Computer Science 2656, pp. 491–506, Springer-Verlag, 2003.
3. 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.
4. Eli Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Journal of Cryptology, Vol. 7, No. 4, pp. 229–246, Springer-Verlag, 1994.
5. Eli Biham, Adi Shamir, *Differential Cryptanalysis of DES-like Cryptosystems*, Advances in Cryptology, proceedings of CRYPTO 1990, Lecture Notes in Computer Science 537, pp. 2–21, Springer-Verlag, 1990.
6. Eli Biham, Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
7. Eli Biham, Orr Dunkelman, Nathan Keller, *The Rectangle Attack — Rectangling the Serpent*, Advances in Cryptology, proceedings of EUROCRYPT 2001, Lecture Notes in Computer Science 2045, pp. 340–357, Springer-Verlag, 2001.

8. Eli Biham, Orr Dunkelman and Nathan Keller, *New Results on Boomerang and Rectangle Attacks*, proceedings of Fast Software Encryption 2002, Lecture Notes in Computer Science 2365, pp. 1–16, Springer-Verlag 2002.
9. Eli Biham, Orr Dunkelman, Nathan Keller, *Related-Key Boomerang and Rectangle Attacks*, Advances in Cryptology, proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science 3494, pp. 507–525, Springer-Verlag, 2005.
10. Eli Biham, Orr Dunkelman, Nathan Keller, *A Related-Key Rectangle Attack on the Full KASUMI*, Advances in Cryptology, proceedings of ASIACRYPT 2005, Lecture Notes in Computer Science 3788, pp. 443–461, Springer-Verlag, 2005.
11. Eli Biham, Orr Dunkelman, Nathan Keller, *New Cryptanalytic Results on IDEA*, Advances in Cryptology, proceedings of ASIACRYPT 2006, Lecture Notes in Computer Science 4284, pp. 412–427, Springer-Verlag, 2006.
12. Eli Biham, Orr Dunkelman, Nathan Keller, *A Unified Framework for Related-Key Attacks*, proceedings of Fast Software Encryption 2008, Lecture Notes in Computer Science 5086, pp. 73–96, Springer-Verlag, 2008.
13. Alex Biryukov, *The Boomerang Attack on 5 and 6-Round AES*, Proceedings of Advance Encryption Standard Fourth Workshop, Lecture Notes in Computer Science 3373, pp. 11–16, Springer-Verlag, 2005.
14. Alex Biryukov, Dmitry Khovratovich, *Related-key Cryptanalysis of the Full AES-192 and AES-256*, IACR ePrint report 2009/317, 2009. Available online at `http://eprint.iacr.org/2009/317`.
15. Mark Blunden, Adrian Escott, *Related Key Attacks on Reduced Round KASUMI*, proceedings of Fast Software Encryption 2001, Lecture Notes in Computer Science 2355, pp. 277–285, Springer-Verlag, 2002.
16. Joan Daemen, Vincent Rijmen, *The design of Rijndael: AES — the Advanced Encryption Standard*, Springer-Verlag, 2002.
17. Joan Daemen, Vincent Rijmen, *Understanding Two-Round Differentials in AES*, proceedings of Security and Cryptography for Networks 2006, Lecture Notes in Computer Science 4116, pp. 78–94, Springer-Verlag, 2006.
18. Hüseyin Demirci, Ali Aydin Selçuk, *A Meet-in-the-Middle Attack on 8-Round AES*, proceedings of Fast Software Encryption 2008, Lecture Notes in Computer Science 5086, pp. 116–126, Springer-Verlag, 2008.
19. Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner, Doug Whiting, *Improved Cryptanalysis of Rijndael*, proceedings of Fast Software Encryption 2000, Lecture Notes in Computer Science 1978, pp. 213–230, Springer-Verlag, 2001.
20. Michael Gorski, Stefan Lucks, *New Related-Key Boomerang Attacks on AES*, proceedings of INDOCRYPT 2008, Lecture Notes in Computer Science 5365, pp. 266–278, Springer-Verlag, 2008.
21. 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 2005, Lecture Notes in Computer Science 3557, pp. 368–383, Springer-Verlag, 2005.
22. 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.
23. John Kelsey, Bruce Schneier, David Wagner, *Key Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES*, Advances in Cryptology, proceedings of CRYPTO 1996, Lecture Notes in Computer Science 1109, pp. 237–251, Springer-Verlag, 1996.

