Related-Key Boomerang and Rectangle Attacks

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

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

Abstract. The boomerang attack and the rectangle attack are two attacks that utilize differential cryptanalysis in a larger construction. Both attacks treat the cipher as a cascade of two sub-ciphers, where there exists a good differential for each sub-cipher, but not for the entire cipher. In this paper we combine the boomerang (and the rectangle) attack with related-key differentials.
The new combination is applicable to many ciphers, and we demonstrate its strength by introducing attacks on reduced-round versions of AES and IDEA. The attack on 192-bit key 9-round AES uses 256 different related keys. The 6.5-round attack on IDEA uses four related keys (and has time complexity of $2^{88.1}$ encryptions). We also apply these techniques to COCONUT98 to obtain a distinguisher that requires only four related-key adaptive chosen plaintexts and ciphertexts. For these ciphers, our results attack larger number of rounds or have smaller complexities then all previously known attacks.

1 Introduction

The *boomerang attack* [23] is an adaptive chosen plaintext and ciphertext attack utilizing differential cryptanalysis [6]. The cipher is treated as a cascade of two sub-ciphers, where a short differential is used in each of these sub-ciphers. These two differentials are combined in an elegant way to suggest an adaptive chosen plaintext and ciphertext property of the cipher that has high probability.

The boomerang attack was further developed in [18] into a chosen plaintext attack called the *amplified boomerang attack*. The transformation uses birthday-paradox techniques to eliminate the adaptive nature of the attack, by encrypting large sets of plaintexts with the required input difference. After the encryption of the plaintext pairs, the attacker searches for quartets of plaintexts that satisfy the same conditions as if these quartets were constructed in the boomerang process. The transformation to a chosen plaintext attack (instead of an adaptive chosen plaintexts and ciphertexts attack) has price both in a much larger data

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

complexity and a much more complicated algorithm for the identification of the right quartets. After its introduction, the amplified boomerang attack was further developed into the rectangle attack [4]. The rectangle attack uses a more careful analysis that shows that the probability of a right quartet is significantly higher than suggested by the amplified boomerang attack. An optimized algorithm for finding and identifying the right rectangle quartets was given in [5].

Related-key attacks [1] consider the information that can be extracted from two encryptions using related keys. The concept was used in [19] to present the idea of related-key differentials. These differentials study the development of differences in two encryptions under two related keys.

In this paper we show how to combine these attacks with related-key differentials. In [20], a boomerang attack that uses one regular differential along with one related-key differential is introduced. Both this paper and [16] independently developed the idea of using two related-key differentials, one for each sub-cipher, simultaneously. The major difference between this work and [16] is the idea of using more than one key difference in the differentials to obtain much better attacks.

The basic related-key boomerang attack (which is similar to the one presented in [16]) is aimed against ciphers whose subkeys are linear functions of the key. In this case, a fixed key difference yields a known subkey differences.

The more complicated version of the attack deals with ciphers whose subkeys are not linear functions of the keys. In this case, the attacker has to take into consideration the fact that the initial key difference does not guarantee the subkey differences used in the differential. In order to overcome this problem, we use differential properties of the key schedule algorithm and use several pairs of keys. This leads to the introduction of structures of keys under which structures of plaintexts are being encrypted or decrypted.

We take advantage of the fact that in boomerang and rectangle attacks the used differentials are shorter, and thus the diffusion of differences in the subkeys can be used better than in ordinary related-key differential case.

Finally, we apply our attack against several block ciphers: AES [12], IDEA [21], and COCONUT98 [22]. The attack on 9-round AES-192 requires 2^{87} related-key chosen plaintexts (2^{79} plaintexts encrypted under 256 different keys), and has running time of 2^{125} encryption. The attack on 6.5-round IDEA requires $2^{59.8}$ related-key chosen plaintexts ($2^{57.8}$ plaintexts encrypted under four keys), and has time complexity of $2^{88.1}$ encryptions. We also apply these techniques to COCONUT98 to obtain a distinguisher that requires only four related-key adaptive chosen plaintexts and ciphertexts encrypted under two different keys. We summarize our results along with previously known results on the respective ciphers in Table 1.

This paper is organized as follows: In Section 2 we give a brief description of the boomerang and the rectangle attacks. In Section 3 we describe the new related-key boomerang and rectangle attacks. In Section 4 we present a related-key rectangle attack on 9-round AES-192 and 10-round AES-256. In Section 5

Cipher	Number of Rounds	Complexity Data	Complexity Time	Number of Source Keys	
AES-192	8	$2^{128}-2^{119}$ CP	2^{188}	1	[14]
(12 rounds)	8	2^{89} RK-CP	2^{183}	2	[17]
	8	$2^{86.5}$ RK-CP	$2^{86.5}$	4	[16]
	9	2^{86} RK-CP	2^{125}	256	Section 4
AES-256	8	$2^{128}-2^{119}$ CP	2^{204}	1	[14]
(14 rounds)	9	2^{85} RK-CP	$5 \cdot 2^{224}$	256	[14]
	10	$2^{114.9}$ RK-CP	$2^{171.8}$	256	Section 4
COCONUT98	full	2^{16} ACPC	2^{38}	1	[23]
	full†	4 RK-ACPC	1	2	Section 5
IDEA	5	2^{24} CP	2^{126}	1	[13]
(8.5 rounds)	5.5†	$2^{51.6}$ RK-ACPC	1	4	Section 6
	6	$2^{51.6}$ RK-ACPC	2^{48}	4	Section 6
	6.5	$2^{59.8}$ RK-CP	$2^{88.1}$	4	Section 6

† – Distinguishing attack, RK – Related-key, CP – Chosen plaintext,
ACPC – Adaptive chosen plaintext and ciphertext
Time complexity is measured in encryption units

Table 1. Summary of the Previous Attacks and of Our New Attacks

we present a related-key boomerang distinguisher for COCONUT98. Section 6 describes our results on IDEA. Finally, Section 7 summarizes this paper.

2 Boomerang and Rectangle Attacks

The main idea behind the boomerang attack [23] is to use two short differentials with high probabilities instead of one long differential with a low probability. The motivation for such an attack is quite apparent, as in many block ciphers it is easier to find short differential with high probability than to find a long differential with high enough probability (or even impossible).

We assume that a block cipher $E:\{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ can be described as a cascade, i.e., $E = E_1 \circ E_0$, such that for E_0 there exists a differential $\alpha \to \beta$ with probability p, and for E_1 there exists a differential $\gamma \to \delta$ with probability q. The distinguisher is the following boomerang process:

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

The boomerang attack uses the first differential ($\alpha \to \beta$) for E_0 with respect to the pairs (P_1, P_2) and (P_3, P_4), and uses the second differential ($\gamma \to \delta$) for E_1 with respect to the pairs (C_1, C_3) and (C_2, C_4). The first differential is used in the backward direction for the pairs (P_3, P_4), and the second differential is used in the backward direction for both respective pairs.

