Meet-in-the-Middle Attacks on Reduced-Round XTEA⋆

Gautham Sekar⋆⋆, Nicky Mouha⋆⋆⋆, Vesselin Velichkov†, and Bart Preneel

[1] Department of Electrical Engineering ESAT/SCD-COSIC,
Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium.
[2] Interdisciplinary Institute for BroadBand Technology (IBBT), Belgium.
{Gautham.Sekar,Nicky.Mouha,Vesselin.Velichkov,
Bart.Preneel}@esat.kuleuven.be

Abstract. The block cipher XTEA, designed by Needham and Wheeler, was published as a technical report in 1997. The cipher was a result of fixing some weaknesses in the cipher TEA (also designed by Wheeler and Needham), which was used in Microsoft's Xbox gaming console. XTEA is a 64-round Feistel cipher with a block size of 64 bits and a key size of 128 bits. In this paper, we present meet-in-the-middle attacks on twelve variants of the XTEA block cipher, where each variant consists of 23 rounds. Two of these require only 18 known plaintexts and a computational effort equivalent to testing about 2^{117} keys, with a success probability of $1-2^{-1025}$. Under the standard (single-key) setting, there is no attack reported on 23 or more rounds of XTEA, that requires less time and fewer data than the above. This paper also discusses a variant of the classical meet-in-the-middle approach. All attacks in this paper are applicable to XETA as well, a block cipher that has not undergone public analysis yet. TEA, XTEA and XETA are implemented in the Linux kernel.

Keywords: Cryptanalysis, block cipher, meet-in-the-middle attack, Feistel network, XTEA, XETA.

1 Introduction

Timeline: the TEA family of block ciphers

– **1994.** The cipher TEA (Tiny Encryption Algorithm) is a 64-round Feistel cipher that operates on 64-bit blocks and uses a 128-bit key. Designed by Wheeler and Needham, it was presented at FSE 1994 [23]. Noted for its

⋆ This work was supported in part by the Research Council K.U.Leuven: GOA TENSE, and by the IAP Program P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and in part by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II.
⋆⋆ This author is supported by an FWO project.
⋆⋆⋆ This author is funded by a research grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).
† DBOF Doctoral Fellow, K.U.Leuven, Belgium.

simple design, the cipher was subsequently well studied and came under a number of attacks.

- **1996.** Kelsey *et al.* established that the effective key size of TEA was 126 bits [11]. This result led to an attack on Microsoft's Xbox gaming console where TEA was used as a hash function [22].
- **1997.** Kelsey, Schneier and Wagner constructed a related-key attack on TEA with 2^{23} chosen plaintexts and 2^{32} time [12]. Following these results, TEA was redesigned by Needham and Wheeler to yield Block TEA and XTEA (eXtended TEA) [17]. While XTEA has the same block size, key size and number of rounds as TEA, Block TEA caters to variable block sizes for it applies the XTEA round function for several iterations. Both TEA and XTEA are implemented in the Linux kernel.
- **1998.** To correct weaknesses in Block TEA, Needham and Wheeler designed Corrected Block TEA or XXTEA, and published it in a technical report [18]. This cipher uses an unbalanced Feistel network and operates on variable-length messages. The number of rounds is determined by the block size, but it is at least six. An attack on the full Block TEA is presented in [19], where some weaknesses in XXTEA are also detailed.
- **2002–2010.** A number of cryptanalysis results on the TEA family were reported in this period. Table 1 lists the attacks on XTEA and their complexities. In [10], it was shown that an ultra-low power implementation of XTEA might be better suited for low resource environments than AES. Note that XTEA's smaller block size also makes it advantageous if an application requires fewer than 128 bits of data to be encrypted at a time.

The meet-in-the-middle attack. The meet-in-the-middle attack was first introduced by Diffie and Hellman in 1977 [5]. Since then, this technique and its variants have been successfully used against several block ciphers, including reduced-round DES [4, 6]. Unlike Diffie and Hellman's original attack, the meet-in-the-middle attacks in this paper[3] have negligible memory requirements.

We denote the message space and the key space by $\mathcal{M}$ and $\mathcal{K}$ respectively. Now consider two block ciphers A_K, $B_K : \mathcal{M} \times \mathcal{K} \to \mathcal{M}$ and let $Y_K = B_K \circ A_K$, where $\circ$ denotes function composition. In a meet-in-the-middle attack, the adversary deduces K from a given plaintext-ciphertext pair (p, c), where $c = Y_K(p)$, by solving the equation

$$A_K(p) = B_K^{-1}(c) \ . \tag{1}$$

Contribution of this paper. This paper presents meet-in-the-middle attacks on block ciphers with 7, 15 and 23 rounds of XTEA. Our attacks are under the

[3] The attack presented in Sect. 5 of this paper can also be seen as a meet-in-the-middle attack, however the (partial) encryptions and decryptions cannot be performed over all rounds, as the attacker only searches exhaustively over parts of the key. We therefore use a technique similar to the partial matching technique of Sasaki and Aoki. This very recent technique was successfully applied to several hash functions, including MD4 [2], MD5 [20], HAS-160 [9] and SHA-2 [1].

Table 1. Key recovery attacks on XTEA where the time complexities are averages, if explicitly stated in the original paper, average success probabilities are given as well (KP: known plaintext, CP: chosen plaintext, RK: in a related-key setting)

Attack	**Ref.**	**# Rounds**	**Time**	**Data**	**Pr[Success]**
• Attacks in the standard (single-key) setting					
Meet-in-the-middle	**This paper**	**7**	$\mathbf{2^{95.00}}$	**2 KPs**	$\mathbf{1 - 2^{-33}}$
Impossible differential	[16]	14	2^{85}	$2^{62.5}$ CPs	Not given
Differential	[8]	15	2^{120}	2^{59} CPs	Not given
Meet-in-the-middle	**This paper**	**15**	$\mathbf{2^{95.00}}$	**3 KPs**	$\mathbf{1 - 2^{-65}}$
Truncated differential	[8]	23	$2^{120.65}$	$2^{20.55}$ CPs	0.969
Meet-in-the-middle	**This paper**	**23**	$\mathbf{2^{117.00}}$	**18 KPs**	$\mathbf{1 - 2^{-1025}}$
• Attacks in a related-key setting					
Related-key truncated differential	[13]	27	$2^{115.15}$	$2^{20.5}$ RK-CPs	0.969
Related-key rectangle (for $2^{108.21}$ weak keys)	[14]	34	$2^{31.94}$	2^{62} RK-CPs	Not given
Related-key rectangle	[15]	36	$2^{126.44}$	$2^{64.98}$ RK-CPs	0.63
Related-key rectangle (for $2^{110.67}$ weak keys)	[15]	36	$2^{104.33}$	$2^{63.83}$ RK-CPs	0.80
Related-key	[3]	37	2^{125}	2^{63} RK-CPs	Not given
Related-key (for $2^{107.5}$ weak keys)	[3]	51	2^{123}	2^{63} RK-CPs	Not given

standard setting, giving the attacker less freedom than under a related-key setting. In Table 1, we see that there is no attack on 23 or more rounds of XTEA, that is better than ours given the standard setting. Furthermore, each of our attacks requires only a few *known plaintexts*, whereas every attack listed in Table 1 requires many *chosen plaintexts*.

