Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT

Onur Özen[1], Kerem Varıcı[2,⋆], Cihangir Tezcan[3], and Çelebi Kocair[4]

[1] EPFL IC LACAL Station 14. CH-1015 Lausanne, Switzerland
onur.ozen@epfl.ch
[2] K.U.Leuven, Dept. of Electrical Engineering, ESAT/SCD/COSIC and IBBT
Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium
kerem.varici@esat.kuleuven.be
[3] METU, Institute of Applied Mathematics,
Department of Cryptography, 06531 Ankara, Turkey
cihangir@metu.edu.tr
[4] METU, Department of Computer Engineering, 06531 Ankara, Turkey
celebi@ceng.metu.edu.tr

Abstract. Design and analysis of lightweight block ciphers have become more popular due to the fact that the future use of block ciphers in ubiquitous devices is generally assumed to be extensive. In this respect, several lightweight block ciphers are designed, of which Present and Hight are two recently proposed ones by Bogdanov *et al.* and Hong *et al.* respectively. In this paper, we propose new attacks on Present and Hight. Firstly, we present the first related-key cryptanalysis of 128-bit keyed Present by introducing 17-round related-key rectangle attack with time complexity approximately 2^{104} memory accesses. Moreover, we further analyze the resistance of Hight against impossible differential attacks by mounting new 26-round impossible differential and 31-round related-key impossible differential attacks where the former requires time complexity of $2^{119.53}$ reduced round Hight evaluations and the latter is slightly better than exhaustive search.

Keywords: Present, Hight, Related-Key Attack, Rectangle Attack, Impossible Differential Attack.

1 Introduction

Lightweight cryptography has become very vital with the emerging needs in sensitive applications like RFID (Radio-frequency identification) systems and

[⋆] This work was sponsored by the Research Fund K.U.Leuven, by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy) and by the European Commission through the ICT Programme under Contract ICT-2007-216676 (ECRYPT II). The information in this paper is provided as is, and no warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

C. Boyd and J. González Nieto (Eds.): ACISP 2009, LNCS 5594, pp. 90–107, 2009.

sensor networks. For these types of special purposes, there is a strong demand in designing secure lightweight cryptographic modules. After the selection of AES (Advanced Encryption Standard) [1], the research on efficient implementation of AES, especially for such constrained environments, brought special attention in research community. Even though it is highly convenient for such devices, the research on designing and analyzing new lightweight block ciphers that are more efficient than AES on these platforms poses huge challenges. For this purpose, several block ciphers are designed as potential candidates such as Hight [2,3], Present [4], mCrypton [5], SEA [6], CGEN [7], DES [8] and DESXL [8][1].

A recent portfolio[2], which contains four software and three hardware oriented stream ciphers [11], has been announced by ECRYPT as part of eSTREAM project to identify new stream ciphers that might become suitable for widespread adoption including lightweight platforms. As a result, stream ciphers are shown to be highly efficient on both software and hardware implementations comparing to block ciphers. To fill this efficiency gap, Present [4] was proposed by Bogdanov *et al.* at CHES '07 as an ultra-lightweight block cipher with 31 rounds offering as good hardware and software performance as current modern stream ciphers while it is more efficient than many known block ciphers.

Basic security analysis of Present is provided in [4] by showing resistance against known attacks such as differential, linear cryptanalysis and their variants. Recent differential attacks [12,13] on 16 and 19 rounds of Present provide similar results as in the original proposal with some practical evidence of applied characteristics where the latter is an attempt to combine algebraic attacks with differential cryptanalysis. Another type of an attack called *bit-pattern based integral attack* [14] is applicable up to seven rounds of Present. More recently, a new type of attack, called *statistical saturation attack* was proposed in [15] and shown to be applicable up to 24 rounds of Present. Previous results on the analysis of Present are summarized in Table 1.

The security of Present against key schedule weaknesses is provided by showing the resistance against slide [16] and related-key differential attacks [17] where slide attacks are inapplicable because of the round dependent counters in key scheduling algorithm. Related-key differential attacks, on the other hand, are also believed to be inapplicable because of the sufficient non-linearity due to key scheduling algorithm.

Hight [2,3] is a South Korean standard encryption algorithm enjoying the use of a low-resource hardware implementation. It is a 32 round block cipher proposed one year before Present at CHES '06 by Hong *et al.* to be used for ubiquitous computing devices. The prominent characteristic of Hight is that it makes use of simple byte oriented operations such as exclusive-or, addition modulo 256 and cyclic rotation which offers nice performance on hardware.