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

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

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

For the complete analysis of the boomerang attack see [23].

As the boomerang attack requires adaptive chosen plaintexts and ciphertexts, many of the techniques that were developed for using distinguishers in key recovery attacks cannot be applied. This led to the introduction of a chosen plaintext variant of the boomerang attack called the *amplified boomerang attack* [18]. The key idea behind the transformation is to encrypt many plaintext pairs with input difference α, and to look for quartets that conform to the requirements of the boomerang process.

This kind of transformation is common, and can be achieved by birthday-paradox arguments. A more careful analysis shows that two pairs, $(P_1, P_2 = P_1 \oplus \alpha)$ and $(P_3, P_4 = P_3 \oplus \alpha)$, form a right quartet if three 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 and the previous one hold).
3. $C_1 \oplus C_3 = \delta = C_2 \oplus C_4$.

The usual assumptions are that each of these conditions is independent of the rest, and that the probability that a quartet would become a right quartet is $p^2 \cdot 2^{-n} \cdot q^2$. We note that if the conditions are dependent on each other, refined algorithms may use these relations for achieving higher probabilities. The low probability follows from the fact that the event $E_0(P_1) \oplus E_0(P_3) = \gamma$ occurs with probability of 2^{-n}. The analysis in [18] shows that out of N plaintext pairs, the number of right quartets is expected to be $N^2 2^{-(n+1)} p^2 q^2$.

Besides the lower probabilities, the transformation into a chosen plaintext attack introduces the problem of identifying the right quartets. In the boomerang attack the pair (P_3, P_4) that we test is known. In the amplified boomerang attack, this is not the case. Instead, the attacker has to search for the right quartets among all possible quartets.

The rectangle attack [4] shows that it is possible to use all the possible β's and γ's simultaneously, and presents additional improvements over the amplified boomerang attack. These improvements increase the probability of a quartet to be a right quartet, and N plaintext pairs with input difference α are expected to produce $N^2 2^{-n} \hat{p}^2 \hat{q}^2$ right quartets[1], where $\hat{p}$ and $\hat{q}$ are as defined above. In [5] an optimized method of finding the right rectangle quartets is presented.

3 Related-Key Boomerang and Rectangle Attacks

A regular differential deals with some plaintext difference ΔP and a ciphertext difference ΔC such that

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

is high enough (or zero [3]). The common assumption is that this probability is quite uniform over all keys and plaintexts. If this is not the case, a weak key class can be found, i.e., a set of keys for which the above probability is far from average (either very high or very low).

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

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

is high enough (or zero). Again, there is an assumption that this probability is independent of P and K. Sometimes the relation between the keys is more complex than XOR with some constant ΔK (see [1, 9]), but for sake of simplicity we shall deal only with this kind of relation in this paper, even though our technique is not restricted for this case.

3.1 Related-Key Boomerang Attacks

Let us assume that we have a related-key differential $\alpha \to \beta$ of E_0 under a key difference ΔK_0 with probability p. Assume also that we have another related-key differential $\gamma \to \delta$ for E_1 under key difference ΔK_1 with probability q.

The related-key boomerang process involves four different unknown (but related) keys — K_a, $K_b = K_a \oplus \Delta K_0$, $K_c = K_a \oplus \Delta K_1$, and $K_d = K_a \oplus \Delta K_0 \oplus \Delta K_1$. The attack is performed by the following algorithm:

- Choose a plaintext P_a at random and compute $P_b = P_a \oplus \alpha$.
- Ask for the encryption of P_x under K_x, i.e., $C_a = E_{K_a}(P_a)$ and $C_b = E_{K_b}(P_b)$.
- Compute $C_c = C_a \oplus \delta$ and $C_d = C_b \oplus \delta$.
- Ask for the decryption of C_x under K_x, i.e., $P_c = E^{-1}_{K_c}(C_c)$ and $P_d = E^{-1}_{K_d}(C_d)$.

[1] This number is a lower bound for the expected number. For the complete analysis see [4].

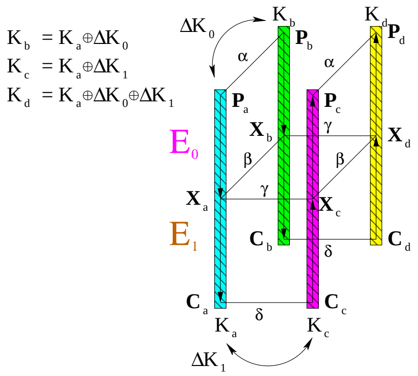

Fig. 1. A Related-Key Boomerang Quartet

– Test whether $P_c \oplus P_d = \alpha$.

It is easy to see that for a random permutation, the probability that the last condition is satisfied is 2^{-n}. For E the probability that this condition is satisfied is p^2q^2 just like for a regular boomerang attack. Figure 1 outlines a quartet satisfying all the required conditions.

The attack can use multiple differentials for E_0 and E_1 (just like in a regular boomerang attack), under the strict condition that all related-key differentials used in E_0 have the same key difference ΔK_0 and the same input difference α, and that all related-key differentials used in E_1 have the same key difference ΔK_1 and the same output difference δ. Thus, the probability of a quartet to be a right quartet is $\hat{p}^2\hat{q}^2$.

When the key schedule algorithm is linear then given a key difference all subkey differences are known, and are easily predicted. In this case the attack algorithm from [5] can be adapted. Otherwise, if the key schedule algorithm is non-linear, the exact key difference needed to satisfy the subkey differences of the related-key differential might be unknown. In the latter case, the attacker examines the differential properties of the key schedule algorithm and computes the probability that a given key differences evolves into the required subkey differences. Then, the attacker repeats the attack with various key differences, such that in one (or more) of the cases, the key difference causes the subkey differences needed for the related-key differential. Note that this attack is actually a multiple application of the basic related-key boomerang/rectangle attacks. An example of such an attack is the attack on AES-192 presented in Section 4.