The Linux kernel not only includes XTEA, but also a variant called XETA [7]. The cipher XETA resulted from a bug in the C implementation of XTEA, where higher precedence was incorrectly given to exclusive-OR over addition in the round function. From this paper, it is easy to verify that all our results to XTEA directly apply to XETA as well. This is because our attacks exploit weaknesses in the key schedule, which is the same for both XTEA and XETA. To the best of our knowledge, this paper is the first to give cryptanalysis results on XETA.

Organization. This paper is organized as follows. Section 2 lists the notation and convention that we follow. The description of XTEA is provided in Sect. 3. Our main observation is presented in Sect. 4 and it is developed into an attack on 15-round XTEA in Sect. 5. Here, we also provide other sets of 15 rounds that could be similarly attacked. Section 6 describes our attack on 23 rounds on XTEA and provides other sets of 23 rounds that could be attacked in a similar way. Section 7 concludes the paper and provides an interesting open problem. In Appendix A, we show which countermeasures can be introduced to XTEA to prevent all the attacks in this paper. The 23-round attack is illustrated in Appendix B.

2 Notation and Convention

The notation used in this paper is listed in Table 2.

Table 2. Notation

Symbol / Notation	Meaning
$\boxplus$	Addition modulo 2^{32}
$\oplus$	Exclusive-OR
$\lll$	Left shift
$\ggg$	Right shift
$\|\|$	Concatenation
$\lfloor x \rfloor$	$\max_{y \in \mathbb{Z}}(y \leq x)$, $\mathbb{Z}$ is the set of integers
LSB	Least significant bit
MSB	Most significant bit
$[i]$	Select bit i, $i = 0$ is the LSB
$[j \ldots i]$	Select bits k where $j \geq k \geq i$, $k = 0$ is the LSB
0^k	Concatenation of k times the string '0'

3 Description of XTEA

The block cipher XTEA has block size of 64 bits and key size of 128 bits. It uses a 64-round Feistel network (see Fig. 1). The F-function of the Feistel network (see Fig. 2) takes a 32-bit input x and produces a 32-bit output as:

$$F(x) = ((x \lll 4) \oplus (x \ggg 5)) + x \ . \tag{2}$$

The 128-bit key K of XTEA is divided into four 32-bit subkeys $K_0, \ldots, K_3$. At every round, one of the 4 subkeys is selected according to a key schedule. A constant $\delta = \lfloor (\sqrt{5} - 1) \cdot 2^{31} \rfloor$ is defined, derived from the golden ratio. Two bits from a different multiple of δ are used at every round as the index of the subkey. The 32-bit subkey α_t used in round t, where $1 \leq t \leq 64$, is chosen from the set $\{K_0, K_1, K_2, K_3\}$ according to the following rule:

$$\alpha_t \leftarrow \begin{cases} K_{\delta_t[1 \ldots 0]} & \text{if } t \text{ is odd} \ , \\ K_{\delta_t[12 \ldots 11]} & \text{if } t \text{ is even} \ , \end{cases} \tag{3}$$

where

$$\delta_t = \left\lfloor \frac{t}{2} \right\rfloor \delta, \quad 1 \le t \le 64 \ . \tag{4}$$

The 64-bit input to round t of XTEA consists of two 32-bit parts L_{t-1} and R_{t-1} (see Fig. 1). For round 1, the plaintext p is used as input: $(L_0 \parallel R_0) \leftarrow p$. The input for round $t+1$ is computed recursively from the input to round t as given by:

$$L_t \leftarrow R_{t-1} \ , \tag{5}$$

$$R_t \leftarrow L_{t-1} \boxplus ((\delta_t \boxplus \alpha_t) \oplus F(R_{t-1})) \ , \tag{6}$$

where α_t is selected according to (3). For reference, we also list the subkeys used in every round in Table 3.

The ciphertext c of XTEA is produced by concatenating the two parts obtained after the 64th round: $c \leftarrow L_{64} \parallel R_{64}$.

Finally, we note that in the description above by *round* we mean a *Feistel round*. This is not to be confused with the term *cycle* used in the original proposal of XTEA [17]. A cycle is equivalent to two Feistel rounds. Therefore XTEA has 64 rounds or 32 cycles.

Table 3. Subkeys used in XTEA

Rounds	**Subkey used**
1, 8, 9, 10, 17, 18, 20, 25, 30, 33, 40, 41, 49, 50, 57, 60	K_0
3, 6, 11, 16, 19, 26, 27, 28, 35, 36, 38, 43, 46, 48, 51, 58, 59	K_1
4, 5, 13, 14, 21, 24, 29, 34, 37, 44, 45, 53, 54, 56, 61, 64	K_2
2, 7, 12, 15, 22, 23, 31, 32, 39, 42, 47, 52, 55, 62, 63	K_3

4 Motivational Observation

We begin by observing that the subkey K_2 is not used in rounds 6–12. For the remainder of this section, let $K \leftarrow (K_0, K_1, X, K_3)$, where X can be any 32-bit value, as subkey K_2 is irrelevant in the analysis. Given one plaintext-ciphertext pair (p_0, c_0), with each key guess, the attacker checks whether

$$E_K^{(6\ldots12)}(p_0) = c_0 \ , \tag{7}$$

where $E_K^{(6\ldots12)}$ denotes the 7-round (rounds 6–12) encryption using the key K. At first glance, it may appear that 1 KP is sufficient. However, it is to be noted that the key space (2^{96} keys K) is larger than the ciphertext space (2^{64} ciphertext blocks).