[1] TEA [9] and XTEA [10] can also be given as lightweight block ciphers which were designed before AES.

[2] The original hardware-oriented portfolio of eSTREAM contains four hardware-oriented stream ciphers. However, F-FSCR-H has recently been eliminated from the eSTREAM portfolio.

Table 1. Summary of the attacks on PRESENT and HIGHT (CP-Chosen Plaintext, MA-Memory Accesses, PR-Reduced round PRESENT evaluation, HE-Reduced round HIGHT evaluation)

Cipher	Rounds	Key Size	Attack Type	Data Complexity	Time Complexity	Memory Complexity	Reference
PRESENT	24	80	Stat. Sat.	2^{60}CP	2^{20} PR	2^{16} bytes	[15]
	24	80	Stat. Sat.	2^{57}CP	2^{57} PR	2^{32} bytes	[15]
	7	128	Bit-Pat. Int.	$2^{24.3}$CP	$2^{100.1}$ MA	2^{77} bytes	[14]
	17	128	Rel.-Key Rec.	2^{63} CP	2^{104} MA	2^{53} bytes	§3.1
	19	128	Alg.-Dif.	6×2^{62} CP	2^{113} MA	not specified	[13]
HIGHT	18	128	Imp. Dif.	$2^{46.8}$ CP	$2^{109.2}$ HE	not specified	[2]
	25	128	Imp. Dif.	2^{60} CP	$2^{126.78}$HE	not specified	[18]
	26	128	Imp. Dif.	2^{61} CP	$2^{119.53}$HE	2^{109} bytes	§4.1
	26	128	Rel.-Key Rec.	$2^{51.2}$ CP	$2^{120.41}$HE	not specified	[18]
	28	128	Rel.-Key Imp.	2^{60} CP	$2^{125.54}$HE	not specified	[18]
	31	128	Rel.-Key Imp.	2^{64} CP	$2^{127.28}$HE	2^{117} bytes	§4.2

The security of HIGHT is investigated in [2] by showing resistance against differential, linear, truncated differential, boomerang, rectangle, impossible differential attacks and their related-key variants. In [2], the safety margin was shown to be 13 rounds as the best attack covers 19 rounds. Recent serious attacks [19] by Lu on reduced round HIGHT make use of 25, 26 and 28 round impossible differential, related-key rectangle and related-key impossible differential attacks: the last attack is the best attack on HIGHT so far that reduced the safety margin from 13 rounds to four rounds.

In this work, we present the first related-key cryptanalysis of PRESENT. For 128-bit keyed version, we introduce 17-round related-key rectangle attack [20,21,22] which is not explicitly mentioned in the original proposal [4]. Moreover, we further analyze the resistance of HIGHT against impossible differential attacks [23,24]. Firstly, we improve 25-round impossible differential attack of Lu by introducing a new characteristic to 26 rounds and update 28-round related-key impossible differential attack on 31 rounds. To the best of our knowledge, these are the best cryptanalytic results on HIGHT. We provide a summary of our results in Table 1.

The organization of the paper is as follows. In Section 2, we give a brief description of the block ciphers PRESENT and HIGHT. Section 3 introduces the idea behind the related-key attacks on PRESENT and contains related-key rectangle attack on 17-round. In Section 4, we introduce our improved impossible differential and related-key impossible differential attacks on reduced round HIGHT. We conclude with Section 5 and provide supplementary details about the paper in Appendices.

2 The Block Ciphers PRESENT and HIGHT

2.1 Notation

For PRESENT and HIGHT, we use the same notation to denote the variables used in this paper. For the sake of clarity and the parallelism with the previous work