3.2 Related-Key Rectangle Attack

The transformation of the related-key boomerang attack into a related-key rectangle attack is similar to the transformation of the boomerang attack into the rectangle attack. Assume that E can be decomposed as before, where $\alpha, \delta, \hat{p}$, and $\hat{q}$ have the same meaning. Then, related-key rectangle distinguisher is as follows:

– Choose N plaintext pairs $(P_a, P_b = P_a \oplus \alpha)$ at random and ask for the encryption of P_a under K_a and of P_b under K_b.
– Choose N plaintext pairs $(P_c, P_d = P_c \oplus \alpha)$ at random and ask for the encryption of P_c under K_c and of P_d under K_d.
– Search for quartets of plaintexts (P_a, P_b, P_c, P_d) and the corresponding ciphertexts (C_a, C_b, C_c, C_d), satisfying $C_a \oplus C_c = C_b \oplus C_d = \delta$.

The analysis of the related-key rectangle attack is similar to the analysis of the rectangle attack (with the same modifications that were presented at the related-key boomerang attack). Starting with N plaintext pairs with input difference α to be encrypted under K_a and K_b, we expect $N^2 2^{-n}(\hat{p}\hat{q})^2$ right quartets. We note that under the requirement that we encrypt distinct values N is no longer bounded by 2^{n-1} (as in the rectangle attack), but rather can be up to 2^n pairs in most of the cases (when $\Delta K_0 = 0, \Delta K_1 = 0$, or $\Delta K_0 = \Delta K_1$ the value of 2^{n-1} is still the bound).

The step of finding the right quartets is technical in nature (the simplest algorithm is trying all possible quartets, which is very inefficient). When the key difference predicts the required subkey differences with probability 1, then the attack algorithm of [5] can be easily adapted. Otherwise, we have to perform a method similar to the one of the boomerang attack — repeat the attack for several quartets of keys.

4 Related Key Rectangle Attacks on AES-192

The advanced encryption standard [12] is an SP-network that supports key sizes of 128, 192, and 256 bits. The 128-bit plaintexts are treated as byte matrices of size 4x4, where each byte represents a value in $GF(2^8)$. An AES round applies four operations to the state matrix: SubBytes (SB) – applying the same S-box 16 times in parallel on each byte of the state, ShiftRows (SR) – cyclic shift of each row (the i'th row is shifted by i bytes to the right), MixColumns (MC) – multiplication of each column by a constant 4x4 matrix over the field $GF(2^8)$, and AddRoundKey (ARK) – XORing the state and a 128-bit subkey.

The MixColumns operation is omitted in the last round, and an additional AddRoundKey operation is performed before the first round (a whitening key). We denote the subkey of round i by superscript K^{i+1}, i.e., the first (whitening) key is K^0, the subkey of the first round is K^1, etc. We also denote the byte in the i'th row and the j'th column of the state matrix by byte $j * 4 + i$, where $i, j \in \{0, 1, 2, 3\}$.

The number of rounds depends on the key length: 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. The rounds are numbered $0, \ldots, Nr-1$, where Nr is the number of rounds ($Nr \in \{10, 12, 14\}$). For sake of simplicity we shall denote AES with n-bit keys by AES-n, i.e., AES with 128-bit keys (and thus with 10 rounds) is denoted by AES-128.

The best published differential-based attack on AES is a boomerang attack on a 6-round reduced version of AES-128 [7]. The attack requires 2^{71} adaptive chosen plaintexts and ciphertexts, and its time complexity is equivalent to 2^{71}

encryptions. The best known attack on AES-192 is a SQUARE attack on 8 rounds [14]. The attack requires almost the entire code book ($2^{128}-2^{119}$ chosen plaintexts) and has a time complexity equivalent to 2^{188} encryptions. The best related-key attack against AES-192 is a related-key rectangle attack on 8 rounds [16]. It requires $2^{86.5}$ chosen plaintexts (encrypted under four keys) and has a time complexity equivalent to $2^{86.5}$ encryptions.

In this section we present a related-key rectangle attack on 9-round AES-192. The attack has data complexity of 2^{87} related-key chosen plaintexts (2^{79} chosen plaintexts encrypted under 256 keys), and time complexity of 2^{125} encryptions. By using similar techniques, one can attack 10-round AES-256 using $2^{114.9}$ related-key chosen plaintexts ($2^{106.9}$ chosen plaintexts encrypted under 256 keys) with time complexity of $2^{171.8}$ encryptions.

We concentrate on AES-192, as it demonstrates our attack when the key schedule is not linear, but is still very close to linear. The attack uses structures of plaintexts, encrypted under structures of keys, where the structures of keys are sets of keys selected to assure that there exists a quartet of keys whose subkeys satisfy the required differences.

4.1 Preliminaries for the Attack on 9-Round AES-192

The application of the related-key rectangle attack to AES-192 is as follows: We start by finding two good related-key differentials. In the second differential the key difference cannot guarantee the required subkey differences needed for the differential. Thus, we repeat the attack with 127 key differences, as we are assured that for at least one of these values the subkey differences are satisfied.

We decompose 9-round AES-192 (starting in the third round — rounds 2–10) as follows: rounds 2 and 3 are the rounds before the distinguisher, rounds 4–6 are E_0 (without the key addition of round 6), rounds 7–9 (with the key addition of round 6) are E_1, and round 10 is the round after the distinguisher.

Let θ_0 be a fixed 8-bit known non-zero difference, and let $\theta = (\theta_0, 0, 0, 0)$ be a 32-bit difference (the three 0's are byte differences). Let $\chi = (\chi_0, \chi_1, \chi_2, \chi_3) = MC(\theta)$, where χ_i are non-zero byte differences, and MC is the MixColumns operation. The value of χ is known as the MC operation is linear, thus, differentials propagate linearly through it as well.

Let θ by an input difference to the $MC(BS())$ operations. We denote all 127 possible output differences by MB, i.e., $MB = \{MC(BS(x)) \oplus MC(BS(x \oplus \theta))$.

4.2 The First Differential (E_0)

The first differential (for rounds 4–6) is as follows: the subkey difference of K^4 is equal to the input difference and being of the form $\alpha = \Delta K^4 = (0, 0, \chi, \chi)$ (here the 0's are 32-bit differences). After the key addition, the difference of the data is zero, which remains through round 4 with probability 1. We set $\Delta K^5 = 0$, and we get that the zero difference remains after round 5 with probability 1 as well. Then, the subkey difference ΔK^6 is necessarily $(\chi, 0, 0, 0)$, which leads to