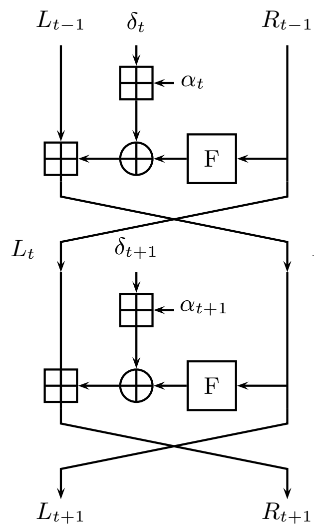

Fig. 1. The Feistel structure of XTEA showing two rounds

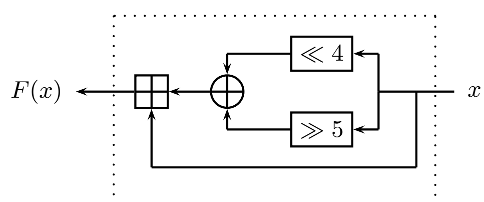

Fig. 2. The function F used in the round function of XTEA

We now show that obtaining a second KP (p_1, c_1) is sufficient for an attack with an average time complexity of $2^{95.00}$ 7-round encryptions and an average success probability of $1 - 2^{-33}$. The attacker iterates over the 2^k keys K, where $k = 96$. For every candidate key K, (7) is tested using the first KP. If this equality is satisfied, the second KP is used to check

$$E_K^{(6\ldots12)}(p_1) = c_1 \quad . \tag{8}$$

If either (7) or (8) is not satisfied, the candidate key K is incorrect and can be *sieved*. The approximate number of plaintext-ciphertext pairs that are needed can also be estimated from Shannon's unicity distance [21].

We make the reasonable assumption throughout this paper, that every block cipher we consider has perfect confusion and diffusion properties [21]. If either the plaintext or the key, or both are changed, it is assumed that the corresponding ciphertext will be generated uniformly at random, independent from previously obtained ciphertexts.

Under this assumption, each of the 64-bit conditions that result from (7) and (8) is satisfied with probability 2^{-64}. All time complexities are stated as the number of equivalent encryptions of the reduced-round block cipher.

The average success probability can be calculated as follows. The two 64-bit conditions are simultaneously satisfied with probability $2^{-2\cdot 64} = 2^{-128}$. We can therefore eliminate a wrong key with probability $1 - 2^{-128}$. Assume that key i is the correct key, where $0 \leq i < 2^k$. It will be output by the algorithm if all previous keys are eliminated. This happens with probability $(1 - 2^{-128})^i$. The correct key can be located anywhere among the list of 2^k candidate keys with equal probability. Therefore, the average success probability is

$$2^{-k} \cdot \sum_{i=0}^{2^k-1} (1 - 2^{-128})^i = 2^{128-k} \cdot (1 - (1 - 2^{-128})^{2^k}) \approx 2^{128-k} \cdot (1 - e^{-2^{k-128}}) \approx 1 - 2^{-33} \ . \tag{9}$$

The approximations result from using the first and the second order Taylor approximations of e^x around 0. We now calculate the time complexity of the attack. For a candidate key K to be determined as wrong, the expected number of trials is $1 + 2^{-64}$. This is because for every key, (7) is always checked, and for 2^{-64} keys (8) is checked as well. If the candidate key is correct, two encryptions are always performed. As the correct key can be located anywhere in the list of 2^k candidates keys with equal probability, the average number of encryptions of the algorithm is

$$2^{-k} \cdot \sum_{i=0}^{2^k-1} \big(i \cdot (1 + 2^{-64}) + 2\big) = 2^{-1} \cdot (1 + 2^{-64}) \cdot (2^k - 1) + 2 \approx 2^{95.00} \ . \tag{10}$$

From Table 3, we obtain several other 7-round block ciphers that can be attacked in a similar way. Table 4 lists all such ciphers. Finally, we note that for

Table 4. All 7-round attacks; each attack requires 2 KPs and on average $2^{95.00}$ 7-round encryptions for an average success probability of $1 - 2^{-33}$

Cipher consisting of XTEA rounds	**Unused subkey**
6–12	K_2
24–30	K_3
42–48	K_0
46–52	K_2

$n = 0$ and $n = 1$ respectively, one can replace both (7) and (8) with

$$E_K^{(6\ldots r-1)}(p_n) = D_K^{(r\ldots 12)}(c_n) \ , \tag{11}$$

where $r \in \{6, \ldots, 12\}$, $E_K^{(6\ldots5)}(p_n) = p_n$, and $D_K^{(r\ldots12)}$ denotes (13-r)-round (rounds r–12) decryption using the key K. Therefore, what we essentially constructed above can be viewed as meet-in-the-middle attacks. In (11), the value of r determines the subkeys that are required for encryption and decryption.

5 Attacks on 15 Rounds of XTEA

The attack described in Sect. 4 on rounds 6–12, can be extended to rounds 6–20 as follows. First, the attacker performs a meet-in-the-middle attack, where (partial) encryptions and decryptions cannot be performed over all rounds, the attacker only exhaustively searches over part of the key. From the remaining rounds, however, the number of possibilities for the full key is reduced. Only three known plaintexts (p_n, c_n), $0 \leq n < 2$ are required for the attack.

Let us now split a reduced-round XTEA block cipher into *outer rounds* and *inner rounds*. In the outer rounds, one particular subkey is not used, whereas the inner rounds use only this subkey. The attack is described for rounds 6–20. As can be seen from Table 3, the outer rounds (6–12) and (15–20) do not involve K_2, whereas the two inner rounds (13–14) use only K_2.

By encrypting plaintext p_0 from round 6 to round 12 (i.e., until the beginning of round 13) and decrypting the corresponding ciphertext c_0 for 6 rounds starting backwards from round 20, we obtain the subkeys used in the inner rounds. They are denoted as $K_2^{'}$ and $K_2^{''}$ for inner rounds 13 and 14 respectively. Then, the attacker checks whether $K_2^{'} = K_2^{''}$. This can be understood from Fig. 1. Therefore, not the ciphertext values (as in Sect. 4), but the key values "meet in the middle". To the best of our knowledge, such an approach has not been described in previous literature.

If $K_2^{'} \neq K_2^{''}$, the candidate key of (K_0, K_1, K_3) cannot be correct, and the attacker proceeds to the next candidate key. Otherwise, the candidate key is extended to (K_0, K_1, K_2, K_3), where $K_2 = K_2^{'} = K_2^{''}$. Then, the meet-in-the-middle attack is performed as described in Sect. 4. That is, a plaintext is encrypted with candidate keys (K_0, K_1, K_2, K_3), to check which of the computed ciphertexts agrees with the actual (corresponding) ciphertext. For the 15-round attack, it is sufficient to use two additional known plaintexts (p_1, c_1) and (p_2, c_2).