Table 2. Notation

$\oplus$	Bitwise logical exclusive OR (XOR)
$\boxplus$	Addition modulo 2^8
$\lll$i	Left cyclic rotation by i bits
PRESENT-n-r	PRESENT reduced to r-rounds with n-bit secret key
K_i	ith subkey of PRESENT
S_i	ith S-Box of PRESENT
$e_{j_1,\ldots,j_k}$	A word with zeros in all positions but bits $j_1,\ldots,j_k$
HIGHT-r	HIGHT reduced to r-rounds
e_j	A byte with zeros in all positions but bit j $(0 \leqslant j \leqslant 7)$
$e_{j,\sim}$	A byte that has zeros in bits 0 to $j-1$, a one in bit j and indeterminate values in bits $(j+1)$ to 7
$e_{\bar{j},\sim}$	A byte that has zeros in bits 0 to j and indeterminate values in bits $(j+1)$ to 7
?	An arbitrary byte
$X_{i,j}$	jth byte of state variable of round i of HIGHT, $(0 \leqslant j \leqslant 7)$ $(0 \leqslant i \leqslant 32)$
MK_i	ith secret key byte of HIGHT
WK_i	ith whitening key byte of HIGHT
SK_i	ith subkey byte of HIGHT

[19], we use exactly the same notation for HIGHT which is provided in Table 2. Throughout the paper, it is assumed that the rounds are numbered from zero and the leftmost bit is the most significant bit in a byte or a word.

2.2 PRESENT

PRESENT is a 31-round (and an output whitening at the end) SPN (Substitution Permutation Network) type block cipher with block size of 64 bits that supports 80 and 128-bit secret key. Round function of PRESENT, which is depicted in Figure 1, is same for both versions of PRESENT and consists of standard operations such as subkey XOR, substitution and permutation: At the beginning of each round, 64-bit input of the round function is XORed with the subkey. Just after the subkey XOR, 16 identical 4 × 4-bit S-boxes are used in parallel as a non-linear substitution layer and finally a permutation is performed so as to provide diffusion.

The subkeys for each round are derived from the user-provided secret key by the key scheduling algorithm. We provide only the details of the key scheduling algorithm of PRESENT-128 as it is the main target of this paper: 128-bit secret key is stored in a key register K and represented as $k_{127}k_{126}\ldots k_0$. The subkeys K_i $(0 \leq i \leq 31)$ consist of 64 leftmost bits of the actual content of register K. After round key K_i is extracted, the key register K is rotated by 61 bit positions

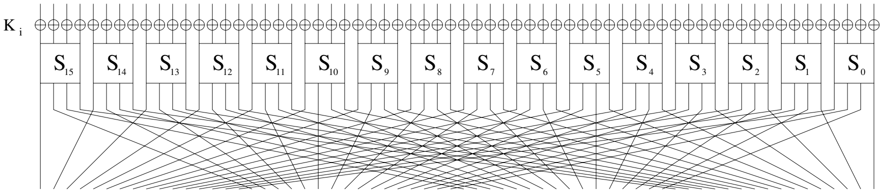

Fig. 1. Round function of PRESENT

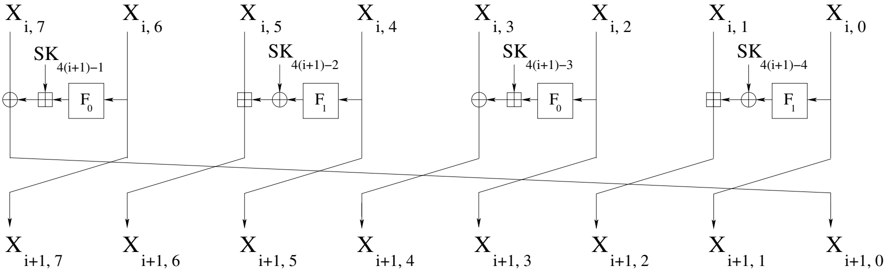

Fig. 2. ith round of HIGHT for $i = 0, \ldots, 31$

to the left, then S-box is applied to the left-most eight bits of the key register and finally the round counter value, which is a different constant for each round, is XORed with bits $k_{66}k_{65}k_{64}k_{63}k_{62}$. Further details about the specification of PRESENT are provided in [4].