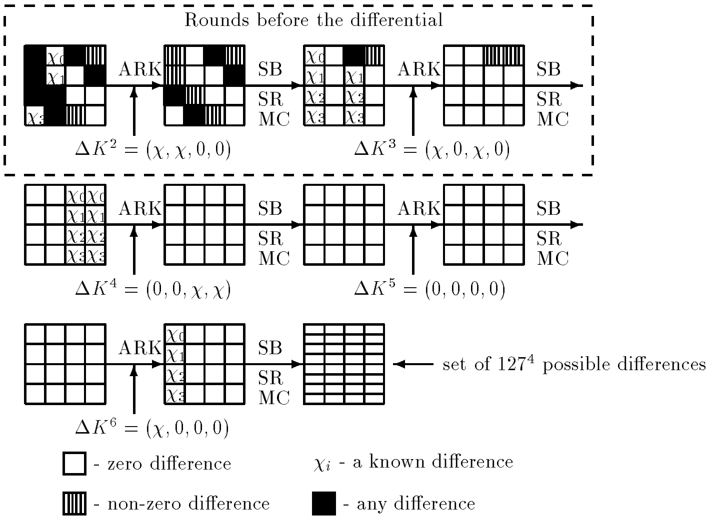

Fig. 2. The First Differential Used in the Attack (and the Two Preceding Rounds)

the activation of four S-boxes at round 6. As χ_i are all known and fixed, there are 127 possible output differences in each of the four active S-boxes, of which one occurs with probability 2^{-6}, and the rest with probability 2^{-7}. Therefore, the probabilities of the 127^4 β output differences are distributed as follows: one has probability 2^{-24}, $4 \cdot 126$ have probability 2^{-25}, and so forth, up to 126^4 with probability 2^{-28}. As we use all these differentials simultaneously, we get that $\hat{p} = \sqrt{\sum_\beta \Pr{}^2[\alpha \to \beta]} = 2^{-13.96}$.

We note that the above describes a differential characteristic (we predict the development of the difference in all rounds). In the case of the AES, the probability that $\alpha \to \beta$ is very close to the probability of the characteristic.

We look for the (related-key) input difference to round 2 that leads to an α difference at the input of round 4. The input difference for round 2 consists of four S-boxes with a zero input difference (bytes $9, 10, 14, 15$), three S-boxes whose non-zero difference is known (bytes $3, 4, 5$), two additional S-boxes with unknown non-zero difference (bytes $11, 12$), and the remaining seven S-boxes can have any difference. We denote the difference in the bytes whose difference is known by ΔM_0, where we put zeroes in the bytes whose difference in unknown (ΔM_0 has four non-zero bytes). We outline these differences of the differential and the preceding rounds in Figure 2. The first differential's key difference is $\Delta K_0 = (\chi, 0, 0, 0, \chi, 0)$, and we outline the subkey differences in Table 2.

4.3 The Second Differential (E_1)

The second differential predicts the differences in E_1 (rounds 7–9). The input difference γ equals the subkey difference, and both are $\gamma = \Delta K^7 = (0, 0, \theta, \theta)$. Thus, after the key addition the difference is zero, which passes with probability 1

Subkey	Difference	Subkey	Difference	Subkey	Difference	Subkey	Difference
K^0	$(\chi, 0, 0, 0)$	K^2	$(\chi, \chi, \mathbf{0}, \mathbf{0})$	K^4	$(\mathbf{0}, \mathbf{0}, \chi, \chi)$	K^6	$(\chi, \mathbf{0}, \mathbf{0}, \mathbf{0})$
K^1	$(\chi, 0, \chi, \chi)$	K^3	$(\chi, \mathbf{0}, \chi, \mathbf{0})$	K^5	$(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$		

Table 2. Subkey Differences for the Key Difference $\Delta K_0 = (\chi, 0, 0, 0, \chi, 0)$ (The subkey differences of the differential are in bold)

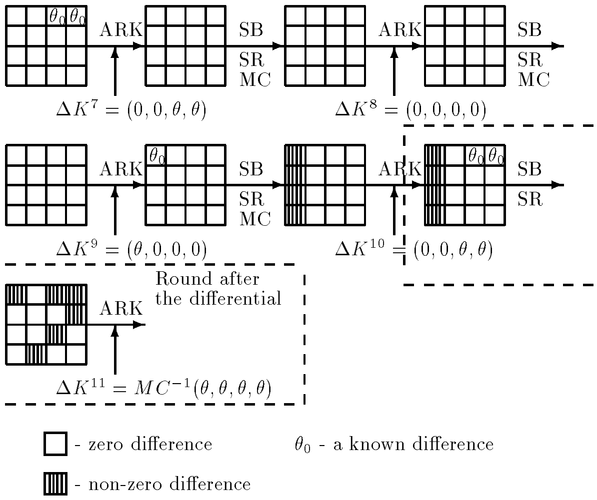

Fig. 3. The Second Differential Used in the Attack (and the Following Round)

through round 7. We take $\Delta K^8 = 0$, and thus, the zero difference remains with probability 1 also after round 8. Once we select ΔK^7 and ΔK^8, then necessarily $\Delta K^9 = (\theta, 0, 0, 0)$. Thus, the key mixing before round 9 introduces a difference θ_0 in byte 0. This byte difference creates a difference in a full column before the addition of K^{10}. The subkey difference is $\Delta K^{10} = (0, 0, \theta, \theta)$, hence, the output difference δ of the differential has four active bytes (in bytes $0, \ldots, 3$) and twelve bytes with a zero difference. The difference in bytes $0, \ldots, 3$ is unknown but is restricted to a set of 127 possible differences. We outline the differences of this differential and the following round in Figure 3. We note that the differential has probability 1, leading to $\hat{q} = 1$.

The subkey differences of the second differential are given in Table 3. The key difference that achieves the required subkey differences for the second differential is of the form $\Delta K_1 = (\mu, \theta \oplus \mu, \theta, 0, \theta, 0)$, where $\mu = (0, \mu_1, 0, 0)$ and $\theta_0 \to \mu_1$ by the S-box. The exact value of μ_1 is unknown, but μ_1 must be one of 127 possible values.

4.4 The Structure of Keys

Let K_a be the unknown key which we would like to recover. The related-key that is required for the first differential is $K_b = K_a \oplus \Delta K_0$.