The average success probability can be calculated as follows. Using (p_0, c_0) a 32-bit condition is obtained when $K_2^{'} = K_2^{''}$ is checked. Then, (p_1, c_1) and (p_2, c_2) each gives an additional 64-bit condition. A wrong key will pass these tests with probability[4] $2^{-32} \cdot \left(2^{-64}\right)^2 = 2^{-160}$. Thus, with probability $1 - 2^{-160}$, a wrong key is eliminated. Assume that i is the correct key, where $0 \leq i < 2^k$. It will be output by the algorithm if all previous keys are eliminated. This happens with probability $(1 - 2^{-160})^i$. The correct key can be located anywhere among the list

[4] If the texts obtained by encrypting p_0 and decrypting c_0, in the 13 outer rounds, are uniformly distributed at random, then so are the subkeys $K_2^{'}$ and $K_2^{''}$. This fact, explained in Appendix C, is explicitly stated here because the assumption of perfect confusion and diffusion was made for ciphertexts, and not for subkeys.

of 2^k candidate keys with equal probability. The average success probability is

$$2^{-96} \cdot \sum_{i=0}^{2^{96}-1} (1-2^{-160})^i = 2^{160-96} \cdot (1-(1-2^{-160})^{2^{96}}) \approx 2^{64} \cdot (1-e^{-2^{64}})$$
$$\approx 1 - 2^{-65} \ . \tag{12}$$

We now calculate the time complexity of the attack. For a candidate key (K_0, K_1, K_3) to be determined as wrong, the expected number of trials is $1 + 2^{-32} + 2^{-96}$. This is because for every candidate key (K_0, K_1, K_3), the attacker always checks whether $K_2^{'} \neq K_2^{''}$. For 2^{-32} and 2^{-96} candidate keys, the attacker encrypts using the second and third known plaintext respectively. If the candidate key is correct, the equivalent of three encryptions is always performed. As the correct key can be located anywhere in the list of 2^{96} candidates keys with equal probability, the average number of (equivalent) encryptions of the algorithm is

$$2^{-96} \cdot \sum_{i=0}^{2^{96}-1} \big(i \cdot (1+2^{-32}+2^{-96})+3\big) = 2^{-1} \cdot (1+2^{-32}+2^{-96}) \cdot (2^{96}-1) + 3$$
$$\approx 2^{95.00} \ . \tag{13}$$

Finally, in Table 5, we provide a list of all 15-round block ciphers that can be attacked with the same complexity.

Table 5. All 15-round attacks; each attack requires 3 KPs and on average $2^{95.00}$ computations of the 15 rounds for an average success probability of $1-2^{-65}$

Cipher consisting of XTEA rounds	Inner rounds	Inner round subkey
6–20	13,14	K_2
16–30	22,23	K_3
24–38	31,32	K_3
34–48	40,41	K_0
38–52	44,45	K_2
42–56	49,50	K_0

6 Attacks on 23 Rounds of XTEA

In this section, we extend the 15-round attack of Sect. 5 to 23 rounds. This 23-round attack has an average time complexity of $2^{117.00}$ (equivalent) encryptions and an average success probability of $1-2^{-1025}$. It requires only 18 known (not chosen) plaintexts and corresponding ciphertexts. For the same number of rounds, both the time complexity and the data complexity of our attack are much lower than those in [8]. Our attack is therefore the best attack on 23-round XTEA so far in the standard setting, and the only attack requiring such a low number of plaintexts and corresponding ciphertexts. We note that we have optimized our attack to have the time complexity as low as possible. It is possible to reduce the number of known plaintexts even further, but not without increasing the time complexity of the attack.

The technique used is a meet-in-the-middle attack, similar to the attacks in [4]. As in Sect. 5, the reduced-round XTEA block cipher is split into *outer rounds* and *inner rounds*. In the outer rounds, one subkey is not used. The inner rounds can contain any of the subkeys. Our attack applies to rounds 16–38 of XTEA. Rounds 16–21 and 33–38 are the outer rounds, and do not involve subkey K_3. The inner rounds are rounds 22–32. The attack is a *sieving attack*, as the correct key is found by eliminating keys that lead to contradictions. The attack is given in Algorithm 1.

The k-bit key is recovered in two stages. First, the attacker exhaustively searches over k_1 bits of the key K and use m known plaintexts to check a one-bit condition that each of the m plaintexts yield. These k_1 bits consist of K_0, K_1, K_2, and the 21 least significant bits of K_3. This one-bit condition, tested in `test_keys_1(`K`)`, results from the following observation, also illustrated in Appendix B. We see that, without using $K_3[31 \ldots 21]$, the attacker can calculate $L_{27}[0] \leftarrow E_K^{(16\ldots27)}(p)[0]$, and $L'_{27}[0] \leftarrow D_K^{(28\ldots38)}(c)[0]$. As $L_{27}[0] = L'_{27}[0]$ always holds if the candidate key K is correct, a wrong key can be discarded if $L_{27}[0] \neq L'_{27}[0]$. Note that only k_1 bits of the candidate key K are used to test this condition, as the remaining k_2 bits do not affect this condition.

If none of the m plaintexts cause a key to be discarded, the attacker exhaustively searches over the remaining k_2 bits of key K in `test_keys_2(`K`)`. These k_2 bits are the 11 most significant bits of K_3. In this stage, $\ell \leq m$ of the m plaintexts are reused. Now, $(L_{27}, R_{27}) \leftarrow E_K^{(16\ldots27)}(p)$ and $(L'_{27}, R'_{27}) \leftarrow D_K^{(28\ldots38)}(c)$ are recalculated using the full key K. For efficiency, this calculation is sped up by using stored values $p_n^\star$ and $c_n^\star$ for the outer rounds, and encrypting only the inner rounds. Equations $L_{27} = R_{27}$ and $L'_{27} = R'_{27}$ yield only 63-bit conditions, as $L_{27}[0] = L'_{27}[0]$ was already tested. If both equations are satisfied for all ℓ plaintexts, the candidate key K is output as the correct key, and the algorithm halts.

Let us now determine the average time complexity and the average success probability of Algorithm 1.

The algorithm succeeds if no wrong key K that passes all $m + \ell$ tests is encountered before the correct key. How efficiently the attacker searches through these candidate keys K, does not influence the success probability of Algorithm 1. We therefore assume that the exhaustive search is over 2^k keys, and then both `test_keys_1(`K`)` and `test_keys_2(`K`)` are performed for each of these keys.

