Impossible Differential Cryptanalysis of the Lightweight Block Ciphers TEA, XTEA and HIGHT ⋆

Jiazhe Chen[1,2,3], Meiqin Wang[1,2,3]⋆⋆, and Bart Preneel[2,3]

[1] Key Laboratory of Cryptologic Technology and Information Security, Ministry of Education, School of Mathematics, Shandong University, Jinan 250100, China

[2] Department of Electrical Engineering ESAT/SCD-COSIC, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium

[3] Interdisciplinary Institute for BroadBand Technology (IBBT), Belgium

mqwang@sdu.edu.cn

Abstract. TEA, XTEA and HIGHT are lightweight block ciphers with 64-bit block sizes and 128-bit keys. The round functions of the three ciphers are based on the simple operations XOR, modular addition and shift/rotation. TEA and XTEA are Feistel ciphers with 64 rounds designed by Needham and Wheeler, where XTEA is a successor of TEA, which was proposed by the same authors as an enhanced version of TEA. Whilst HIGHT, which is designed by Hong et al., is a generalized Feistel cipher with 32 rounds and eight 8-bit words in each round. On the one hand, all these ciphers are simple and easy to implement; on the other hand, the diffusion is slow, which allow us to find some impossible properties.

This paper proposes a method to identify the impossible differentials for TEA and XTEA by using the diffusion property of these block ciphers, where the impossible differential comes from one bit contradiction. By means of the method, 14-round impossible differential of XTEA and 13-round impossible differential of TEA are derived, which results in improved impossible differential attacks on 23-round XTEA and 17-round TEA, respectively. These attacks significantly improve the previous 11-round impossible differential attack on TEA and 14-round impossible differential attack on XTEA given by Moon et al. from FSE 2002. For HIGHT, we improve the 26-round impossible differential attack proposed by Özen et al.; an impossible differential attack on 27-round HIGHT that is slightly faster that the exhaustive search is also given. The attacks on TEA, XTEA and HIGHT are also the best attacks in terms of time complexity.

1 Introduction

TEA [18], XTEA [15] and HIGHT [4] are lightweight block ciphers that suit for low resource devices like RFID tags and sensor nodes. TEA was proposed by Needham and Wheeler in 1994; it is a simple design that is easy to understand and implement. By exploiting its too simple key schedule, Kelsey et al. proposed a related-key attack on full TEA [7]. In order to preclude the attack, the authors enhanced the cipher with an improved key schedule and a different round function by rearranging the operations; the new version is called XTEA. Both TEA and XTEA are implemented in the Linux kernel; they use modular addition (modulo 2^{32}), shift (left and right) and XOR in their round functions. A few cryptanalytic results on TEA and XTEA have been presented during the last several years. In the single-key setting, Moon et al. gave impossible differential attacks on 11-round TEA and 14-round XTEA [14]. Then truncated differential attacks were proposed by Hong et al. [5] that can break TEA reduced to 17 rounds with $2^{123.73}$ encryptios and XTEA reduced to 23 rounds with $2^{120.65}$ encryptions. Later, Sekar et al. presented a meet-in-the-middle of 23-round XTEA with complexity 2^{117} [17]. There are also attacks on XTEA in the related-key setting, which are given in [2][9][11][13].

⋆ This work was supported in part by the Research Council K.U.Leuven: GOA TENSE, 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 NSFC Projects (No.61133013, No.61070244 and No.60931160442), Outstanding Young Scientists Foundation Grant of Shandong Province (No.BS2009DX030), IIFSDU Project (No. 2009TS087).

HIGHT, designed by Hong et al., was standardized by the Telecommunications Technology Association (TTA) of Korea. It is an 8-branch generalized Feistel with initial and final whitening layers; its round function uses addition modulo 2^8, rotation and XOR are used. The best related-key attack of HIGHT is a full-round rectangle attack with complexity $2^{125.83}$ [10]. The best single-key attack is a 26-round impossible differential cryptanalysis proposed by [16], which does not take the initial whitening layer into account and needs $2^{119.53}$ encryptions.

The impossible differential attack, which was independently proposed by Biham et al. [1] and Knudsen [8], is a widely used cryptanalytic method. The attack starts with finding an input difference that can never result in an output difference, which makes up an impossible differential. By adding rounds before and/or after the impossible differential, one can collect pairs with certain plaintext and ciphertext differences in the data collection phase. Then the guessed subkey bits in the added rounds must be wrong, if there exits a pair that meets the input and output values of the impossible differential under these subkey bits. In this way, we discard as many wrong keys as possible and exhaustively search the rest of the keys, this phase is called key recovery phase. The early abort technique is usually used during the key recovery phase, that is, one does not guess all the subkey bits at once, but guess some subkey bits instead to discard some pairs that do not satisfy certain conditions step by step. In this case, we can discard the unwished pairs as soon as possible to reduce the time complexity.

Our Contribution. This paper presents a novel method to derive the impossible differentials for TEA and XTEA. Due to the one-directional diffusion property of TEA and XTEA, one can determine a one-bit difference after a chosen difference propagates several steps forward/backward, which might lead to a one-bit contradiction in certain rounds if we choose two differences and make them propagate towards each other. Then we propose 13-round and 14-round impossible differentials for TEA and XTEA respectively, resulting in improved impossible differential attacks on 17-round TEA and 23-round XTEA. Using 2^{57} chosen plaintexts, 17-round TEA can be attacked with time complexity $2^{106.8}$. If we use 2^{62} plaintexts, the time complexity of the attack on 23-round XTEA is $2^{116.9}$ encryptions; while if we increase the data complexity to 2^{63}, the complexity of the attack will become 2^{106} memory accesses and $2^{105.6}$ encryptions. The attacks on TEA and XTEA greatly improve corresponding the impossible differential attacks in [14], as well as reduce the best known time complexities of the attacks on these two ciphers [5,17].

Furthermore, we present impossible differential attacks on HIGHT reduced to 26 and 27 rounds which improve the result of [16]. The complexity of the 26-round attack is $2^{61.6}$ chosen plaintexts and $2^{114.35}$ encryptions, while the 27-round attack needs 2^{59} chosen plaintexts, $2^{126.6}$ 27-round encryptions and 2^{120} memory accesses. We summarize our results of TEA, XTEA and HIGHT, as well as the major previous results in Table 1.

The rest of the paper is organized as follows. We give some notations and brief descriptions of TEA, XTEA and HIGHT in Sect. 2. Some properties of TEA, XTEA and HIGHT are described in Sect. 3. Section 4 gives the impossible differential and our attacks of reduced TEA and XTEA. The impossible differential cryptanalysis of HIGHT is presented in Sect. 5. Finally, Section 6 concludes the paper.

2 Preliminary

2.1 Notations

- $\boxplus$: addition modular 2^{32} or 2^8
- $\oplus$: exclusive-OR (XOR)
- MSB: most significant bit, which is the left-most bit
- LSB: least significant bit, which is the right-most bit
- ?: an indeterminate difference

Table 1. Summary of Single-Key Attacks on TEA, XTEA and HIGHT

Attack	#Rounds	Data	Time	Ref.
TEA				
Impossible Differential	11	$2^{52.5}$ CP	2^{84}EN	[14]
Truncated Differential	17	1920 CP	$2^{123.37}$EN	[5]
Impossible Differential	17	2^{57} CP	$2^{106.8}$EN	this paper
XTEA				
Impossible Differential	14	$2^{62.5}$ CP	2^{85}EN	[14]
Truncated Differential	23	$2^{20.55}$ CP	$2^{120.65}$EN	[5]
Meet-in-the-Middle	23	18 KP	2^{117}EN	[17]
Impossible Differential	23	2^{62} CP	$2^{116.9}$EN	this paper
Impossible Differential	23	2^{63} CP	2^{101}MA+$2^{105.6}$EN	this paper
HIGHT				
Saturation	22	$2^{62.04}$ CP	$2^{118.71}$EN	[19]
Impossible Differential	25	2^{60} CP	$2^{126.78}$EN	[12]
Impossible Differential	26	2^{61} CP	$2^{119.53}$EN	[16]
Impossible Differential	26	$2^{61.6}$ CP	$2^{114.35}$EN	this paper
Impossible Differential	27	2^{59} CP	2^{120} MA+$2^{126.6}$EN	this paper

RK: Related-Key; CP: Chosen Plaintext; KP: Known Plaintext;
EN: Encryptions; MA: Memory Accesses.

- $||$: cascade of bits
- ΔA: the XOR difference of a pair (A, A'), where A and A' are values of arbitrary length
- A_i: the i-th bit of A, where the 1st bit is the LSB
- $A_{i\sim j}$: the i-th to j-th bits of A
- $(\cdot)_2$: the binary representation a byte, where the left-most bit is the MSB
- $D[i]$: a 32-bit difference where the i-th bit is 1, the first to the $(i-1)$-th bits are 0, and the $(i+1)$-th to 32-th bits are indeterminate

2.2 Brief Description of TEA and XTEA

TEA and XTEA are 64-bit block ciphers with 128-bit key-length. The key K can be described as follows: $K = (K_0, K_1, K_2, K_3)$, where K_i $(i = 0, ..., 3)$ are 32-bit words. Denote the plaintext by (P_L, P_R), the ciphertext by (C_L, C_R), and the input of the i-th round by (L_{i-1}, R_{i-1}), so $(L_0 = P_L, R_0 = P_R)$. Then we can briefly describe the encryption procedure of TEA.