Subkey	Difference	Subkey	Difference	Subkey	Difference	Subkey	Difference
K^0	$(\mu, \theta \oplus \mu, \theta, 0)$	K^3	$(\theta, 0, 0, 0)$	K^6	$(\theta, 0, \theta, 0)$	K^9	$(\theta, \mathbf{0}, \mathbf{0}, \mathbf{0})$
K^1	$(\theta, 0, \mu, \theta)$	K^4	$(\theta, 0, \theta, \theta)$	K^7	$(\mathbf{0}, \mathbf{0}, \theta, \theta)$	K^{10}	$(\mathbf{0}, \mathbf{0}, \theta, \theta)$
K^2	$(0, 0, \theta, \theta)$	K^5	$(\theta, \theta, 0, 0)$	K^8	$(\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0})$	K^{11}	$(\theta, \theta, \theta, \theta)$

Table 3. Subkey Differences for the Key Difference $\Delta K_1 = (\mu, \theta \oplus \mu, \theta, 0, \theta, 0)$ (The subkey differences in the rounds of the attack are in bold)

Key	Values	#	Key	Values	#
K_a	K_a	1	K_c	$\{K_a \oplus (\mu, \theta \oplus \mu, \theta, 0, \theta, 0)\}$	127
K_b	$K_a \oplus (\chi, 0, 0, 0, \chi, 0)$	1	K_d	$\{K_a \oplus (\mu \oplus \chi, \theta \oplus \mu, \theta, 0, \theta \oplus \chi, 0)\}$	127

Table 4. The Keys Required for the Related-Key Rectangle Attack

Examine K_a and the subkey differences needed for the second differential. There are 127 possible related keys with which K_a may have the required subkey differences. Denote this set of keys by KS_c, which is actually all the keys that satisfy $K_a \oplus \Delta K_1$ for some value of μ_1, where $\mu = (0, \mu_1, 0, 0)$ and $\Delta K_1 = (\mu, \theta \oplus \mu, \theta, 0, \theta, 0)$. One key of this set satisfies the required subkey differences with respect to K_a, and we denote it by K_c.

We denote the key with difference ΔK_0 from K_c by K_d, i.e., $K_d = K_c \oplus \Delta K_0$. If we want to use the four keys, it must hold that K_b and K_d have the subkey difference required by the second differential, i.e., $K_b = K_d \oplus \Delta K_1$. In this case this is true, as there is no difference between K_a and K_b in the word that we need the equality (for the S-box application). Thus, $K_a \oplus K_c = K_b \oplus K_d$ is true. We outline the sets of keys in Table 4.

Note that we can choose a smaller number of keys, such that using the birthday-paradox we get with high probability the required quartets of keys, but then the attack may fail in a small fraction of the cases.

4.5 The Attack

The main idea of the attack is to try all possible quartets of keys (K_a, K_b, K_c, K_d) that can satisfy the required subkey differences, by performing the rectangle attack from [5] on each of these possibilities. The attack presented here is an optimized version of this idea, in which we take advantage of the fact that once the keys with the right relations are encountered, then there is no need to continue the attack for other quartets of keys. The attack algorithm is as follows:

1. Data Generation:
   (a) Generate 64 structures $S_1^a, \ldots, S_{64}^a$ of 2^{72} plaintexts each, where in each structure the 56 bits of bytes $3, 4, 5, 9, 10, 14, 15$ are fixed. Ask for the encryption of the structures under K_a.

(b) XOR any plaintext encrypted under K_a with ΔM_0, and ask for the encryption of the resulting plaintext under K_b (to obtain $S_1^b, \ldots, S_{64}^b$).

(c) For any possible value μ_1 such that $\theta_0 \to \mu_1$, let $\Delta K_1 = (\mu, \theta \oplus \mu, \theta, 0, \theta, 0)$, perform:

   i. Generate 64 structures $S_1^{c'}, \ldots, S_{64}^{c'}$ of 2^{72} plaintexts each, where in each structure the 56 bits of bytes $3, 4, 5, 9, 10, 14, 15$ are fixed. Ask for the encryption of the structure under $K_{c'} = K_a \oplus \Delta K_1$.

   ii. XOR any plaintext encrypted under $K_{c'}$ with ΔM_0, and ask for the encryption of the resulting plaintexts under $K_{d'} = K_{c'} \oplus \Delta K_0$ (to obtain $S_1^{d'}, \ldots, S_{64}^{d'}$).

2. Data Analysis: For any $K_{c'}$, the respective $K_{d'}$, and the corresponding structures:

   (a) For any pair of structures $S_i^b, S_j^{d'}$ perform:

      i. Insert the 2^{72} ciphertexts of $S_i^b, S_j^{d'}$ into a hash table indexed by the 80 bits of bytes $4, 5, 6, 7, 9, 10, 11, 13, 14, 15$. About $2^{72} \cdot 2^{72}/2^{80} = 2^{64}$ collisions are expected.

      ii. For each 80-bit collision (where one ciphertext is from S_i^b and one is from $S_j^{d'}$) check that the ciphertext difference in bytes 2 and 3 may be caused by an input difference θ_0 to the S-box. If this is not the case, discard the pair.

      iii. For each of the expected 2^{62} remaining pairs, try all 2^{32} possible values of bytes $0, 7, 10, 13$ of K^{11}, and partially decrypt the pair. If the difference of the partially decrypted pair is in MB, add the pair to a list of pairs related to the subkey. We note that each pair is expected to be in 127 lists, and that each list contains about $127 \cdot 2^{30} \approx 2^{37}$ pairs.

   (b) Insert **all** the ciphertexts of $S_1^a, \ldots, S_{64}^a, S_1^{c'}, \ldots, S_{64}^{c'}$ into a hash table indexed by the 80 bits of bytes $4, 5, 6, 7, 9, 10, 11, 13, 14, 15$. About $2^{78} \cdot 2^{78}/2^{80} = 2^{76}$ collisions are expected.

   (c) For each pair of ciphertexts that collide on the 80 bits (one encrypted under K_a and one under $K_{c'}$) do:

      i. Check that the ciphertext difference in bytes 2 and 3 may be caused by an input difference θ_0 to the S-box. If this is not the case, discard the pair. (about 1/4 of the pairs remain after this step).

      ii. Let $C_a \in S_l^a$ and $C_{c'} \in S_m^{c'}$ be the colliding ciphertexts, and P_a and $P_{c'}$ the respective plaintexts. Try all 2^{32} values for the bytes $0, 7, 10, 13$ of K^{11}, and partially decrypt the pair. If the difference of the partially decrypted values is in MB, access the list of pairs that corresponds to this subkey guess and the structures $S_l^b, S_m^{d'}$ from Step 2(a)(iii). Consider the pair P_a, $P_{c'}$, and each of the 2^{37} pairs of plaintexts under that subkey (as part of K_b and $K_{d'}$) as a candidate quartet (this leads to a total of 2^{44} candidate quartets with P_a and $P_{c'}$).

      iii. For any candidate quartet:

         – Check what is the key value for which the pairs $(C_a, C_{c'})$ and $(C_b, C_{d'})$ are partially decrypted to have a θ_0 difference in byte 8.