Each of the m plaintexts yields a one-bit condition in `test_keys_1(`K`)`, satisfied randomly with a probability of 2^{-1}. When $\ell \leq m$ of these plaintexts are reused in `test_keys_2(`K`)`, there is a condition on the 63 remaining bits, sat-

Algorithm 1 Recovering the key of the 23-round XTEA block cipher consisting of rounds 16–38; an average $2^{117.00}$ (equivalent) encryptions and 18 KPs are required for an average success probability of $1 - 2^{-1025}$

Require: m known plaintexts $p_0 \dots p_{m-1}$ and corresponding ciphertexts $c_0 \dots c_{m-1}$.
Ensure: The output key K (of length k bits) is the correct key with probability $2^{m+63\ell-k}(1 - e^{-2^{k-m-63\ell}})$, where ℓ is chosen such that $\ell \leq m$.

global $p_0^\star \dots p_{m-1}^\star$, $c_0^\star \dots c_{m-1}^\star$.
function test_key_1(K) **do**
  for $n \leftarrow 0 \dots m-1$ **do**
    $p_n^\star \leftarrow E_K^{(16\dots21)}(p_n)$
    $c_n^\star \leftarrow D_K^{(33\dots38)}(c_n)$
    $(L_{27}, R_{27}) \leftarrow E_K^{(22\dots27)}(p_n^\star)$
    $(L'_{27}, R'_{27}) \leftarrow D_K^{(28\dots32)}(c_n^\star)$
    if $L_{27}[0] \neq L'_{27}[0]$ **then**
      return *false*
  return *true*
function test_key_2(K) **do**
  for $n \leftarrow 0 \dots \ell-1$ **do**
    $(L_{27}, R_{27}) \leftarrow E_K^{(22\dots27)}(p_n^\star)$
    $(L'_{27}, R'_{27}) \leftarrow D_K^{(28\dots32)}(c_n^\star)$
    if $L_{27} \neq L'_{27}$ **or** $R_{27} \neq R'_{27}$ **then**
      return *false*
  return *true*
for $(K_0, K_1, K_2) \leftarrow (0 \dots 2^{32}-1, 0 \dots 2^{32}-1, 0 \dots 2^{32}-1)$ **do**
  for $K_3[20 \dots 0] \leftarrow 0 \dots 2^{21}-1$ **do**
    $K \leftarrow (K_0, K_1, K_2, 0^{11} \parallel K_3[20 \dots 0])^\dagger$
    if test_key_1(K) **then**
      for $K_3[31 \dots 21] \leftarrow 0 \dots 2^{11}-1$ **do**
        if test_key_2(K) **then**
          output K **and halt**

†Since the 11 bits $K_3[31 \dots 21]$ do not affect $L_{27}[0]$ or $L'_{27}[0]$, one can have any value β from the set $\{1, \dots, 2^{11}-1\}$ in place of 0^{11}. We have used 0^{11} for ease of understanding how the attack works.

isfied randomly with a probability of 2^{-63}. A wrong key will be detected if at least one of the $m + \ell$ tests fail. This eliminates a wrong key with a probability of $1 - 2^{-m} \cdot 2^{-63\ell}$. Assume that i is the correct key, where $0 \leq i < 2^k$. Then, it will be output by the algorithm if all previous candidate keys lead to contradictions. This happens with probability $(1 - 2^{-m} \cdot 2^{-63\ell})^i$. As the correct key can be located anywhere in the list of 2^k candidate keys with equal probability, the average success probability of the algorithm is

$$2^{-k} \cdot \sum_{i=0}^{2^k-1} (1 - 2^{-m} \cdot 2^{-63\ell})^i = 2^{m+63\ell-k} \cdot (1 - (1 - 2^{-m-63\ell})^{2^k})$$
$$\approx 2^{m+63\ell-k} \cdot (1 - e^{-2^{k-m-63\ell}}) \ . \qquad (14)$$

We now calculate the time complexity of the attack. Let i and j (where $0 \le i < 2^{k_1}$ and $0 \le j < 2^{k_2}$) be parts of the correct key K^c where $i = (K_0^c, K_1^c, K_2^c, K_3^c[20\ldots0])$ and $j = K_3^c[31\ldots21]$. Any 117-bit key $(K_0, K_1, K_2, K_3 [20\ldots0])$, tested in `test_keys_1(`$K$`)` before the correct key, passes `test_keys_1(`K`)` with probability 2^{-m}. Therefore, of the i 117-bit keys tested before the correct key, $i \cdot 2^{-m}$ keys are expected to pass `test_keys_1(`K`)`. For each of these $i \cdot 2^{-m}$ keys, `test_keys_2()` is performed 2^{k_2} times. Summarizing,

- the attacker performs an expected $i \cdot T_1$ 23-round computations, where T_1 is the expected number of 23-round computations for a wrong key under `test_keys_1()`;
- the attacker additionally performs an expected $i \cdot 2^{-m} \cdot 2^{k_2} \cdot T_2$ 23-round computations, where T_2 is the expected number of 23-round computations for a wrong key under `test_keys_2()`.

It is easy to see that

$$T_1 \triangleq \sum_{i=0}^{m-1} 2^{-i} \quad . \tag{15}$$

To compute T_2, note that `test_keys_2()` only encrypts the 11 inner rounds again, and uses stored values for (partial) encryptions and decryptions of the outer rounds. This is equivalent to 11/23 encryptions of the 23-round block cipher and therefore

$$T_2 \triangleq \frac{11}{23} \cdot \sum_{j=0}^{\ell-1} 2^{-63j} \quad . \tag{16}$$

For the correct (partial) key i, the number of steps under `test_keys_1()` is m. To determine the remaining part of the correct 128-bit key K^c, the attacker performs an expected $j \cdot T_2 + (11/23) \cdot \ell$ 23-round computations, where

1. $j \cdot T_2$ is the expected number of 23-round computations, under `test_keys_2()`, for all the j wrong (partial) keys preceding key j;
2. ℓ is the number of 11-round steps under `test_keys_2()` for the correct key j.