For $i = 1$ to 64, if $i \mod 2 = 1$, then

$$L_i = R_{i-1} \ ,$$
$$R_i = L_{i-1} + (((R_{i-1} \ll 4) + K_0) \oplus (R_{i-1} + (i+1)/2 \times \delta) \oplus ((R_{i-1} \gg 5) + K_1)) \ .$$

If $i \mod 2 = 0$, then

$$L_i = R_{i-1} \ ,$$
$$R_i = L_{i-1} + (((R_{i-1} \ll 4) + K_2) \oplus (R_{i-1} + (i+1)/2 \times \delta) \oplus ((R_{i-1} \gg 5) + K_3)) \ .$$

Finally, $(C_L = L_{64}, C_R = R_{64})$. Where the constant $\delta = 0x9e3779b9$. XTEA is also very simple, it has similar structure and round function as TEA. To make the cipher resist against related-key attack that was mounted on TEA, XTEA has a key schedule which is a bit complicated than that of TEA's. By using the same notion as TEA, the encryption procedure of XTEA is depicted as follows.

For $i = 1$ to 64,

$$L_i = R_{i-1} \ ,$$
$$R_i = L_{i-1} + (((R_{i-1} \ll 4 \oplus R_{i-1} \gg 5) + R_{i-1}) \oplus (i/2 \times \delta + K_{((i-1)/2 \times \delta \gg 11) \cap 3}) \ .$$

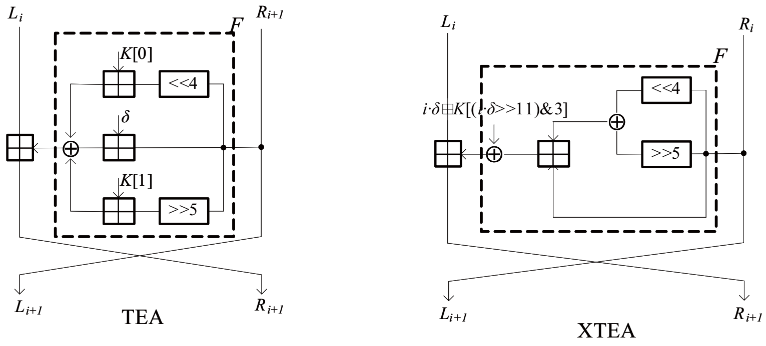

Fig. 1. Round Functions of TEA and XTEA

The round functions of TEA and XTEA are illustrated in Fig. 1.
The sequence K_i that is used in each round of XTEA can be found in Table 2.

Table 2. Subkey Used in Each Round of XTEA

K_0	K_3	K_1	K_2	K_2	K_1	K_3	K_0	K_0	K_0	K_1	K_3	K_2	K_2	K_3	K_1
K_0	K_0	K_1	K_0	K_2	K_3	K_3	K_2	K_0	K_1	K_1	K_1	K_2	K_0	K_3	K_3
K_0	K_2	K_1	K_1	K_2	K_1	K_3	K_0	K_0	K_3	K_1	K_2	K_2	K_1	K_3	K_1
K_0	K_0	K_1	K_3	K_2	K_2	K_3	K_2	K_0	K_1	K_1	K_0	K_2	K_3	K_3	K_2

2.3 Brief Description of HIGHT

Set $s_0 \leftarrow 0$, $s_1 \leftarrow 1$, $s_2 \leftarrow 0$, $s_3 \leftarrow 1$, $s_4 \leftarrow 1$, $s_5 \leftarrow 0$ and $s_6 \leftarrow 1$.
$\delta_0 = s_6||s_5||s_4||s_3||s_2||s_1||s_0$.
For $i = 1$ to 127,
    $s_{i+6} = s_{i+2} \oplus s_{i-1}$,
    $\delta_i = s_{i+6}||s_{i+5}||s_{i+4}||s_{i+3}||s_{i+2}||s_{i+1}||s_i$.

For $i = 0$ to 7,
    for $j = 0$ to 7,
        $SK_{16i+j} = MK_{(j-i) \mod 8} \boxplus \delta_{16i+j}$.
    for $j = 0$ to 7,
        $SK_{16i+j+8} = MK_{((j-i) \mod 8)+8} \boxplus \delta_{16i+j+8}$.

Fig. 2. Subkey Generation of HIGHT

HIGHT is a lightweight block cipher with a 64-bit block size and a 128-bit key. The cipher consists of 32 rounds with four parallel Feistel functions in each round; whitening keys are applied before the first round and after the last round. The master key of HIGHT is composed of 16 bytes MK=(MK_{15}, MK_{14}, MK_{13}, MK_{12}, MK_{11}, MK_{10}, MK_9, MK_8, MK_7, MK_6,MK_5, MK_4, MK_3, MK_2, MK_1, MK_0); the whitening keys (WK_0, WK_1, WK_2, WK_3, WK_4, WK_5, WK_6, WK_7) and round subkeys (SK_0, ..., SK_{127}) are generated from the master key by the key schedule algorithm. The schedule of whitening keys is relatively simple and results in $WK_0 = MK_{12}$, $WK_1 = MK_{13}$, $WK_2 = MK_{14}$, $WK_3 = MK_{15}$, $WK_4 = MK_0$, $WK_5 = MK_1$, $WK_6 = MK_2$, $WK_7 = MK_8$. The 128 7-bit constants $\delta_0, ..., \delta_{127}$ have to be generated before generating the round subkeys; the algorithm is described in Fig. 2.

Let the plaintext and ciphertext be $P = (P_7, P_6, P_5, P_4, P_3, P_2, P_1, P_0)$ and $C= (C_7, C_6, C_5, C_4, C_3, C_2, C_1, C_0)$, where P_j, C_j $(j = 0, ..., 7)$ are 8-bit values. If we denote the input of the $(i+1)$-round be $X^i = (X_7^i, X_6^i, X_5^i, X_4^i, X_3^i, X_2^i, X_1^i, X_0^i)$, then an initial transformation is first applied to P by setting $X_0^0 \leftarrow P_0 \boxplus WK_0$, $X_1^0 \leftarrow P_1$, $X_2^0 \leftarrow P_2 \oplus WK_1$, $X_3^0 \leftarrow P_3$, $X_4^0 \leftarrow P_4 \boxplus WK_2$, $X_5^0 \leftarrow P_5$, $X_6^0 \leftarrow P_6 \oplus WK_3$ and $X_7^0 \leftarrow P_7$. After this, the round transformation iterates for 32 times:

$$\boxed{\begin{array}{l} \text{For } i = 0 \text{ to } 32, \\ \quad X_1^{i+1} = X_0^i,\ X_3^{i+1} = X_2^i,\ X_5^{i+1} = X_4^i,\ X_7^{i+1} = X_6^i, \\ \quad X_0^{i+1} = X_7^i \oplus (F_0(X_6^i) \boxplus SK_{4i+3}), \\ \quad X_2^{i+1} = X_1^i \boxplus (F_1(X_0^i) \oplus SK_{4i+2}), \\ \quad X_4^{i+1} = X_3^i \oplus (F_0(X_2^i) \boxplus SK_{4i+1}), \\ \quad X_6^{i+1} = X_5^i \boxplus (F_1(X_4^i) \oplus SK_{4i}). \end{array}}$$

Here $F_0(x) = (x \lll 1) \oplus (x \lll 2) \oplus (x \lll 7)$, and $F_1(x) = (x \lll 3) \oplus (x \lll 4) \oplus (x \lll 6)$. One round of HIGHT is illustrated in Fig. 3.

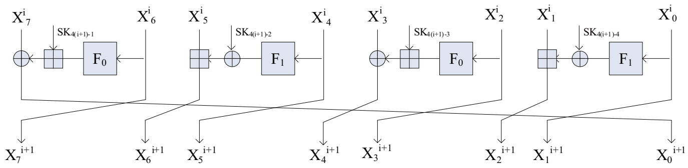

Fig. 3. One Round of HIGHT

A final transformation is used to obtain the ciphertext C, where $C_0 = X_1^{32} \boxplus WK_4$, $C_1 = X_2^{32}$, $C_2 = X_3^{32} \oplus WK_5$, $C_3 = X_4^{32}$, $C_4 = X_5^{32} \boxplus WK_6$, $C_5 = X_6^{32}$, $C_6 = X_7^{32} \oplus WK_7$ and $C_7 = X_0^{32}$.

3 Diffusion Properties of TEA, XTEA and HIGHT

For XTEA and TEA, instead of rotations, shifts (left and right) are used, hence the differences that are shifted beyond MSB/LSB will be absorbed, which results in a slower diffusion than for rotations. In other words, the difference in the most significant bits can only influence the least signification bits after several rounds. This is the starting point of our attacks, which allows us to construct impossible differentials. The derivation of the impossible differentials will be elaborated in Sect. 4.1.

There is also a common property in the block ciphers TEA, XTEA and HIGHT, that is, the round subkeys are added (or XORed) to the intermediate values after the diffusion operations. Furthermore, the operations used in all the three ciphers are modular addition, XOR and shift (rotation), which may allow us to guess the subkey bit by bit from the LSB to the MSB to abort the wrong pairs as soon as possible to reduce the time complexity.

Theorem 1 and Property 1 below are also useful for attacks on TEA and XTEA, as the adversary can guess as few key bits as possible by using them.