2.3 HIGHT

HIGHT is a 32-round block cipher with 64-bit block size and 128-bit user key that makes use of an unbalanced Feistel Network. The encryption function starts with an Initial Transformation (IT) that is applied to plaintexts together with input whitening keys WKs. At the end of 32 rounds, in order to obtain the ciphertexts, a Final Transformation (FT) is applied to the output of the last round together with an output whitening. The byte-oriented round function, shown in Figure 2, uses modular addition, XOR and linear subround functions F_0 and F_1; the latter can be described as follows:

$$\begin{array}{rcl} F_0(x) & = & (x \lll 1) \oplus (x \lll 2) \oplus (x \lll 7) \\ F_1(x) & = & (x \lll 3) \oplus (x \lll 4) \oplus (x \lll 6) \end{array}$$

HIGHT only works with 128-bit secret key MK which is treated as 16 bytes, $(MK_{15}, \ldots, MK_0)$. The key schedule of HIGHT uses additional constants to avoid the self similarity in the key scheduling algorithm which prevents cipher from slide attacks. Input-output whitening keys and round subkeys are obtained by permuting the 16 bytes of the original key and using addition with constants. Table 9 will be extensively used in this paper that shows the relations between the original and the subkey bytes. Namely, each value in a row represents the obtained whitening and subkey bytes once the corresponding byte in the first column of the same row of the original key is known. Further details about the specification of HIGHT are provided in [2,3].

3 The Related-Key Attacks on PRESENT

The idea behind the related-key attacks on PRESENT is to benefit from the slow mixing in the key scheduling algorithm which makes use of only one or two S-box operations (depending on the version) during each iteration. To achieve this goal, we made an efficient search for related-key differentials of Present which was done by flipping at most two bits of the original key. The crucial part of the key differentials is that we only consider the trivial differentials. More precisely, all reduced round key differentials in our attacks work with probability one.

In the original proposal of Present [4], the resistance against differential and linear attacks are given by the bounds provided by the minimum number of active S-boxes. This approach also works for showing resistance against wide variety of attacks. A recent differential attack [12] uses same idea to attack the cipher by increasing the overall probability of the characteristics more effectively. Although there is no contradiction with the security claims given in [4], the differential attack in [12] provides a practical evidence. In this work, however, our aim is quite different and simple in that we try to decrease the number of active S-boxes (NAS). In order to do so, we cancel the intermediate differences with the subkey differences and construct our differentials by activating at most five S-boxes at the beginning. At the end, we are able to construct related-key differentials having less active S-boxes than given in [12]. As an example, for Present-80, the minimum number of active S-boxes for any five-round differential characteristic is given to be ten in [4,12]. However, we found several five-round related-key differentials with only three active S-boxes.

Although it seems quite promising, as the number of rounds increases, the minimum number of active S-boxes gets closer to the one given in the original proposal [12] and the overall probabilities of the characteristics are not optimal. Still, for less number of rounds the related-key differentials are efficient and the number of possible characteristics are quite high. So, attacks like the related-key rectangle attack are easily applicable.

3.1 The Related-Key Rectangle Attack on PRESENT-128-17

The related-key rectangle attack is the clever extension of differential cryptanalysis. In rectangle-boomerang style attacks, the attacker uses two short differential characteristics instead of one long differential characteristic. The aim is to benefit from the slow mixing in relatively reduced round versions of the attacked cipher. We provide a brief description about the related-key rectangle attack in Appendix A and follow the mounted attack on Present. Throughout the paper, the related-key rectangle attack is assumed to be mounted by using four related keys.

Let E denote the encryption function of Present-n-r. We treat E as a cascade of four subciphers as $E = E_f \circ E_1 \circ E_0 \circ E_b$ where E is composed of a core $E' = E_1 \circ E_0$ covered by additional rounds, E_b and E_f which are the subciphers before and after the core function respectively.

For the related-key rectangle attack on Present-128-17, we use the following decomposition: E_0 starts with the first round and ends just after the subkey XOR in round eight. E_1, on the other hand, commences with the substitution layer in round eight and stops at the end of round 14[3]. Round 0 and round

[3] This decomposition is not unique and can be done in various ways.

Table 3. An example of related-key differential used in E_0

r	Input Difference $\Delta(I)$	Key Difference $\Delta(K)$	$\Delta(I) \oplus \Delta(K)$	Output Difference $\Delta(O)$	NAS	P
1	00000000000000bb	0000000000000000	00000000000000bb	0003000000000000	2	2^{-4}
2	0003000000000000	0003000000000000	0000000000000000	0000000000000000	0	1
3	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0	1
4	0000000000000000	00000c0000000000	00000c0000000000	0000000004000000	1	2^{-3}
5	0000000004000000	0000000000000000	0000000004000000	0000004000000040	1	2^{-2}
6	0000004000000040	0000003000000000	0000007000000040	0000000200000202	2	2^{-4}
7	0000000200000202	0000000000000000	0000000200000202	0000010500000105	3	2^{-6}
8	0000010500000105	00000000c0000000	00000105c0000105			1

$15 - 16$ serve as the round before and after the distinguisher respectively (E_b and E_f)[4].