This operation can be performed efficiently using precomputed tables.
– Do the same for byte 12 of the ciphertext pairs.
– Check what is the key value for which the pairs (P_a, P_b) and $(P_{c'}, P_{d'})$ are partially encrypted to have a θ_0 difference in byte 2.
– Check what is the key value for which the pairs (P_a, P_b) and $(P_{c'}, P_{d'})$ are partially encrypted to have a $(x, 0, 0, 0)$ difference in bytes 1,5,11,12, where $x \to \theta_0$ by the S-box.
– Do the same for the bytes $2, 7, 8, 13$ of the plaintext pairs for the difference $(x \oplus \chi_0, \chi_1, \chi_2, \chi_3)$, where $x \to \theta_0$ by the S-box.

iv. If some quartet still remains at this point, assume that the subkey that it suggests is the correct one. Either perform an exhaustive key search on the remaining key bits, or use key ranking methods to find the right key.

4.6 Analysis of the Attack

We note that the first two tests of Step 2(c)(iii) can be done efficiently by computing for each pair independently the possible subkey values for which the condition is satisfied. Then, these tests are reduced to the problem of finding the intersection of these lists.

Once a right quartet is encountered, then it suggests the right value for 120 subkey bits (the relation between K_a and K_c suggests seven more bits of the key). Due to technical reasons, the time complexity of this search is 2^{112} (and not 2^{65} as might be expected).

For a given $K_{c'}$ value, out of the $2^{74} \cdot 2^{44} = 2^{118}$ quartets composed in Step 2(c)(ii), we expect 2^{60} quartets to reach Step 2(c)(iv) (a quartet has a probability of 2^{-7} to pass each of the first two tests, a probability of 2^{-8} to pass the third test, and a probability of 2^{-18} to pass each of the last two tests). The time complexity of the attack is $127 \cdot 2^{112} \cdot 2^{60} \approx 2^{179}$ encryptions.

If we take twice the data (i.e., 128 structures of 2^{72} encrypted under each key), we expect four right quartets. In that case, for any guess of the relation between K_a and $K_{c'}$, we shall perform the exhaustive search on the remaining key bits only if the same 120-bit value is suggested by two (or more) quartets. The time complexity in this case is dominated by the filtering done in Step 2(c), and is equal to 2^{118} encryptions for a given $K_{c'}$. Repeating the attack for every $K_{c'}$ (until one succeeds) has a worst-case time complexity of 2^{125} encryptions.

We conclude that the data complexity of our attack is 2^{87} chosen plaintexts, encrypted under 256 related keys (each key is used to encrypt 2^{79} values). The time complexity of the attack is 2^{125} 9-round encryptions.

The application of the attack to 10-round AES-256 is very similar. The attack AES-192). The data complexity of the attack is $2^{114.9}$ related-key chosen plaintexts (encrypted under 256 related keys) and the time complexity is $2^{171.8}$ encryptions.

5 Related-Key Boomerang Distinguisher for COCONUT98

COCONUT98 is a block cipher built according to the decorrelation methodology [22]. It contains two 4-round Feistel constructions with a decorrelation module in between. The differential (and linear) behavior of the decorrelation module is optimal in the sense that given a non-zero input difference, the probability to get any non-zero output difference is equal, when taken over all the keys. However, for any given key, the decorrelation module is an affine permutation.

The key schedule algorithm of COCONUT98 takes 256-bit keys, and divides them into two parts: a 128-bit subkey that enters the decorrelation module, and four 32-bit values, denoted by K_1, K_2, K_3, and K_4 that are used to derive the eight subkeys for the Feistel rounds. The subkeys of the first 4-round Feistel construction are: $K_1, K_1 \oplus K_3, K_1 \oplus K_3 \oplus K_4, K_1 \oplus K_4$, and the subkeys of the last 4-round Feistel construction are $K_2, K_2 \oplus K_3, K_2 \oplus K_3 \oplus K_4, K_2 \oplus K_4$.

The round function of the Feistel construction of COCONUT98 is

$$\begin{aligned} F_i((x,y)) &= (y, x \oplus \phi((ROL_{11}(\phi(y \oplus k_i)) + c) \bmod 2^{32})) \\ \text{where} \qquad \phi(x) &= (x + 256 \cdot S(x \& FF_x)) \bmod 2^{32} \end{aligned}$$

where $S(\cdot)$ is an 8-bit to 24-bit S-box, c is a known 32-bit constant, & is the AND operator, and k_i is the 32-bit round subkey.

Our decomposition of COCONUT98 to sub-ciphers is as follows: the first sub-cipher consists of the first 4-round Feistel construction and the decorrelation module (denoted by DM). The second sub-cipher consists of the remaining 4-round Feistel construction.

For the first sub-cipher we use following related-key differential: Let the key difference be $\Delta K_0 = (0, 0, z, z, 0, 0, 0, 0)$, then the differential is:

$$(z, 0) \to (0, z) \to (z, 0) \to (0, z) \to (0, z) \xrightarrow{DM} (z_1, z_2)$$

for some two unknown fixed (z_1, z_2), with probability 1. This is due to the fact that in each round where a difference z enters the round function, there is a subkey difference z to cancel it.

The related-key differential for the second sub-cipher is similar — the key difference is $\Delta K_1 = (0, 0, z, z, 0, 0, 0, 0)$ and the second differential is:

$$(z, 0) \to (0, z) \to (z, 0) \to (0, z) \to (z, 0)$$

with probability 1.

We note that $\Delta K_0 = \Delta K_1$. Hence, $K_c = K_a \oplus \Delta K_1 = K_a \oplus \Delta K_0 = K_b$, and $K_d = K_b \oplus \Delta K_1 = K_a$.

Thus, to find whether a given black box is COCONUT98, one can use the following algorithm:

– Choose a non-zero value z of 32 bits. Choose a plaintext P_a.

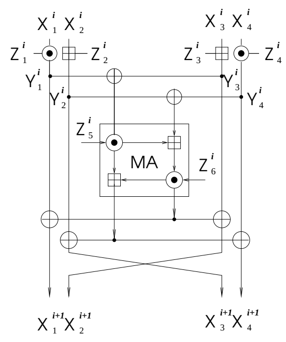

Fig. 4. One Round of IDEA

- Ask the encryption of P_a under the key K_a, and denote the ciphertext by C_a. Ask the encryption of $P_b = P_a \oplus (z, 0)$ under the related-key K_b, and denote the ciphertext by C_b.
- Compute $C_c = C_a \oplus (z, 0)$ and $C_d = C_b \oplus (z, 0)$.
- Ask the decryption of C_c under $K_c = K_b$ to obtain P_c. Ask the decryption of C_d under $K_d = K_a$ to obtain P_d.
- If $P_c \oplus P_d = (z, 0)$, then output COCONUT98.