As the correct key j can take any value in the set $\{0, \ldots, 2^{k_2} - 1\}$, the average number of 23-round computations corresponding to the correct key i, is

$$2^{-k_2} \cdot \sum_{j=0}^{2^{k_2}-1} \left(j \cdot T_2 + \frac{11}{23} \cdot \ell \right) \quad . \tag{17}$$

As the correct key i can take any value in the set $\{0, \ldots, 2^{k_1} - 1\}$, the average number of 23-round computations in total is

$$2^{-k_1} \cdot \sum_{i=0}^{2^{k_1}-1} \left(i \cdot T_1 + m + i \cdot 2^{-m} \cdot 2^{k_2} \cdot T_2 + 2^{-k_2} \cdot \sum_{j=0}^{2^{k_2}-1} \left(j \cdot T_2 + \frac{11}{23} \cdot \ell \right) \right) \tag{18}$$

The derivation of (18) will be more clear from Fig. 3 in Appendix B.

We now choose the parameters m and ℓ for the attack on rounds 16–38. From (18), we find that we cannot lower the average time complexity below $2^{117.00}$. Therefore, we choose m and ℓ such that we have the lowest number of known plaintexts, and the highest success probability for this particular time complexity. Setting $m = \ell = 18$, we find that 18 KPs are sufficient, and that the corresponding success probability using (14) is $1 - 2^{-1025}$. Note that the success probability of exhaustive search over the full k-bit key using 18 KPs has the same success probability. This shows that all KPs are optimally used in our attack from an information theoretic point of view [21]. Note that the number of KPs can still be lowered further, but then the time complexity must increase. This can be done by either increasing ℓ (which would make the second stage dominate in the attack), or by increasing k_1 (and thus perform the meet-in-the-middle on more than one bit).[5] We do not consider such options, as the number of KPs in our attack is already low enough for a practical attack. The time complexity, however, is still beyond reach with current hardware. Each of these attacks requires only negligible memory (about $4 \cdot 64 \cdot 18 = 2^{12.17}$ bits to store (p_n, c_n) and $(p_n^\star, c_n^\star)$ values).

As shown in Table 6, a total of 12 variants of the XTEA block cipher can be attacked, where each variant consists of 23 rounds. For rounds 34–56, the attack works in exactly the same way as for 16–38, and has the same complexities. The 10 other attacks require that $k_1 = 122$: the exhaustive search is now over all but the 6 most significant bits of one subkey in Algorithm 1, in order to obtain a condition on one bit to perform the meet-in-the-middle. The *middle bit* involved in this condition is given as well in Table 6.

Using (18), we calculate the time complexity for the 10 attacks that use 12 or 13 inner rounds. The lowest possible average time complexity for our attack strategy is $2^{122.00}$. For this time complexity, the best parameters are $m = \ell = 13$. We then obtain an average success probability of $1 - 2^{-705}$, using 13 KPs. Again, each of these attacks requires only negligible memory (about $2^{11.70}$ bits to store (p_n, c_n) and $(p_n^\star, c_n^\star)$ values).

7 Conclusions and Open Problems

This paper presented several meet-in-the-middle attacks on 7-, 15- and 23-round XTEA. The main highlight of our attacks is that they require very few known plaintexts (not more than 18) as opposed to previously reported attacks (the best of these attacks requires 2^{20} chosen plaintexts). Furthermore, our attacks use different approaches - the 7- and 23-round attacks use a straightforward meet-in-the-middle approach; in the 15-round attacks, the meet-in-the-middle corresponds to inner round subkeys rather than intermediary text values.

[5] In the attack, one bit in the middle is independent of 11 key bits. Two bits in the middle are simultaneously independent of fewer than 11 key bits, thereby corresponding to a larger k_1.

Table 6. All 23-round attacks

Total rounds	Inner rounds	Middle bit	Unused key bits	# Inner rounds
16–38	22–32	$L_{27}[0]$	$K_3[31\dots21]$	11 rounds
34–56	40–50	$L_{45}[0]$	$K_0[31\dots21]$	11 rounds
6–28	13–24	$L_{19}[0]$	$K_2[31\dots26]$	12 rounds
8–30	12–23	$L_{18}[0]$	$K_3[31\dots26]$	12 rounds
24–46	31–42	$L_{37}[0]$	$K_3[31\dots26]$	12 rounds
26–48	30–41	$L_{36}[0]$	$K_0[31\dots26]$	12 rounds
30–52	34–45	$L_{40}[0]$	$K_2[31\dots26]$	12 rounds
42–64	49–60	$L_{55}[0]$	$K_0[31\dots26]$	12 rounds
20–42	26–38	$L_{32}[0]$	$K_1[31\dots26]$	13 rounds
38–60	44–56	$L_{50}[0]$	$K_2[31\dots26]$	13 rounds
2–24	8–20	$L_{14}[0]$	$K_0[31\dots26]$	13 rounds
12–34	16–28	$L_{22}[0]$	$K_1[31\dots26]$	13 rounds

Each of our attacks on 23-round XTEA requires less time ($2^{117.00}$ 23-round computations) than the previously best-known attack on 23 rounds ($2^{120.65}$ 23-round computations) in the standard setting. The time complexities of the 7- and 15-round attacks are also significantly better than exhaustive key search, with each of these attacks requiring about 2^{95} time.

Our attacks apply to XETA as well, a close variant of XTEA that is also implemented in the Linux kernel. We are unaware of any other published cryptanalysis results on XETA.

An interesting observation from one of the anonymous reviewers, is that there is also a 15-round attack on rounds 2–16. In this case, subkey K_0 is used consecutively in the inner rounds 8, 9 and 10, but not elsewhere. By exhaustively searching over K_1, K_2, K_3 and six of the least significant bits of K_0, we can perform the same meet-in-the-middle attack that is described in Sect. 6. However, this attack has a higher time and data complexity than the other 15-round attacks of Sect. 5, for a comparable success probability.

When constructing the 23-round attack in Sect. 6, we found that for any number of *inner rounds* (where all subkeys can be used) up to 16, there is no corresponding attack on more than 23 rounds. However, if the number of inner rounds can be increased to 17, this leads to a 29-round attack. All such 29-round attacks are listed in Table 7. We present the cryptanalysis of these 29-round XTEA block ciphers as an interesting open problem.

Acknowledgments. The authors would like to thank Tor E. Bjørstad, Gaëtan Leurent, Matt Robshaw and Aleksander Wittlin for their useful comments and suggestions. Part of this work was performed at the Cryptanalysis of Light-Weight Ciphers Research Meeting, hosted by Katholieke Universiteit Leuven as an initiative of SymLab-WG2: Lightweight Cryptography of the ECRYPT

Table 7. All reduced-round XTEA block ciphers for which a 29-round attack consists of 17 *inner rounds*

Total rounds	Inner rounds	Subkey containing unused key bits
11–39	27–33	K_0
15–43	21–37	K_2
29–57	35–51	K_1
33–61	40–56	K_3

II project. The authors would like to thank the anonymous reviewers for their constructive comments as well.