24. 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 Communications Security 1997, Lecture Notes in Computer Science 1334, pp. 233–246, Springer-Verlag, 1997.
25. John Kelsey, Tadayoshi Kohno, Bruce Schneier, *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent*, proceedings of Fast Software Encryption 2001, Lecture Notes in Computer Science 1978, pp. 75–93, Springer-Verlag, 2002.
26. Jongsung Kim, Guil Kim, Seokhie Hong, Dowon Hong, *The Related-Key Rectangle Attack — Application to SHACAL-1*, proceedings of Australasian Conference on Information Security and Privacy 2004, Lecture Notes in Computer Science 3108, pp. 123–136, Springer-Verlag, 2004.
27. Jongsung Kim, Seokhie Hong, Bart Preneel, *Related-Key Rectangle Attacks on Reduced AES-192 and AES-256*, proceedings of FSE 2007, Lecture Notes in Computer Science 4593, pp. 225–241, Springer-Verlag, 2007.
28. Lars R. Knudsen, *Cryptanalysis of LOKI91*, proceedings of Auscrypt 1992, Lecture Notes in Computer Science 718, pp. 196–208, Springer-Verlag, 1993.
29. Ulrich Kühn, *Cryptanalysis of Reduced-Round MISTY*, Advances in Cryptology, proceedings of EUROCRYPT 2001, Lecture Notes in Computer Science 2045, pp. 325–339, Springer-Verlag, 2001.
30. X. Lai, J.L. Massey, S. Murphy, *Markov Ciphers and Differential Cryptanalysis*, Advances in Cryptology, proceedings of EUROCRYPT '91, Lecture Notes in Computer Science 547, pp. 17–38, Springer-Verlag, 1992.
31. Mitsuru Matsui, *Linear Cryptanalysis Method for DES Cipher*, Advances in Cryptology, proceedings of EUROCRYPT '93, Lecture Notes in Computer Science 765, pp. 386–397, Springer-Verlag, 1994.
32. National Institute of Standards and Technology, *Advanced Encryption Standard*, Federal Information Processing Standards Publications No. 197, 2001.
33. Kaisa Nyberg, *Perfect Nonlinear S-boxes*, Advances in Cryptology, proceedings of EUROCRYPT 1991, Lecture Notes in Computer Science 547, pp. 378–386, Springer-Verlag, 1991.
34. Kaisa Nyberg, Lars R. Knudsen, *Provable Security Against Differential Cryptanalysis*, Advances in Cryptology, proceedings of CRYPTO 1992, Lecture Notes in Computer Science 740, pp. 566–578, Springer-Verlag, 1993.
35. Ali Aydin Selçuk, *On Probability of Success in Linear and Differential Cryptanalysis*, Journal of Cryptology Vol. 21 No. 1, pp. 131–147, Springer-Verlag, 2008.
36. 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.
37. Gene Tsudik, Els Van Herreweghen, *On simple and secure key distribution*, Conference on Computer and Communications Security, Proceedings of the 1st ACM conference on Computer and communications security, pp. 49–57, ACM, 1993.
38. David Wagner, *The Boomerang Attack*, proceedings of Fast Software Encryption 1999, Lecture Notes in Computer Science 1636, pp. 156–170, Springer-Verlag, 1999.
39. Gaoli Wang, Nathan Keller, Orr Dunkelman, *The Delicate Issues of Addition with Respect to XOR Differences*, proceedings of Selected Areas in Cryptography 2007, Lecture Notes in Computer Science 4876, pp. 212–231, Springer-Verlag, 2007.
40. David J. Wheeler, Roger M. Needham, *TEA, a Tiny Encryption Algorithm*, proceedings of Fast Software Encryption 1994, Lecture Notes in Computer Science 1008, pp. 363–366, Springer-Verlag, 1995.

41. ZDNet, New Xbox security cracked by Linux fans, 2002. Available on-line at `http://news.zdnet.co.uk/software/developer/0,39020387,2123851,00.htm`.

42. Wentao Zhang, Wenling Wu, Lei Zhang, Dengguo Feng, *Improved Related-Key Impossible Differential Attacks on Reduced-Round AES-192*, proceedings of Selected Areas in Cryptography 2006, Lecture Notes in Computer Science 4356, pp. 15–27, Springer-Verlag, 2007.

43. Wentao Zhang, Lei Zhang, Wenling Wu, Dengguo Feng, *Related-Key Differential-Linear Attacks on Reduced AES-192*, proceedings of INDOCRYPT 2007, Lecture Notes in Computer Science 4859, pp. 73-85, Springer-Verlag, 2007.