We note that if the key of the decorrelation module is such that the input difference of $(z, 0)$ to the decorrelation module remains $(z, 0)$ after the module, then we get a related-key differential with probability 1 for the entire COCONUT98 cipher. Otherwise, as $(z, 0) \not\to (z, 0)$ we get a related-key impossible differential (due to the miss in the middle attack [3, 2]).

We conclude that COCONUT98 can be easily distinguished using one related-key adaptive chosen plaintext and ciphertext quartet under two keys. By using two different z values, one for the first differential, and one for the second differential, the distinguisher remains the same in nature, but uses four keys instead of two.

6 Related-Key Rectangle Attack on IDEA

IDEA [21] is a 64-bit block cipher with 128-bit keys. It uses a composition of XOR operations, additions modulo 2^{16} and multiplications over $GF(2^{16}+1)$. It has 8.5 rounds — 8 full rounds as described in Figure 4, and a final half-round consists of a layer of key additions and multiplications (Z_i^j are round subkeys). IDEA's key schedule is linear: each subkey is composed of bits of the key.

Since its introduction in 1991, IDEA has resisted a comprehensive cryptanalytic effort [10, 15, 2, 8, 13]. The best known attack against IDEA is on 5-round reduced version of IDEA. The attack uses 2^{24} chosen plaintexts and has a time complexity of 2^{116} encryptions [13]. IDEA also has several weak key classes. The largest weak key class (identified by a boomerang technique) contains 2^{64} keys, and the membership test requires 2^{16} adaptive chosen plaintexts and ciphertexts and has a time complexity of 2^{16} encryptions [8].

Basically, our rectangle attack on 6.5-round IDEA uses a 6-round boomerang attack. The 6.5 rounds that we attack, start before the first MA function.

6.1 A Related-Key Boomerang Distinguisher for 5.5-Round IDEA

The 5.5-round distinguisher is used in rounds 2–6.5 (from the second round). The decomposition of the 5.5-round IDEA into sub-ciphers is as follows: The first sub-cipher has three rounds, while the second sub-cipher has two and a half rounds.

The input difference to round 2 (and the first sub-cipher) is $\alpha = (0_x, 0_x, 8000_x, 0_x)$. The key difference $\Delta K_0 = e_{25}$ (where e_i is a difference only in the i'th bit). The input difference is cancelled by the subkey difference, and the zero difference remains with probability 1 up to the MA of round 4. The key difference enters Z_5^4, leading to an unknown β difference after the MA. However, there are at most 2^{32} β values after the MA, and in the worst case all of them are equiprobable with probability 2^{-32}. As we use all these differentials simultaneously, we obtain that $\hat{p} = 2^{-16}$.

The second differential has a similar structure (but in the backward direction). The output difference is $\delta = (0_x, 0_x, 0100_x, 0_x)$ and the key difference is $\Delta K_1 = e_{75}$. In the decryption of a pair with difference δ, the key difference cancels the difference with probability 1/2. If this is the case, the zero difference remains through the decryption up till the first half round of round 5. This time, there are only 2^{16} possible γ values, and in the worst case, all of them are equiprobable [2] with probability 2^{-17}. Again, we use all of them simultaneously, and thus, $\hat{q}^2 = 2^{16} \cdot 2^{-34} = 2^{-18}$, and $\hat{q} = 2^{-9}$. We note that there are 8 more δ values (and respective ΔK_1 value) for which the attack can be mounted.

This leads to a distinguishing attack on a 5.5-round IDEA using $2^{51.6}$ quartets of adaptive chosen plaintexts and ciphertexts ($2^{49.6}$ values are to be encrypted/decrypted under four keys).

6.2 A Related-Key Boomerang Attack on 6-Round IDEA

Let K_a be the unknown key, $K_b = K_a \oplus e_{25}$, $K_c = K_a \oplus e_{75}$, and $K_d = K_a \oplus e_{25} \oplus e_{75}$. The boomerang attack on six rounds of IDEA (starting at the MA in round 1 till before the MA in round 7) is as follows:

1. For each guess of bits 64–95 of K_a set a counter initialized to 0.
2. Choose $2^{17.6}$ 32-bit value (r, t), and for each such value:
   - Choose a structure A of 2^{32} plaintexts of the form (x, y, z, w), such that $x \oplus z = t$ and $y \oplus w = r$.
   - Choose a structure B of 2^{32} plaintexts of the form $(x, y \oplus 8000_x, z, w)$, such that $x \oplus z = t$ and $y \oplus w = r$.

[2] We have checked the claim experimentally. The values are not equiprobable, and the true value is $\hat{q} > 2^{-8.8}$. For more than 99% of the keys $\hat{q} > 2^{-8}$.

- Ask for the encryption of the structure A under K_a to receive A' and similarly ask for the encryption of B under K_b to receive B'.
- For any ciphertext in A' compute its XOR with the output difference of the second differential to obtain C'. Ask for the decryption of C' under K_c to obtain C.
- For any ciphertext in B' compute its XOR with the output difference of the second differential to obtain D'. Ask for the decryption of D' under K_d to obtain D.
- Insert all the plaintexts in C and D to a hash table indexed by the XOR value of the first and third word and by the XOR value of the second and fourth words.
- Examine a pair of colliding plaintexts (P_c, P_d). Let P_a be the plaintext that was encrypted, δ-shifted, and decrypted to P_c, and let P_b be the plaintext that was encrypted, δ-shifted and decrypted to P_d. For any guess of the bits 64–95 of K_a:
  (a) Partially encrypt P_a, P_b, P_c, P_d through the first MA. If the differences of the partial encryptions of P_a and P_b, and of the pair P_c and P_d are both α continue the analysis, if not so, try the next subkey.
  (b) Verify that the difference after a partial decryption of the respective ciphertext pairs is zero (bits 64–95 of the subkey contain the entire subkey which deals with the third word of the ciphertext). If this is the case, increment the counter of the subkey.

3. Output the subkey whose counter is maximal.

The structures are chosen so that in each pair of structures A, B there are 2^{32} pairs with input α after the first MA. For each such pair of structures we expect 2^{32} pairs of plaintexts (P_c, P_d) that are analyzed. Under random distribution assumptions, 2^{32} quartets from each pair of structures are encountered. However, most of them are discarded, and wrong quartet has probability of 2^{-32} to agree on the subkey of the first MA. Hence, we have about $2^{17.6}$ quartets in total, each suggesting 32-bit subkey value. The second filtering done, reduces this number by a factor of four.

We conclude that the attack suggests $2^{15.6}$ possible values to 32 bits of the key. As we expect four right quartets, then the right value is expected to be with the maximal counter with very high probability. The data complexity of the attack is $2^{51.6}$ adaptive chosen plaintexts and ciphertexts. The time complexity of the attack is $2^{51.6}$ MA evaluations which are equivalent to about 2^{48} encryptions.