Theorem 1 *(From [3]). Let $[x+y]$ be $(x+y) \mod 2^n$, then $[x+y]_i = x_i \oplus y_i \oplus c_i$ $(i = 1, ..., n)$, where $c_1 = 0$ and $c_i = x_{i-1}y_{i-1} \oplus x_{i-1}c_{i-1} \oplus y_{i-1}c_{i-1}$, for $i = 2, \ldots, n$.*

Property 1. Given x, x', y, y' be n-bit values, and $z = (x+y) \mod 2^n$, $z' = (x'+y') \mod 2^n$. If the i-th (counting from 1) to j-th bits of x, x', y, y' and the i-th carry c_i, c_i' of $x+y$, $x'+y'$ are known, then the i-th to j-th $(i < j < n)$ bits of Δz can be obtained, regardless of the values of least significant $i-1$ bits of x (or x'), y (or y'). Note that if there are no differences in the the least significant $i-1$ bits of $x+y$ and $x'+y'$, then $c_i = c_i'$

Proof. Let us start from Δz_i. As we know from Theorem 1, $z_i = x_i \oplus y_i \oplus c_i$, $z'_i = x'_i \oplus y'_i \oplus c'_i$, it is clear that $\Delta z_i = \Delta x_i \oplus \Delta y_i \oplus \Delta c_i$. Furthermore, $c_{i+1} = x_i y_i \oplus x_i c_i \oplus y_i c_i$, $c'_{i+1} = x'_i y'_i \oplus x'_i c'_i \oplus y'_i c'_i$. As a result, $\Delta z_{i+1} = \Delta x_{i+1} \oplus \Delta y_{i+1} \oplus \Delta c_{i+1}$ can be calculated. As the values of the most significant bits have nothing to do with the differences of the least significant bits after the modular addition, we can similarly get bits $i+2$ to j of Δz by iteratively computing c_{i+2}, c'_{i+2} to c_j, c'_j. □

4 Impossible Differential Attacks on Reduced XTEA and TEA

In this section, we first explain how to obtain the impossible differentials for TEA and XTEA. Then 13-round impossible differential for TEA and 14-round impossible differential for XTEA are given, which are used to attack 17-round TEA and 23-round XTEA .

4.1 Impossible Differentials of TEA and XTEA

As mentioned in Sect. 3, we know that the differences in the most significant bits propagate only in one direction. Since both TEA and XTEA use operations that shift to the left for 4 bits and shift to the right for 5 bits, they share the following properties.

Property 2. If the input difference of the i-th round of XTEA (TEA) is $(0, D[n])$, then the output difference is $(D[n], D[n-5])$. Vice versa, if the output difference of the j-th round of XTEA (TEA) is $(D[p], 0)$, then the input difference is $(D[p-5], D[p])$.

Property 3. If the input difference of the i-th round of XTEA (TEA) is $(D[m], D[n])$, where $(m > n-5)$, then the output difference is $(D[n], D[n-5])$. Vice versa, if the output difference of the j-th round of XTEA (TEA) is $(D[p], D[q])$, where $(q > p-5)$, then the input difference is $(D[p-5], D[p])$.

From Property 2 and Property 3, we propose a method to construct an impossible differential for TEA and XTEA. If we choose the input difference to be $(0, D[n])$ (or $(D[m], D[n])$ $(m > n-5)$), then after i rounds, the difference should be of the form $(D[n-5(i-1)], D[n-5i])$. Similarly, if we choose the output difference $(D[p], 0)$ (or $(D[p], D[q])$ $(q > p-5)$), then after propagating backwards for j rounds, the difference should be of the form $(D[p-5j], D[p-5(j-1)])$. For an $x < 0$, $D[x]$ means that all the 32 bits of the difference are indeterminate. Then a one-bit contradiction will appear if

$$n - 5(i-1) > 0, p - 5j > 0, n - 5(i-1) \neq p - 5j,$$

or

$$n - 5i > 0, p - 5(j-1) > 0, n - 5i \neq p - 5(j-1).$$

With this method, we can derived a 14-round impossible differential for XTEA and a 13-round impossible differential for TEA (see Fig. 4), where the left-most bit is the MSB, each small rectangle stands for one bit: blank rectangles mean that there are no differences in these bits, while black ones mean the differences are equal to 1, and gray ones mean that the differences are indeterminate. Actually, we can derive 15-round impossible differentials for both XTEA and TEA. However, in this case, we should use almost the whole data space to mount impossible differential attacks (maybe for more rounds), hence we use the impossible differentials above to achieve attacks with lower data complexities.

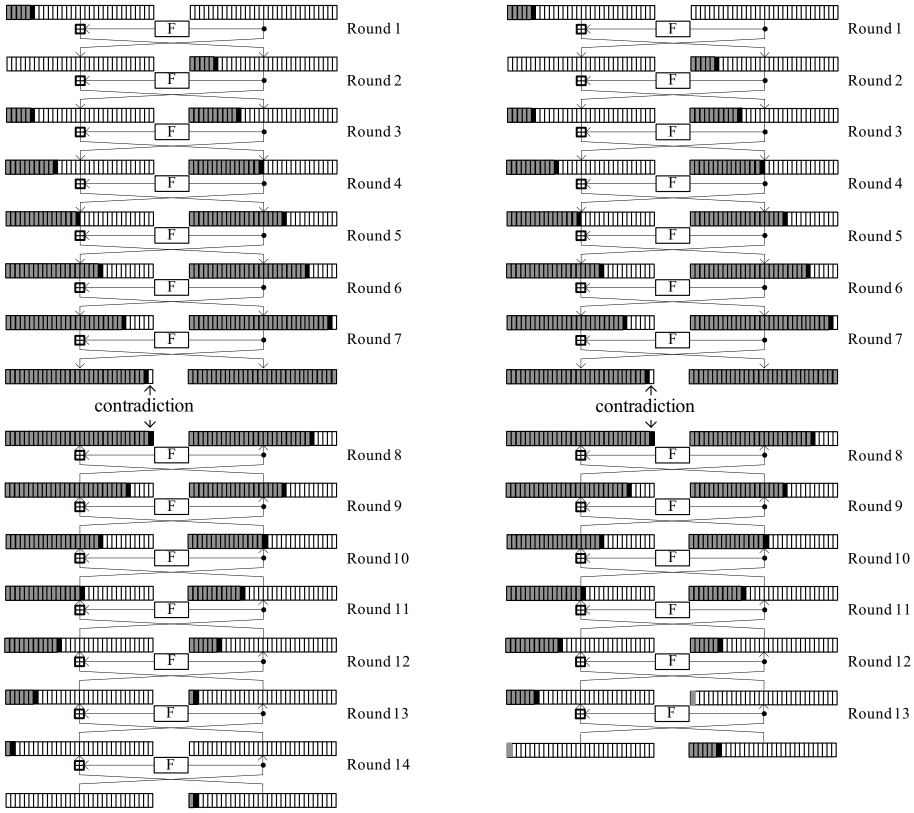

Fig. 4. Impossible Differential of XTEA (left) and TEA (right)

4.2 Impossible Differential Attack of 23-Round XTEA

By placing the 14-round impossible differential on rounds 11-24, we can attack XTEA from round 6 to round 28. This is clarified in Fig. 5.

Data Collection. We first construct 2^5 structures of plaintexts, where in each structure the LSB of P_L and the 6 least signification bits of P_R are fixed, whereas the other bits take all values. For each structure, ask for the encryption of the plaintexts to get the corresponding ciphertexts. By the birthday paradox, we can get $2^{57\times2-1} \times 2^{-29} = 2^{84}$ pairs that satisfy $(\Delta P_L)_1 = 1$, $(\Delta P_R)_6 = 1$, $(\Delta C_L)_{15} = 1$, $(\Delta C_R)_{10} = 1$, the 15 least signification bits of ΔC_L are 0, and the 10 least signification bits of ΔC_R are 0 in each structure. As a result, 2^{89} pairs are obtained since we have 2^5 structures; the number of chosen plaintexts is 2^{62}.

Key Recovery. In order to find if there are pairs obtained from the data collection phase that may follow the differential in Fig. 5, we need to guess the key bits and sieve the pairs in rounds 6-10 and 25-28. From Table 2 we know the subkey used in each round (which are K_1, K_3, K_0, K_0, K_0; and K_0, K_1, K_1, K_1), hence we know the key bits we have to guess in each step.

As mentioned above, for XTEA the round subkeys intervene in the round functions after the diffusion, hence from Property 1 one can deduce that the attacker does not have to guess all the 32 bits of the subkey to sieve the pairs according to the required differences.

The key recovery process is described in Table 3, where the second column stands for the bits that have to be guessed in each step. Note that in Step 6, guessing bits $1 \sim 6$ of K_3 only takes 2^5 times, since one-bit information is known from c_2. Similarly, the it takes 2^{12} guesses for bits $1 \sim 11$ and $23 \sim 25$ of K_0. The fifth and fourth columns of Table 3 are the rounds where the sieving is launched and the conditions that can be used to sieve; the last column shows the number of remaining pairs after each step (for each key guess). Consequently, we can get the time complexity (measured by the number of 23-round encryptions) of each step, which is given in column 3 of the table.

In Step 7, if there is a pair kept, then we discard the key guess and try another one. Otherwise, for this key guess we exhaustively search the remaining 2^{32} keys by trial encryptions, and then either output the correct key or try another 96-bit key guess.