References

1. K. Aoki, J. Guo, K. Matusiewicz, Y. Sasaki, L. Wang, "Preimages for Step-Reduced SHA-2," *ASIACRYPT 2009* (M. Matsui, ed.), vol. 5912 of *LNCS*, pp. 578–597, Springer-Verlag, 2009.
2. K. Aoki, Y. Sasaki, "Preimage Attacks on One-Block MD4, 63-Step MD5 and More," *SAC 2008* (R. Avanzi, L. Keliher, F. Sica, eds.), vol. 5381 of *LNCS*, pp. 103–119, Springer-Verlag, 2009.
3. C. Bouillaguet, O. Dunkelman, G. Leurent, P.-A. Fouque, "Another Look at Complementation Properties," *FSE 2010* (S. Hong, T. Iwata, eds.), vol. 6147 of *LNCS*, pp. 347–364, Springer-Verlag, 2010.
4. D. Chaum, J.-H. Evertse, "Cryptanalysis of DES with a Reduced Number of Rounds: Sequences of Linear Factors in Block Ciphers," *CRYPTO 1985* (H.C. Williams, ed.), vol. 218 of *LNCS*, pp. 192–211, Springer-Verlag, 1986.
5. W. Diffie, M.E. Hellman, "Exhaustive Cryptanalysis of the NBS Data Encryption Standard," *Computer*, vol. 10(6), pp. 74–84, IEEE Computer Society Press, 1977.
6. O. Dunkelman, G. Sekar, B. Preneel, "Improved Meet-in-the-Middle Attacks on Reduced-Round DES," *INDOCRYPT 2007* (K. Srinathan, C. Pandu Rangan, M. Yung, eds.), vol. 4859 of *LNCS*, pp. 86–100, Springer-Verlag, 2007.
7. A. Grothe, "Kernel v2.6.14 tea.c," *Linux Headquarters*, 2004, available at `http://www.linuxhq.com/kernel/v2.6/14/crypto/tea.c`.
8. S. Hong, D. Hong, Y. Ko, D. Chang, W. Lee, S. Lee, "Differential Cryptanalysis of TEA and XTEA," *ICISC 2003* (J.I. Lim, D.H. Lee, eds.), vol. 2971 of *LNCS*, pp. 402–417, Springer-Verlag, 2004.
9. D. Hong, B. Koo, Y. Sasaki, "Improved Preimage Attack for 67-Step HAS-160," *ICISC 2009* (D. Lee, S. Hong, eds.), vol. 5984 of *LNCS*, pp. 332–348, Springer-Verlag, 2009.
10. J.-P. Kaps, "Chai-Tea, Cryptographic Hardware Implementations of xTEA," *INDOCRYPT 2008* (D.R. Chowdhury, V. Rijmen, A. Das, eds.), vol. 5365 of *LNCS*, pp. 363–375, Springer-Verlag, 2008.
11. J. Kelsey, B. Schneier, D. Wagner, "Key-Schedule Cryptoanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES," *CRYPTO 1996* (N. Koblitz, ed.), vol. 1109 of *LNCS*, pp. 237–251, Springer-Verlag, 1996.
12. J. Kelsey, B. Schneier, D. Wagner, "Related-key cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA," *ICICS 1997* (Y. Han,

T. Okamoto, S. Qing, eds.), vol. 1334 of *LNCS*, pp. 233–246, Springer-Verlag, 1997.
13. Y. Ko, S. Hong, W. Lee, S. Lee, J.-S. Kang, "Related-Key Differential Attacks on 27 Rounds of XTEA and Full-Round GOST," *FSE 2004* (B.K. Roy, W. Meier, eds.), vol. 3017 of *LNCS*, pp. 299–316, Springer-Verlag, 2004.
14. E. Lee, D. Hong, D. Chang, S. Hong, J. Lim, "A Weak Key Class of XTEA for a Related-Key Rectangle Attack," *VIETCRYPT 2006* (P.Q. Nguyen, ed.), vol. 4341 of *LNCS*, pp. 286–297, Springer-Verlag, 2006.
15. J. Lu, "Related-key rectangle attack on 36 rounds of the XTEA block cipher," *International Journal of Information Security*, vol. 8(1), pp. 1–11, Springer-Verlag, 2009, also available at `http://jiqiang.googlepages.com/IJIS8.pdf`.
16. D. Moon, K. Hwang, W. Lee, S. Lee, J. Lim, "Impossible Differential Cryptanalysis of Reduced Round XTEA and TEA," *FSE 2002* (J. Daemen, V. Rijmen, eds.), vol. 2365 of *LNCS*, pp. 49–60, Springer-Verlag, 2002.
17. R.M. Needham, D.J. Wheeler, "Tea extensions," technical report, Computer Laboratory, University of Cambridge, October 1997, available at `http://www.cix.co.uk/ klockstone/xtea.pdf`.
18. R.M. Needham, D.J. Wheeler, "Correction to xtea," technical report, Computer Laboratory, University of Cambridge, October 1998, available at `http://www.movable-type.co.uk/scripts/xxtea.pdf`.
19. M.-J. Saarinen, "Cryptanalysis of Block TEA," unpublished manuscript, October 1998, available at `http://groups.google.com/group/sci.crypt.research/msg/f52a533d1e2fa15e`.
20. Y. Sasaki, K. Aoki, "Finding Preimages in Full MD5 Faster Than Exhaustive Search," *EUROCRYPT 2009* (A. Joux, ed.), vol. 5479 of *LNCS*, pp. 134–152, Springer-Verlag, 2009.
21. C.E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal*, vol. 28-4, pp. 656–715, 1949.
22. M. Steil, "17 Mistakes Microsoft Made in the Xbox Security System," *Chaos Communication Congress 2005*, available at `http://events.ccc.de/congress/2005/fahrplan/events/559.en.html`.
23. D.J. Wheeler, R.M. Needham, "TEA, a Tiny Encryption Algorithm," *FSE 1994* (B. Preneel, ed.), vol. 1008 of *LNCS*, pp. 363–366, Springer-Verlag, 1994.