6.3 A Related-Key Rectangle Attack on 6.5-round IDEA

The attack can be extended to a rectangle attack on 6.5-round IDEA. The 6.5 rounds starts after the first half round, and end after round 7. We use the same differentials as before. The algorithm of the attack is as follows:

- Choose $2^{25.8}$ values of the pair (r, t). Generate two structures of plaintexts for each value of (r, t) like in the boomerang attack.

- Ask the encryption of the structures under K_a and K_b, as before.
- Ask the encryption under K_c of each plaintext encrypted under K_a.
- Ask the encryption under K_d of each plaintext encrypted under K_b.
- For each guess of the subkey of the last MA, partially decrypt all ciphertexts under the guessed subkey, and call the boomerang attack on 6 rounds.

Out of the $2^{57.8}$ plaintexts encrypted under each key, we get about about $2^{51.6}$ pairs with the differences required for the previous attack. The attack has time complexity of $2^{32} \cdot (4 \cdot 2^{57.8}/13 + 2^{51.6}/13) = 2^{88.1}$ 6.5-round IDEA encryptions, and data complexity of $2^{59.8}$ chosen plaintexts under four related-keys.

7 Summary and Conclusions

In this paper we introduced related-key boomerang attacks and related-key rectangle attacks. The attacks use weaknesses in the key schedule algorithms of ciphers to achieve significant improvements over ordinary boomerang and rectangle distinguishers.

It is commonly believed that linearity of the key schedule is not a threat to the security of a block cipher if only its design (except for the key schedule) is moderate enough. Many strong block ciphers use linear or close to linear key schedule algorithms, e.g., AES, and IDEA. Despite the strong related-key requirements, our attacks show that it is important to maintain some non-linearity in the key schedule, even if the other components of the cipher seem strong enough.

References

1. Eli Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Journal of Cryptology, vol. 7, number 4, pp. 229–246, Springer-Verlag, 1994.
2. Eli Biham, Alex Biryukov, Adi Shamir, *Miss in the Middle Attacks on IDEA and Khufu*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 124–138, Springer-Verlag, 1999.
3. Eli Biham, Alex Biryukov, Adi Shamir, *Cryptanalysis of Skipjack Reduced to 31 Rounds*, Advances in Cryptology, proceedings of EUROCRYPT '99, Lecture Notes in Computer Science 1592, pp. 12–23, Springer-Verlag, 1999.
4. Eli Biham, Orr Dunkelman, Nathan Keller, *The Rectangle Attack – Rectangling the Serpent*, Advances in Cryptology, proceeding of EUROCRYPT 2001, Lecture Notes in Computer Science 2045, pp. 340–357, Springer-Verlag, 2001.
5. Eli Biham, Orr Dunkelman, Nathan Keller, *New Results on Boomerang and Rectangle Attacks*, proceeding of Fast Software Encryption 9, Lecture Notes in Computer Science 2365, pp. 1–16, Springer-Verlag, 2002.
6. Eli Biham, Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
7. Alex Biryukov, *The Boomerang Attack on 5 and 6-round AES*, pre-proceedings of Advanced Encryption Standard 4, available on-line at *http://www.esat.kuleuven.ac.be/~abiryuko/*.
8. Alex Biryukov, Jorge Nakahara Jr., Bart Preneel, Joos Vandewalle *New Weak-Key Classes of IDEA*, proceedings of Information and Communications Security 4, Lecture Notes in Computer Science 2513, pp. 315–326, Springer-Verlag, 2002.

9. Alex Biryukov, David Wagner, *Slide Attacks*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 245–259, Springer-Verlag, 1999.
10. Johan Borst, Lars R. Knudsen, Vincent Rijmen, *Two Attacks on Reduced Round IDEA*, Advances in Cryptology, proceedings of EUROCRYPT '97, Lecture Notes in Computer Science 1233, pp. 1–13, Springer-Verlag, 1997.
11. Joan Daemen, Lars R. Knudsen, Vincent Rijmen, *The Block Cipher Square*, proceedings of Fast Software Encryption 4, Lecture Notes in Computer Science 1267, pp. 149–165, Springer-Verlag, 1997.
12. Joan Daemen, Vincent Rijmen *The design of Rijndael: AES — the Advanced Encryption Standard*, Springer-Verlag, 2002.
13. Hüseyin Demirci, Ali A. Selçuk, Erkan Türe, *A New Meet-in-the-Middle Attack on the IDEA Block Cipher*, proceedings of Selected Areas in Cryptography 2003, Lecture Notes in Computer Science 3006, pp. 117–129, Springer-Verlag, 2004.
14. Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner, Doug Whiting, *Improved Cryptanalysis of Rijndael*, proceeding of Fast Software Encryption 8, Lecture Notes in Computer Science 1978, pp. 213–230, Springer-Verlag, 2001.
15. Philip Hawkes, *Differential-Linear Weak Keys Classes of IDEA*, Advances in Cryptology, proceedings of EUROCRYPT '98, Lecture Notes in Computer Science 1403, pp. 112–126, Springer-Verlag, 1998.
16. Seokhie Hong, Jongsung Kim, Guil Kim, Sangjin Lee, Bart Preneel, *Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192*, proceedings of Fast Software Encryption 12, to appear.
17. 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.
18. John Kelsey, Tadayoshi Kohno, Bruce Schneier, *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent*, proceedings of Fast Software Encryption 7, Lecture Notes in Computer Science 1978, pp. 75–93, Springer-Verlag, 2000.
19. John Kelsey, Bruce Schneier, David Wagner, *Related-Key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA*, proceedings of Information and Communication Security 1997, Lecture Notes in Computer Science 1334, pp. 233–246, Springer-Verlag, 1997.
20. Jongsung Kim, Guil Kim, Seokhie Hong, Dowon Hong, *The Related-Key Rectangle Attack — Application to SHACAL-1*, proceedings of ACISP 2004, Lecture Notes in Computer Science 3108, pp. 123–136, Springer-Verlag, 2004.
21. Xuejia Lai, James L. Massey, *A Proposal for a New Block Cipher Encryption Standard*, Advances in Cryptology, proceeding of EUROCRYPT '90, Lecture Notes in Computer Science 473, pp. 389–404, Springer-Verlag, 1991.
22. Serge Vaudenay, *Provable Security for Block Ciphers by Decorrelation*, proceedings of Annual Symposium on Theoretical Aspects of Computer Science '98, Lecture Notes in Computer Science 1373, pp. 249–275, Springer-Verlag, 1998.
23. David Wagner, *The Boomerang Attack*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 156–170, Springer-Verlag, 1999.