Analysis of the Attack. From the data collection phase we know that the data complexity, i.e., the number of plaintexts we need is equal to 2^{62}. In Step 7 of the key recovery phase, about $2^{96} \times (1 - 2^{-20})^{2^{23}} \approx 2^{84.8}$ 96-bit values (K_0, K_1, K_3) will remain. Since the trial encryptions need two plaintext-ciphertext pairs, the cost of the trial encryptions is about $2^{32} \times 2^{84.8} + 2^{52.8} = 2^{116.8}$ 27-round XTEA encryptions. The complexity of this step is about $2 \times 2^{96} \times (1 + (1-2^{-20}) + ... + (1-2^{-20})^{2^{23}-1}) \times 2/23 + 2^{116.8} \approx 2^{113.5} + 2^{116.8} \approx 2^{116.9}$ encryptions, which is also the dominating time complexity of the attack. The memory complexity is 2^{95} bytes which are used to store the pairs.

Table 3. Attack on 23-Round XTEA

Step	Guess Bits	Complexity	Sieve on	Conds	Pairs Left
1	$K_1 : 1 \sim 12$	$2^{97.5}$	round 6	10	2^{79}
2	$K_1 : 13 \sim 22$	$2^{97.5}$	round 28	10	2^{69}
3	$K_1 : 23 \sim 32$	$2^{98.5}$	round 26,27	20	2^{49}
4	$K_0 : 26 \sim 32, c_1$*	$2^{85.5}$	round 25	6	2^{43}
5	$K_3 : 7 \sim 17, c_2$†	$2^{90.5}$	round 7	10	2^{33}
6	$K_3 : 1 \sim 6, 18 \sim 32, K_0 : 12 \sim 22, c_3$‡	$2^{113.5}$	round 8	10	2^{23}
7	$K_0 : 1 \sim 11, 23 \sim 25$	$2^{116.9}$	round 9,10	20	–

*c_1 is the 26th carry in the left modular addition of the 25th round
† c_2 is the 7th carry in the left modular addition of the 7th round
‡ c_3 is the 12th carry in the left modular addition of the 8th round

Reducing the Time Complexity. If we prepare the pairs that satisfy the conditions of rounds 8, 9 and 10 by precomputation, we can avoid guessing bits $1 \sim 25$ of K_0 by doing some table look-ups and memory accesses. If the same data complexity is used, the time complexity will be dominated by the trials encryptions that used to discard the remaining keys. Hence we also increase the data complexity to 2^{63} by choosing 2^6 structures. First we illustrate the procedure of precomputation: we choose $\Delta L_{10} = D[27]$ and $\Delta R_{10} = 0$, for each K_0, L_{10} and R_{10}, decrypt all $(L_{10}, L_{10} \oplus \Delta L_{10})$ and $(R_{10}, R_{10} \oplus \Delta R_{10})$ to get $(L_7, L_7 \oplus \Delta L_7)$ and $(R_7, R_7 \oplus \Delta R_7)$ (the subkey used in round 8, 9 and 10 is K_0); then insert bits $1 \sim 25$ of K_0 into a hash table T indexed by $(L_7, R_7, \Delta L_7, \Delta R_7, (K_0)_{26 \sim 32})$. Since $\Delta L_7 = D[12]$ and $\Delta R_7 = D[17]$, there are $2^{64} \times 2^{35} \times 2^7 = 2^{106}$ $(L_7, R_7, \Delta L_7, \Delta R_7, (K_0)_{26 \sim 32})$s. However, only $2^{64} \times 2^5 \times 2^{32} = 2^{101}$ $(L_{10}, \Delta L_{10}, R_{10}, \Delta R_{10}, K_0)$s can be chosen, which results in 2^{-5} $(K_0)_{26 \sim 32}$ in each row of table T. The complexity of precomputation is $2 \times 2^{101} = 2^{102}$ 3-round encryptions.

With table T, we can replace Step 6 and Step 7 of the key-recovery procedure as follows: we construct another table Γ that contains all values of bits $1 \sim 25$ of K_0. In Step 6, after guessing bits $1 \sim 6, 18 \sim 32$ of K_3, we calculate $(L_7, R_7, \Delta L_7, \Delta R_7)$ and access the value from the corresponding row of table T. If there is a value in the row, we delete this $(K_0)_{1 \sim 25}$ from table Γ. For each guess of K_1, K_3 and bits $26 \sim 32$ of K_0, we get 2^{34} pairs before accessing table T; a fraction 2^{-5} of the 2^{34} pairs will access table T to get a $(K_0)_{1 \sim 25}$, which will be then deleted from Γ. Consequently, $2^{64} \times 2^7 \times 2^{25} \times (1 - 2^{-25})^{2^{29}} \approx 2^{73.6}$ (K_0, K_1, K_3) will remain, which have to be further tested by trial encryptions with each K_2. The complexity of this procedure is 2^{107} one-round encryptions, 2^{101} memory accesses to table T, 2^{101} memory accesses to table Γ and $2^{105.6}$ trial encryptions. If we assume that one memory access to table Γ is equivalent to one one-round encryption, then the dominated complexity is 2^{101} memory accesses to table T and $2^{105.6}$ trial encryptions, which is also the dominated complexity of the whole attack. The memory complexity of the attack is about 2^{104} bytes contributed by table T.

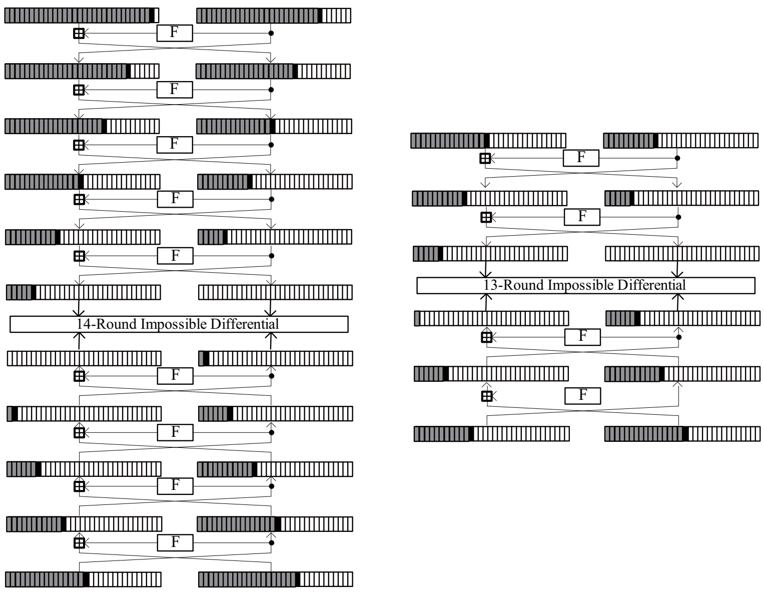

Fig. 5. 23-Round Attack on XTEA (left) and 17-Round Attack on TEA (right)

4.3 Impossible Differential Attack of 17-Round TEA

Using the 13-round impossible differential, we can attack the first 17 rounds of TEA by extending the impossible differential forward and backward for one round and two rounds, respectively (see Fig. 5).

Note from [6] one can deduce that the effective key size of TEA is only 126 bits: if the MSBs of K_0 and K_1 flip simultaneously, the output value of the round will be the same; actually, the same phenomenon happens for K_2 and K_3. As a result, every key value has three equivalent keys, which allows us to guess only one of the 4 equivalent keys when we mount an impossible differential attack on TEA. At the end of the attack, if we output one correct key, there are three other keys that are also correct.

In the data collection phase, we construct 2^{30} structures of plaintexts with the least 16 bits of P_L and the least 21 bits of P_R fixed, while the other bits take all values. Ask for the encryptions to get the ciphertexts, for each structure we can get $2^{53-39} = 2^{14}$ pairs that satisfy the required differences of the plaintext and ciphertext by the birthday paradox. Then the total number of pairs kept after the data collecting phase is 2^{44}.

Observe that K_0 and K_1 are used in the first and the 17th round, and K_2 and K_3 are used in the second and the 16th round. Hence for the remaining pairs, we first guess K_0 and K_1, partially encrypt the first round and discard the pairs do not meet the condition of ΔR_1; then decrypt the 17th round and discard the pairs whose ΔL_{16} do not satisfy the required form. The number of pairs that meet the conditions should be 2^{24}; and the complexity of this step is about $2 \times 2^{107} + 2 \times 2^{97} = 2^{108}$ one-round encryptions, equivalent to 2^{104} 17-round encryptions.

Then we guess bits $21 \sim 32$ of K_2 and K_3, the 22th carry of the left modular addition in round 2, and the 26th carry of the left modular addition in round 16. For the remaining pairs, we partially encrypt round 2 and round 16, and keep only the pairs that satisfy the required differences. If there is a pair kept, then we discard the key guess and try another one. Otherwise, for this key guess we exhaustively search the remaining key values by trial encryption, and then either output the correct key or try another guess. Considering the equivalent keys, the key values we guessed are 88 bits (including the guessed carries); the expected number of remaining 88-bit key guesses is about $2^{88} \times (1 - 2^{-20})^{2^{24}} \approx 2^{65.6}$. Since each of the remaining key guesses has to be exhaustively searched with the other 2^{38} key values, so the time complexity of this step is about $2 \times 2^{88} \times (1 + (1 - 2^{-20}) + (1 - 2^{-20})^2 + ... + (1 - 2^{-20})^{2^{24}}) \times 2/17 + 2^{65.6+38} \approx 2^{106.8}$ encryptions; this is also the dominated time complexity of the attack. The data complexity is 2^{57} and the memory complexity is 2^{50} bytes.