A Countermeasures

The attacks in this paper are made possible because a particular subkey K_i is often not used for a large number of rounds. To prevent against the attacks in this paper, we propose to use each of the subkeys K_0, K_1, K_2, K_3 once every four rounds, in a random order. This countermeasure does not prevent trivial meet-in-the-middle attacks on 6 rounds. Note that the subkeys cannot repeat in a cyclic manner, as we want to avoid the possibility of slide attacks.

B Illustration of the Attack on Rounds 16–38

In Fig. 4, we illustrate the 23-round attack of Sect. 6. The attack is on rounds 16–38, and uses 11 inner rounds (22–32). Grey boxes represent bits that do not depend on the value of $K_3[31 \ldots 21]$. In Fig. 3, we illustrate Algorithm 1 from the point of view of computation of its time complexity.

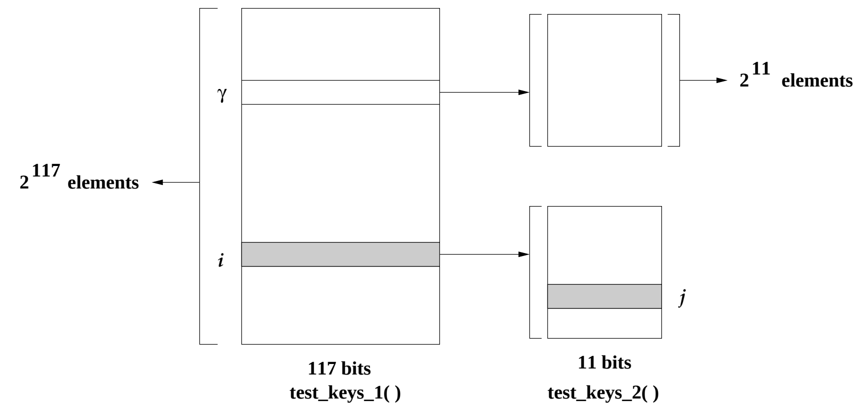

Fig. 3. Attack on rounds 16–38 using Algorithm 1: the tables (not stored in memory) denote the two stages of Algorithm 1 and the shaded 128 bits denote the correct 128-bit key; for a wrong key γ, `test_keys_2()` is performed 2^{11} times

C Randomness of the Inner-Round Subkeys in the 15-Round Attacks

Here, we show that if the texts obtained by encrypting p_0 and decrypting c_0 in the 13 outer rounds (of a 15-round attack) are uniformly distributed at random, then so are the subkeys in the inner rounds. As there are only two inner rounds, the problem may be viewed as follows. In Fig. 1, if $L_{t-1}||R_{t-1}$ and $L_{t+1}||R_{t+1}$ are uniformly distributed at random, then we need to show that α_t and α_{t+1} are also uniformly distributed at random. Henceforth, the term *random* means *uniformly distributed at random*.

Since F is a bijection, the output of F is random given R_{t-1} is random. We know that modular addition (or subtraction) or exclusive-OR of two random values results in a random value. Given this, since $R_t = L_{t+1}$ and $L_{t+1}||L_{t-1}$ is random, from Fig. 1 we obtain that $\delta_t \boxplus \alpha_t$ is random. As δ_t is a constant, α_t is random. By similar arguments, it is easily seen that α_{t+1} is also random.

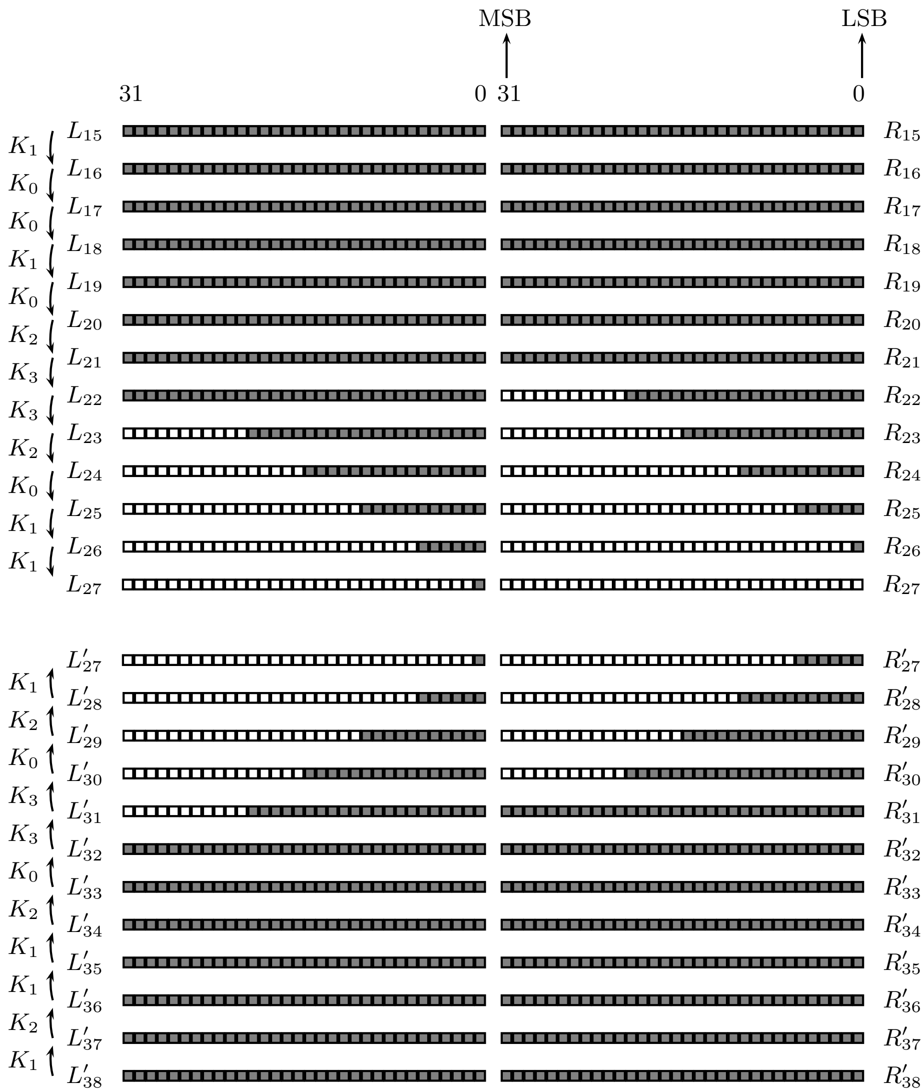

Fig. 4. 23-round attack (rounds 16–38), using 11 *inner rounds* (the grey boxes represent bits that do not depend on the value of $K_3[31\ldots21]$)