5 Impossible Differential Cryptanalysis of Reduced HIGHT

In this section, we improve the 26-round impossible differential attack on HIGHT in [16] by using a 16-round impossible differential that is similar as that of [16] (see Fig. 6). In order to take advantage of the redundancy in the key schedule, we carefully choose the beginning and ending rounds of the impossible differential, which are round 10 and round 25, respectively. The attack excludes the pre-whitening layer (just as what they did in [16]), and works for round 5 to round 30 (see Fig. 7). In addition, a 27-round impossible differential attack, which is slightly better than exhaustive search, is also proposed based on the 16-round impossible differential in [16] (see Fig. 8).

5.1 Improved Impossible Differential Attack on 26-Round HIGHT

Fig. 6. A 16-Round Impossible Differential Similar as that of [16]

i	ΔX_7^i	ΔX_6^i	ΔX_5^i	ΔX_4^i	ΔX_3^i	ΔX_2^i	ΔX_1^i	ΔX_0^i
9	0	0	0	0	$(???????1)_2$	0	0	0
10	0	0	0	$(???????1)_2$	0	0	0	0
11	0	?	$(???????1)_2$	0	0	0	0	0
12	?	$(???????1)_2$	0	0	0	0	0	?
13	$(???????1)_2$	0	0	0	0	?	?	?
14	0	0	0	?	?	?	?	$(???????1)_2$
15	0	?	?	?	?	?	$(???????1)_2$	0
16	?	?	?	?	?	$(???????1)_2$	0	?
17	?	?	?	?	$(???????1)_2$	?	?	?
17	?	?	?	?	$(???????0)_2$	$0x80$	?	?
18	?	?	?	$(???????1)_2$	$0x80$	0	?	?
19	?	?	$(???????1)_2$	$0x80$	0	0	?	?
20	?	$(???????1)_2$	$0x80$	0	0	0	?	?
21	$(???????1)_2$	$0x80$	0	0	0	0	?	?
22	$0x80$	0	0	0	0	0	?	$(?????100)_2$
23	0	0	0	0	0	0	$(?????100)_2$	$0x80$
24	0	0	0	0	0	0	$0x80$	0
25	0	0	0	0	0	$0x80$	0	0

Fig. 7. Impossible Differential Attack on 26-Round HIGHT

i	ΔX_7^i	ΔX_6^i	ΔX_5^i	ΔX_4^i	ΔX_3^i	ΔX_2^i	ΔX_1^i	ΔX_0^i
P	?	$(???????1)_2$	0	0	?	?	?	?
4	?	$(???????1)_2$	0	0	?	?	?	?
5	$(???????1)_2$	0	0	0	?	?	?	?
6	0	0	0	0	?	?	?	$(???????1)_2$
7	0	0	0	0	?	?	$(???????1)_2$	0
8	0	0	0	0	?	$(???????1)_2$	0	0
9	0	0	0	0	$(???????1)_2$	0	0	0
Impossible Differential								
25	0	0	0	0	0	$0x80$	0	0
26	0	0	0	$(???????1)_2$	$0x80$	0	0	0
27	0	?	$(???????1)_2$	$0x80$	0	0	0	0
28	?	$(???????1)_2$	$0x80$	0	0	0	0	?
29	$(???????1)_2$	0	0	0	0	?	?	?
30	$0x80$	0	0	?	?	?	?	$(???????0)_2$
C	$(???????0)_2$	$0x80$	0	0	?	?	?	?

In order to reduce the time complexity of the 26-round attack in [16], we choose a similar impossible differential and a different beginning round; the data complexity is slightly higher because we want to reduce the complexity of the final trial encryptions that might dominates the complexity if the chosen data is not enough. Precomputation is also used to reduce the time complexity.

Data Collection. Construct $2^{13.6}$ structures with P_4, P_5 fixed and for which $P_0, ..., P_3, P_6, P_7$ take all values. Ask for the encryptions of all the plaintexts to get the corresponding ciphertexts. Since the ciphertext pairs with the difference $((???????0)_2, 0x80, 0, 0, ?, ?, ?, ?)$ are required, and there is one more condition in the plaintext difference, which is $\Delta P_{6,0} = 1$; by the birthday paradox, there are $2^{82.6}$ pairs left.

Precomputation. Three pre-computed tables α, β and ϵ will be set up for the sake of reducing the complexity in the key recovery phase. For setting up α, we choose all values of X_4^8, X_3^8, ΔX_3^8, X_1^7, ΔX_1^7, MK_{12} and MK_{15}, calculate $(X_2^6, \Delta X_2^6)$, $(X_1^6, \Delta X_1^6)$, and $(X_0^6, \Delta X_0^6)$ by 1/2 round decryptions and insert MK_{15} to the row of α indexed by $(X_2^6, \Delta X_2^6, X_1^6, \Delta X_1^6, X_0^6, \Delta X_0^6, MK_{12})$. Hence there are one MK_{15} in each row on average; the size of α is 2^{55} bytes as there are only 2^7 ΔX_0^6s. When constructing table β, all values of X_2^{25}, X_3^{25}, X_2^{26}, MK_7 and MK_{11} are chosen, then we compute X_3^{27}, X_4^{27} and $(X_5^{25}, \Delta X_5^{25})$, and insert MK_{11} to the row indexed by $(X_3^{27}, X_4^{27}, X_5^{25}, \Delta X_5^{25}, MK_7)$. Since only 2^7 ΔX_5^{25} are possible, we have 2^{39} rows in β with 2 MK_{11}s in each row on average. The setting of table ϵ is also similar, we choose all values of X_4^9, X_3^9, ΔX_3^9, X_1^8, MK_7 and MK_{11}, and calculate X_0^7, $(X_1^7, \Delta X_1^7)$, $(X_2^7, \Delta X_2^7)$; then insert X_0^7 to the row indexed by $(X_1^7, \Delta X_1^7, X_2^7, \Delta X_2^7, MK_7, MK_{11})$. There is one X_0^7 in each row on average. The sizes of β and ϵ are 2^{40} bytes and 2^{48} bytes, respectively. Constructing table α dominates the time complexity of the precomputation, which is about 2^{56} 1/2-round encryptions.

Key Recovery. The key recovery phase is described in Table 4. Where the second column contains the key bytes/bits which are guessed in the step, the third column indicates the whitening keys/subkey keys used in the step to calculate the values that are needed, the fourth column stands for the time complexity of each step, the fifth column gives the number of bit conditions which can be used, the sixth column indicates the number of the pairs that are kept after each step and the last column gives the position of Feistel branches where the sieving occurs ((x, y) means the y-th branch of the x-th round, where the right most branch is the 0th one). To better

Table 4. Key Recovery Procedure of the Attack on 26-Round HIGHT

Step	Guess Bits	Known Keys	Complexity	Conds	Pairs Kept	Sieve on
1	MK_0	SK_{17}	$2^{84.9}$ EN	8	$2^{74.6}$	(5,1)
2	MK_1, MK_6	WK_5, SK_{117}	$2^{92.9}$ EN	8	$2^{66.6}$	(30,1)
3	MK_5	WK_4, SK_{116}, SK_{112}	$2^{93.9}$ EN	8	$2^{58.6}$	(29,0)
4	MK_4, MK_7	SK_{16}, SK_{21}	$2^{101.9}$ EN	8	$2^{50.6}$	(6, 1)
5	MK_3, MK_9	WK_7, SK_{119}, SK_{115}, SK_{111}	$2^{110.8}$ EN	8	$2^{42.6}$	(28,3)
6	–	SK_{118}, SK_{114}	$2^{102.9}$ EN	–	$2^{42.6}$	– (calculate X_4^{27})
7	MK_2	SK_{19}, SK_{20}	$2^{110.9}$ EN	–	$2^{42.6}$	– (calculate $X_2^6, \Delta X_2^6$)
8	MK_8†	SK_{25}	$2^{111.9}$ EN	8	$2^{34.6}$	(7,1)
9	MK_{12}†	SK_{110}, SK_{106}	$2^{112.9}$ EN	8	$2^{26.6}$	(27,2)
10	–	SK_{18}, SK_{19}, SK_{22}, SK_{23}	$2^{109.9}$ EN	–	$2^{26.6}$	– (calculate X_6^6, X_7^6, $(X_0^6, \Delta X_0^6), (X_1^6, \Delta X_1^6)$)
11	(accessing the pre-computed tables)		$2^{116.6}$MA+2^{111}EN			

MA: memory accesses; EN: 26-round HIGHT encryptions
† The key byte is guessed bit by bit from the LSB to the MSB.

illustrate the procedure, we also give the subkeys used, as well as the corresponding master key bytes, in Table 5 in the Appendix; the subkeys that have to be guessed in the attack are in bold.

In Step 1, for each remaining pair from the data collection phase, we guess MK_7 and discard the pairs that do not satisfy $\Delta X_4^5 = 0$ by 1/4-round encryptions. So the number of pairs kept after this step is $2^{82.6-8} = 2^{74.6}$ and the complexity of this step is about $2 \times 2^{82.6} \times 2^8 \times 1/4 \times 1/26 \approx 2^{84.9}$ 26-round encryptions. Steps 2 $\sim$ 7 are similar, we guess the subkey bytes, calculate the intermediate value and discard the pairs that do not meet the conditions. In Step 8, we do not guess all 8 bits of MK_8 at once, but guess them bit by bit from the LSB to the MSB by using the diffusion property mentioned in Sect. 3. Once we guess one bit of MK_8, we can compute the corresponding bit of ΔX_4^7 and discard the pairs that do not meet the condition. Since 8 bits of MK_8 should be guessed in 8 times, the complexity of this step is $2 \times 8 \times 2^{72} \times 2^{42.6} \times 1/4 \times 1/26 \approx 2^{111.9}$. Step 9 is similar to Step 8, except that we have to carry out 1/2-round decryption for each pair other than 1/4-round in Step 8.

In Step 11, for each pair obtained from Step 10 we first access table α to get a value of MK_{15}, then we calculate X_3^{27} to access table β. Two MK_{11}s can be obtained on average, for each of the values, we access table ϵ to get X_0^7 and calculate MK_{10} as X_6^6, X_7^6 are already known. The corresponding $(MK_{15}, MK_{11}, MK_{10})$ should be discarded. After processing all the pairs, if there are (MK_{15}, MK_{11}, MK_{10})s kept, we output them with the guessed (MK_0, MK_1, MK_2, MK_3, MK_4, MK_5, MK_6, MK_7, MK_8, MK_9, MK_{12}), and exhaustively search them with the remaining 16-bit key. Otherwise, we try another guess for (MK_0, MK_1, MK_2, MK_3, MK_4, MK_5, MK_6, MK_7, MK_8, MK_9, MK_{12}). In this step, $2^{88} \times 2^{26.6} = 2^{114.6}$ 1/4-round decryptions (equivalent to $2^{107.9}$ encryptions) should be performed to compute X_3^{27}; $2 \times 2^{88} \times 2^{26.6} = 2^{115.6}$ 1/4-round decryptions (equivalent to $2^{108.9}$ encryptions) should be performed to calculate MK_{10}. We also need $2^{88} \times 2^{26.6} = 2^{114.6}$ memory accesses to table α, $2^{115.6}$ memory accesses to table β and $2^{115.6}$ memory accesses to table ϵ. After analyzing all the pairs, we expect $2^{112} \times (1 - 2/2^{24})^{2^{26.6}} \approx 2^{95}$ 112-bit key (MK_0, MK_1, MK_2, MK_3, MK_4, MK_5, MK_6, MK_7, MK_8, MK_9, MK_{10}, MK_{11} MK_{12}, MK_{15}) will remain. So the complexity of the exhaustively search is about $2^{111} + 2^{47} \approx 2^{111}$ since each trial encryption needs two plaintext-ciphertext pairs.

If we count one memory access to tables α, β and ϵ as one-round encryption, then the complexity of Step 13 will be about $2^{116.6} \times 1/26 + 2^{111} \approx 2^{112.5}$. From Table 4, we can deduce the time complexity, which is about $2^{110.8} + 2^{110.9} + 2^{111.9} + 2^{112.9} + 2^{112.5} \approx 2^{114.35}$ encryptions. The data complexity of the attack is $2^{61.6}$ and the memory complexity is $2^{86.6}$ bytes.

5.2 Impossible Differential Attack on 27-Round HIGHT

Placing the impossible differential of [16] on round 10 to round 25, an attack on 27-round HIGHT can be mounted by discarding some of the wrong subkeys in rounds $4 \sim 9$ and $26 \sim 30$, see Fig. 9. Note that for the 27-round attack, we take both the pre-whitening and post-whitening layers into account.

Data Collection. Construct 2^3 structures with P_0 fixed and for which $P_1, ..., P_6$ take all values. Ask for the encryptions of all the plaintexts to get the corresponding ciphertexts. Since the ciphertext pairs with the difference $(?, ?, ?, ?, (???????0)_2, 0x80, 0, 0)$ are required, and there is one more condition in the plaintext difference, which is $\Delta P_{1,0} = 1$; by the birthday paradox, there are 2^{87} pairs left. Since the whitening keys are considered in our attack, we have:

$$X_0^3 = P_0 \boxplus WK_0, X_1^3 = P_1, X_2^3 = P_2 \oplus WK_1, X_3^3 = P_3,$$
$$X_4^3 = P_4 \boxplus WK_2, X_5^3 = P_5, X_6^3 = P_6 \oplus WK_3, X_7^3 = P_7.$$
$$X_0^{30} = C_7, X_1^{30} = C_0 \boxplus WK_4, X_2^{30} = C_1, X_3^{30} = C_2 \oplus WK_5,$$
$$X_4^{30} = C_3, X_5^{30} = C_4 \boxplus WK_6, X_6^{30} = C_5, X_7^{30} = C_6 \boxplus WK_7.$$

Precomputation. Before the key recovery procedure, a precomputation is carried out for the sake of reducing the time complexity. We first choose all values of MK_1, MK_8, MK_9, MK_{13}, MK_{14}, X_0^9, X_7^9, ΔX_7^9, X_0^8, X_5^8, X_3^7, X_6^{25}, X_7^{25}, X_6^{26} and X_6^{27}, calculate $(X_6^6, X_6'^6)$, $(X_5^6, X_5'^6)$, $(X_4^6, X_4'^6)$, X_3^6 and X_2^6 by 3-round decryption; and X_1^{29}, $(X_3^{29}, \Delta X_3^{29})$, X_7^{28} and X_3^{30} by 5-round encryption (see Fig. 10 in the Appendix). Then insert (MK_8, MK_9) to a hash table H indexed by (MK_1, MK_{13}, MK_{14}, $(X_6^6, X_6'^6)$, $(X_5^6, X_5'^6)$, $(X_4^6, \Delta X_4^6)$, X_3^6, X_2^6, X_1^{29}, $(X_3^{29}, \Delta X_3^{29})$, X_7^{28}, X_3^{30}). There are 2^7 ΔX_7^9s, 2^7 ΔX_4^6s and 2^7 ΔX_3^{29}s, hence on average only a fraction 2^{-7} of the rows consist one value (MK_8, MK_9). The complexity of the precomputation is about 2^{89} three-round encryptions.

Key Recovery. The key recovery procedure is demonstrated in Table 7 in the Appendix; Table 7 has the same meaning as Table 4. Table 6 is also given in the Appendix to demonstrate the subkeys that have to be guessed.

Step 1 and Step 2 are trivial: we guess the key bytes and test whether a 0 difference can be obtained. In Step 3 we guess MK_1 and MK_6 to calculate $(X_3^{29}, X_3'^{29})$; in Step 4, MK_{14} is guessed to calculate $(X_6^4, X_6'^4)$ without discarding any pairs. In order to reduce the time complexity of Step 5, we guess MK_2 bit by bit, instead of guessing the whole byte at once. We guess the bits from the LSB to the MSB, so once we guess one bit of MK_2, we can compute the corresponding bit of ΔX_0^6 and discard the pairs that do not meet the condition. In Step 6, we do not guess any key byte, but calculate ΔX_4^{28} which can be used to sieve the pairs in Step 7. The other steps are similar except Step 13, to which have to be paid more attention. In Step 13, we first construct a small table γ which consists all values of (MK_8, MK_9); then guess MK_4 to look up table H. If the corresponding row is not empty, then access the value (MK_8, MK_9) and delete the value from γ. After analyzing all the pairs, if there are (MK_8, MK_9)s kept, we output them with the guessed (MK_0, MK_1, MK_2, MK_3, MK_4, MK_5, MK_6, MK_7, MK_{10}, MK_{12}, MK_{13}, MK_{14}, MK_{15}), and exhaustively search them with the remaining 8-bit key. Otherwise, we try another guess for (MK_0, MK_1, MK_2, MK_3, MK_4, MK_5, MK_6, MK_7, MK_{10}, MK_{12}, MK_{13}, MK_{14}, MK_{15}).

We can see from Table 7 that all the values required to access table H can be calculated in Step 13 after guessing MK_4, since the only unknown values are X_4^6, ΔX_4^6 and X_7^{28}. The complexity to compute the values is less than 2^{128} one round encryptions, equivalent to $2^{123.25}$ 27-round encryptions. Since for each pair, table H will be accessed with probability 2^{-7}, it will be accessed for 2^{16} times under each key guess; hence the number of memory accesses is about $2^{104} \times 2^{16} = 2^{120}$. As each memory access discards one (MK_8, MK_9) on average, about

$2^{120} \times (1-2^{-16})^{2^{16}} = 2^{118.6}$ 112-bit keys will remain after processing all the pairs. For these remaining keys, we also need to guess the remaining 8 bits of the main key and test the $2^{118.6} \times 2^8 = 2^{126.6}$ keys by trial encryptions. Since the trial encryption needs 2 plaintext-ciphertext pairs, the complexity of the trial encryptions is about $2^{126.6} + 2^{62.6} \approx 2^{126.6}$ encryptions. Step 13 dominates the time complexity of the attack, which is $2^{126.6}$ encryptions and 2^{120} memory accesses. The data complexity is 2^{59} and the memory complexity is 2^{120} bytes for storing table H.

6 Conclusion

This paper introduces impossible differential attacks on the lightweight block ciphers TEA, XTEA and HIGHT which are based on simple operations like modular addition, XOR, shift and rotation. These attacks benefit from the slow diffusion and the simple key schedules of these ciphers.

References

1. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In: Stern, J. (ed.) Advances in Cryptology - EUROCRYPT '99. LNCS, vol. 1592, pp. 12–23. Springer (1999)
2. Bouillaguet, C., Dunkelman, O., Leurent, G., Fouque, P.A.: Another Look at Complementation Properties. In: Hong, S., Iwata, T. (eds.) FSE 2010. LNCS, vol. 6147, pp. 347–364. Springer (2010)
3. Daum, M.: Cryptanalysis of Hash Functions of the MD4-Family. PhD thesis `http://www.cits.rub.de/imperia/md/content/magnus/idissmd4.pdf`
4. Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., Chee, S.: HIGHT: A New Block Cipher Suitable for Low-Resource Device. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer (2006)
5. Hong, S., Hong, D., Ko, Y., Chang, D., Lee, W., Lee, S.: Differential Cryptanalysis of TEA and XTEA. In: Lim, J.I., Lee, D.H. (eds.) ICISC 2003. LNCS, vol. 2971, pp. 402–417. Springer (2003)
6. Kelsey, J., Schneier, B., Wagner, D.: Key-Schedule Cryptoanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES. In: Koblitz, N. (ed.) Advances in Cryptology - CRYPTO '96. Lecture Notes in Computer Science, vol. 1109, pp. 237–251. Springer (1996)
7. Kelsey, J., Schneier, B., Wagner, D.: Related-key Cryptanalysis of 3-WAY, Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In: Han, Y., Okamoto, T., Qing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 233–246. Springer (1997)
8. Knudsen, L.: DEAL - A 128-bit Block Cipher. In: NIST AES Proposal (1998)
9. Ko, Y., Hong, S., Lee, W., Lee, S., Kang, J.S.: Related Key Differential Attacks on 27 Rounds of XTEA and Full-Round GOST. In: Roy, B.K., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 299–316. Springer (2004)
10. Koo, B., Hong, D., Kwon, D.: Related-Key Attack on the Full HIGHT. In: Rhee, K.H., Nyang, D. (eds.) ICISC 2010. LNCS, vol. 6829, pp. 49–67. Springer (2010)
11. Lee, E., Hong, D., Chang, D., Hong, S., Lim, J.: A Weak Key Class of XTEA for a Related-Key Rectangle Attack. In: Nguyen, P.Q. (ed.) VIETCRYPT 2006. Lecture Notes in Computer Science, vol. 4341, pp. 286–297. Springer (2006)
12. Lu, J.: Cryptanalysis of Reduced Versions of the HIGHT Block Cipher from CHES 2006. In: Nam, K.H., Rhee, G. (eds.) ICISC 2007. LNCS, vol. 4817, pp. 11–26. Springer (2007)
13. Lu, J.: Related-key Rectangle Attack on 36 Rounds of the XTEA Block Cipher. Int. J. Inf. Sec. 8(1), 1–11 (2009)
14. Moon, D., Hwang, K., Lee, W., Lee, S., Lim, J.: Impossible Differential Cryptanalysis of Reduced Round XTEA and TEA. In: Daemen, J., Rijmen, V. (eds.) FSE 2002. LNCS, vol. 2365, pp. 49–60. Springer (2002)
15. Needham, R.M., Wheeler, D.J.: TEA Extensions. Tech. rep., University of Cambridge (Oct 1997)
16. Özen, O., Varici, K., Tezcan, C., Çelebi Kocair: Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT. In: Boyd, C., Nieto, J.M.G. (eds.) ACISP 2009. LNCS, vol. 5594, pp. 90–107. Springer (2009)
17. Sekar, G., Mouha, N., Velichkov, V., Preneel, B.: Meet-in-the-Middle Attacks on Reduced-Round XTEA. In: Kiayias, A. (ed.) Topics in Cryptology - CT-RSA 2011. LNCS, vol. 6558, pp. 250–267. Springer (2011)
18. Wheeler, D.J., Needham, R.M.: TEA, a Tiny Encryption Algorithm. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 363–366. Springer (1994)
19. Zhang, P., Sun, B., Li, C.: Saturation Attack on the Block Cipher HIGHT. In: Garay, J.A., Miyaji, A., Otsuka, A. (eds.) CANS 2009. LNCS, vol. 5888, pp. 76–86. Springer (2009)

Appendix

Fig. 8. 16-Round Impossible Differential from [16]

i	ΔX_7^i	ΔX_6^i	ΔX_5^i	ΔX_4^i	ΔX_3^i	ΔX_2^i	ΔX_1^i	ΔX_0^i
9	$(???????1)_2$	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	$(???????1)_2$
11	0	0	0	0	0	?	$(???????1)_2$	0
12	0	0	0	?	?	$(???????1)_2$	0	0
13	0	?	?	?	$(???????1)_2$	0	0	0
14	?	?	?	$(???????1)_2$	0	0	0	?
15	?	?	$(???????1)_2$	0	0	?	?	?
16	?	$(???????1)_2$	0	?	?	?	?	?
17	$(???????1)_2$	?	?	?	?	?	?	?
17	$(???????0)_2$	$0x80$	?	?	?	?	?	?
18	$0x80$	0	?	?	?	?	?	$(???????1)_2$
19	0	0	?	?	?	?	$(???????1)_2$	$0x80$
20	0	0	?	?	?	$(???????1)_2$	$0x80$	0
21	0	0	?	?	$(???????1)_2$	$0x80$	0	0
22	0	0	?	$(??????100)_2$	$0x80$	0	0	0
23	0	0	$(??????100)_2$	$0x80$	0	0	0	0
24	0	0	$0x80$	0	0	0	0	0
25	0	$0x80$	0	0	0	0	0	0

Fig. 9. Impossible Differential Attack on 27-Round HIGHT

i	ΔX_7^i	ΔX_6^i	ΔX_5^i	ΔX_4^i	ΔX_3^i	ΔX_2^i	ΔX_1^i	ΔX_0^i
P	?	?	?	?	?	?	$(???????1)_2$	0
3	?	?	?	?	?	?	$(???????1)_2$	0
4	?	?	?	?	?	$(???????1)_2$	0	0
5	?	?	?	?	$(???????1)_2$	0	0	0
6	?	?	?	$(???????1)_2$	0	0	0	0
7	?	?	$(???????1)_2$	0	0	0	0	0
8	?	$(???????1)_2$	0	0	0	0	0	0
9	$(???????1)_2$	0	0	0	0	0	0	0
Impossible Differential								
25	0	$0x80$	0	0	0	0	0	0
26	$0x80$	0	0	0	0	0	0	b
27	0	0	0	0	0	?	b	$0x80$
28	0	0	0	?	?	$(???????1)_2$	$0x80$	0
29	0	?	?	?	$(???????1)_2$	$0x80$	0	0
30	?	?	?	$(???????0)_2$	$0x80$	0	0	?
C	?	?	?	?	$(???????0)_2$	$0x80$	0	0

Table 5. Subkeys Used in the Attack on 26-Round HIGHT

#Round	Subkey Used			
5	$\boldsymbol{SK_{19}(MK_2)}$	$\boldsymbol{SK_{18}(MK_1)}$	$\boldsymbol{SK_{17}(MK_0)}$	$\boldsymbol{SK_{16}(MK_7)}$
6	$\boldsymbol{SK_{23}(MK_6)}$	$\boldsymbol{SK_{22}(MK_5)}$	$\boldsymbol{SK_{21}(MK_4)}$	$\boldsymbol{SK_{20}(MK_3)}$
7	$\boldsymbol{SK_{27}(MK_{10})}$	$SK_{26}(MK_9)$	$\boldsymbol{SK_{25}(MK_8)}$	$\boldsymbol{SK_{24}(MK_{15})}$
8	$SK_{31}(MK_{14})$	$SK_{30}(MK_{13})$	$\boldsymbol{SK_{29}(MK_{12})}$	$\boldsymbol{SK_{28}(MK_{11})}$
9	$SK_{35}(MK_1)$	$SK_{34}(MK_0)$	$\boldsymbol{SK_{33}(MK_7)}$	$SK_{32}(MK_6)$
...	...	...	...	...
26	$SK_{103}(MK_1)$	$SK_{102}(MK_0)$	$\boldsymbol{SK_{101}(MK_7)}$	$SK_{100}(MK_6)$
27	$SK_{107}(MK_{13})$	$\boldsymbol{SK_{106}(MK_{12})}$	$\boldsymbol{SK_{105}(MK_{11})}$	$SK_{104}(MK_{10})$
28	$\boldsymbol{SK_{111}(MK_9)}$	$\boldsymbol{SK_{110}(MK_8)}$	$\boldsymbol{SK_{109}(MK_{15})}$	$SK_{108}(MK_{14})$
29	$\boldsymbol{SK_{115}(MK_4)}$	$\boldsymbol{SK_{114}(MK_3)}$	$\boldsymbol{SK_{113}(MK_2)}$	$\boldsymbol{SK_{112}(MK_1)}$
30	$\boldsymbol{SK_{119}(MK_0)}$	$\boldsymbol{SK_{118}(MK_7)}$	$\boldsymbol{SK_{117}(MK_6)}$	$\boldsymbol{SK_{116}(MK_5)}$
Post-Whitening	$\boldsymbol{WK_7(MK_3)}$	$\boldsymbol{WK_6(MK_2)}$	$\boldsymbol{WK_5(MK_1)}$	$\boldsymbol{WK_4(MK_0)}$

Table 6. Subkeys Used in the Attack on 27-Round HIGHT

#Round	Subkey Used			
Pre-Whitening	$\boldsymbol{WK_3(MK_{15})}$	$\boldsymbol{WK_2(MK_{14})}$	$\boldsymbol{WK_1(MK_{13})}$	$\boldsymbol{WK_0(MK_{12})}$
4	$\boldsymbol{SK_{15}(MK_{15})}$	$\boldsymbol{SK_{14}(MK_{14})}$	$\boldsymbol{SK_{13}(MK_{13})}$	$\boldsymbol{SK_{12}(MK_{12})}$
5	$\boldsymbol{SK_{19}(MK_2)}$	$\boldsymbol{SK_{18}(MK_1)}$	$\boldsymbol{SK_{17}(MK_0)}$	$\boldsymbol{SK_{16}(MK_7)}$
6	$\boldsymbol{SK_{23}(MK_6)}$	$\boldsymbol{SK_{22}(MK_5)}$	$\boldsymbol{SK_{21}(MK_4)}$	$\boldsymbol{SK_{20}(MK_3)}$
7	$\boldsymbol{SK_{27}(MK_{10})}$	$\boldsymbol{SK_{26}(MK_9)}$	$\boldsymbol{SK_{25}(MK_8)}$	$SK_{24}(MK_{15})$
8	$\boldsymbol{SK_{31}(MK_{14})}$	$\boldsymbol{SK_{30}(MK_{13})}$	$SK_{29}(MK_{12})$	$SK_{28}(MK_{11})$
9	$\boldsymbol{SK_{35}(MK_1)}$	$SK_{34}(MK_0)$	$SK_{33}(MK_7)$	$SK_{32}(MK_6)$
...	...	...	...	...
26	$\boldsymbol{SK_{103}(MK_1)}$	$SK_{102}(MK_0)$	$SK_{101}(MK_7)$	$SK_{100}(MK_6)$
27	$\boldsymbol{SK_{107}(MK_{13})}$	$SK_{106}(MK_{12})$	$SK_{105}(MK_{11})$	$\boldsymbol{SK_{104}(MK_{10})}$
28	$\boldsymbol{SK_{111}(MK_9)}$	$SK_{110}(MK_8)$	$\boldsymbol{SK_{109}(MK_{15})}$	$\boldsymbol{SK_{108}(MK_{14})}$
29	$\boldsymbol{SK_{115}(MK_4)}$	$\boldsymbol{SK_{114}(MK_3)}$	$\boldsymbol{SK_{113}(MK_2)}$	$\boldsymbol{SK_{112}(MK_1)}$
30	$\boldsymbol{SK_{119}(MK_0)}$	$\boldsymbol{SK_{118}(MK_7)}$	$\boldsymbol{SK_{117}(MK_6)}$	$\boldsymbol{SK_{116}(MK_5)}$
Post-Whitening	$\boldsymbol{WK_7(MK_3)}$	$\boldsymbol{WK_6(MK_2)}$	$\boldsymbol{WK_5(MK_1)}$	$\boldsymbol{WK_4(MK_0)}$

Table 7. Key Recovery Procedure of the Attack on 27-Round HIGHT

Step	Guess Bits	Known Keys	Complexity	Conds	Pairs Left	Sieve on
1	MK_{15}	WK_3, SK_{15}	$2^{91.2}$ EN	8	2^{79}	(4,3)
2	MK_0, MK_3	WK_7, SK_{119}	$2^{99.2}$ EN	8	2^{71}	(30,3)
3	MK_6, MK_1	SK_{117}, $WK_{5,8}$	$2^{107.2}$ EN	–	2^{71}	–
4	MK_{14}	WK_1, SK_{14}	$2^{115.2}$ EN	–	2^{71}	–
5	MK_2 †	SK_{19}	$2^{118.2}$ EN	8	2^{63}	(5,3)
6	–	SK_{113}, SK_{109}	$2^{115.2}$ EN	–	2^{63}	– (calculate ΔX_4^{28})
7	MK_7 †	WK_6, SK_{118}	$2^{118.2}$ EN	8	2^{55}	(30,2)
8	–	SK_{114}	$2^{115.2}$ EN	8	2^{47}	(29,2)
9	MK_{13}	$SK_{13}, SK_{18}, SK_{23}$	$2^{115.2}$ EN	8	2^{39}	(6,3)
10	MK_5	WK_4, SK_{116}, SK_{108}	$2^{115.2}$ EN	–	2^{39}	–
11	MK_{10} †	SK_{104}	$2^{118.2}$ EN	8	2^{31}	(27,0)
12	MK_{12}	$WK_0, SK_{12}, SK_{17}, SK_{22}, SK_{27}$	$2^{123.2}$ EN	8	2^{23}	(7,3)
13	MK_4	$SK_{16}, SK_{21}, SK_{26}, SK_{31}$ $SK_{107}, SK_{111}, SK_{115}$	2^{120} MA + $2^{126.6}$ EN			(pre-com)

MA: memory accesses; EN: 27-round HIGHT encryptions
† The key byte is guessed bit by bit from the LSB to the MSB.
pre-com: the sieving is already done by precomputation

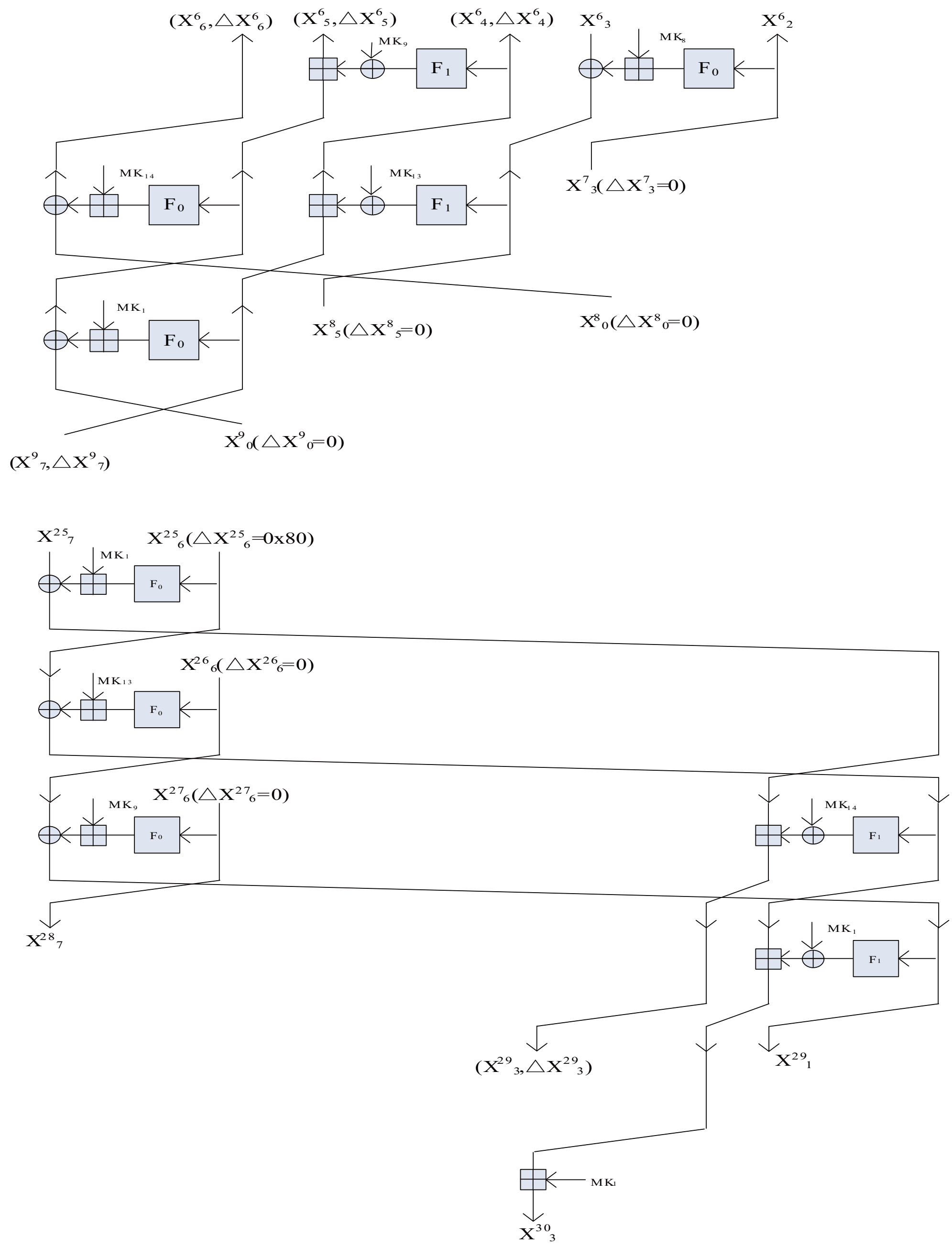

Fig. 10. Precomputation for Rounds 7 ∼ 9 (top) and Rounds 26 ∼ 30 (bottom) in the 27-Round Attack on HIGHT