All the differentials used in E_0 have the same input difference $\alpha = e_{0,1,3,4,5,7}$ and they all work with the key difference $\Delta K^{12} = e_{118,119}$. There are at least 343 such characteristics with varying differences at the beginning of the seventh round: there exist one characteristics of probability $p = 2^{-19}$, 18 characteristics of probability $p = 2^{-20}$, 108 characteristics of probability $p = 2^{-21}$ and 216 characteristics of probability $p = 2^{-22}$. Therefore the overall probability for E_0 is $\hat{p} = \sqrt{1 \cdot (2^{-19})^2 + 18 \cdot (2^{-20})^2 + 108 \cdot (2^{-21})^2 + 216 \cdot (2^{-22})^2} \approx 2^{-17}$. Table 3 shows one of the characteristics used for E_0.

Given the α difference, the number of active S-boxes in E_b which lead to an α difference in round 1 can be found by applying the inverse permutation to α difference. This leads to six active S-boxes in the first round with varying output differences after the substitution layer. These six S-boxes are used to create the α difference before the core function. Since the output difference is known for all active S-boxes in round 0, namely 1_x, not all of the input differences are possible. The number of possible input differences is only $2^{15.5}$ instead of 2^{24}.

For the second subcipher E_1, we use the fixed output difference $\delta = e_{11,15}$ under the key difference $\Delta K^{13} = e_{117,121}$. The most efficient characteristic is provided in Table 4 with probability $p = 2^{-12}$. As it can be seen from the table, the overall probability for E_1 is $\hat{q} \approx 2^{-12}$. Thus, the probability of the related-key rectangle distinguisher is given by $Pr = 2^{-64}\hat{p}^2\hat{q}^2 \approx 2^{-122}$. Similarly, the subcipher E_f after the core function can be defined by letting δ propagate. In round 15, there exist two active S-boxes with input differences 8_x each leading to six output differences and these outputs are diffused to six different S-boxes after key addition in round 16 which produce $2^{21.22}$ possible output differences in total out of 2^{24}.

To attack 17 round Present, we request 2^{39} structures of 2^{24} plaintexts each. The structures are chosen in such a way that each structure varies all over the possible inputs to the active S-boxes in E_b, while the differences for the other S-boxes are kept zero. Our aim is to get an α difference at the beginning of E_0. This technique for choosing plaintexts lets us 2^{47} pairs in total for each structure in which 2^{23} of them satisfy α difference before E_0. Thus, the total

[4] We exclude the output whitening in our attack.

Table 4. An example of related-key differential used in E_1

r	Output Difference $\Delta(O)$	Key Difference $\Delta(K)$	$\Delta(I) \oplus \Delta(K)$	$P^{-1}(S^{-1}(I \oplus K))$	NAS	P
14	0000000000008800	0000000000008800	0000000000000000	0000000000000000	0	1
13	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0	1
12	0000000000000000	0000000000220000	0000000000220000	0000000000600060	2	2^{-6}
11	0000000000600060	0000000000000000	0000000000600060	0000000008800000	2	2^{-6}
10	0000000008800000	0000000008800000	0000000000000000	0000000000000000	0	1
9	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0	1
8			0000000000000000	0000000000000000		

number of pairs with an α difference before the core function is 2^{62} that produce approximately 2^{124} quartets of which $2^{124} \cdot 2^{-64} \cdot 2^{-58} = 2^2 = 4$ are expected to be *right*. The overall attack works as follows:

1. Generate 2^{39} structures of 2^{24} plaintexts and encrypt each structure of plaintexts with K^1, K^2, K^3 and K^4 to obtain the corresponding pool of ciphertexts C^j where $1 \leq j \leq 4$.
   - This step requires data complexity of 2^{63} chosen plaintexts and time complexity of 2^{65} Present-128-17 encryptions.
2. Generate $2^{24+8+24} = 2^{56}$ counters each of which corresponds to a different key guess in E_b and E_f respectively.
   - Time complexity of this step is 2^{56} memory accesses.
3. Insert 2^{65} ciphertexts (C^1, C^3) and (C^2, C^4) into hash tables (T^1_{13}, T^3_{13}) and (T^2_{24}, T^4_{24}) respectively indexed by 40 (expected inactive) bits. If a collision occurs in the same bins of (T^1_{13}, T^3_{13}) and (T^2_{24}, T^4_{24}), check whether the differences of the collided ciphertexts are one of the $2^{21.22}$ expected ciphertext differences.
   - This step has time complexity of 2^{65} memory accesses from inserting all the ciphertext into hash tables. In the hash tables there exist 2^{40} bins and in each bin we expect to have 2^{23} ciphertexts. Therefore, we can form $(2^{23})^2 = 2^{46}$ pairs where one of the components from $T^1_{13}(T^2_{24})$ and the other is from $T^3_{13}(T^4_{24})$. That makes 2^{86} pairs in total for each pair of tables (T^1_{13}, T^3_{13}) and (T^2_{24}, T^4_{24}). In order to check whether colliding ciphertexts' differences are one of the expected ciphertext differences we have to make 2^{87} memory accesses in total. The number of remaining pairs is $2^{86-2.78} = 2^{83.22}$ for each of (T^1_{13}, T^3_{13}) and (T^2_{24}, T^4_{24}) since the probability of having expected difference in the ciphertexts is $2^{21.22-24} = 2^{-2.78}$.
4. For each surviving pair (C_1, C_3) (and (C_2, C_4)) from the previous step, find the corresponding plaintext pairs (P_1, P_3) (and (P_2, P_4)) from the structures. For each such pair, check whether $P_1 \oplus P_2$ satisfies the required difference in E_b. If this check succeeds, examine the ciphertexts C_3 and C_4 that collided with C_1 and C_2 respectively. If the difference between the corresponding plaintexts P_3 and P_4 also satisfies the required difference in E_b, continue to analyze the quartet $((P_1, P_2), (P_3, P_4))$.

- The probability that $P_1 \oplus P_2$ satisfies the required difference is $2^{15.5-24} = 2^{-8.5}$, if they are in the same structure. So, the probability that the required difference is satisfied is $2^{-8.5-39} = 2^{-47.5}$ under the assumption of uniform distribution of plaintexts and structures. This reduces the number of pairs satisfying the condition in (T^1_{13}, T^3_{13}) to $2^{83.22-47.5} = 2^{35.72}$. Similarly, there exist $2^{35.72}$ pairs in (T^2_{24}, T^4_{24}). Thus, we can form $(2^{35.72})^2 = 2^{71.44}$ quartets satisfying the conditions in E_b and E_f. In order to do this filtering we have to make $2^{84.22}$ memory accesses.

5. For each remaining quartet $((P_1, P_2), (P_3, P_4)), ((C_1, C_2), (C_3, C_4))$ and every possible subkey value (k_b and k_f independently) of E_b and E_f test whether

$$\begin{aligned}
&E_{b_{k_b}}(P_1) \oplus E_{b_{k_b^{'}}}(P_2) = E_{b_{k_b^{''}}}(P_3) \oplus E_{b_{k_b^{'''}}}(P_4) = \alpha && \text{where } k_b^{'} = k_b \oplus \Delta K^{12} \\
& && \text{and } k_b^{''} = k_b^{'''} \oplus \Delta K^{12}, \\
&E^{-1}_{f_{k_f}}(C_1) \oplus E^{-1}_{f_{k_f^{''}}}(C_3) = E^{-1}_{f_{k_f^{'}}}(C_2) \oplus E^{-1}_{f_{k_f^{'''}}}(P_4) = \delta && \text{where } k_f^{''} = k_f \oplus \Delta K^{13} \\
& && \text{and } k_f^{'} = k_f^{'''} \oplus \Delta K^{13} \text{ hold.}
\end{aligned}$$

If this is the case, increment the counters that correspond to k_b, k_f.
   - In this step, every surviving quartet is partially encrypted (in E_b) and decrypted (in E_f) independently. As the number of subkeys guessed are more in E_f than in E_b, the overall complexity of this step is $2^{71.44+32} = 2^{103.44}$ memory accesses and half round decryptions (as eight S boxes are affected in total); the latter is equivalent to 2^{99} Present-128-17 evaluations.

6. Output the subkeys whose counters are maximal.
   - This step requires 2^{56} memory accesses.

Since α difference after E_b can be obtained from $2^{15.5}$ input differences in step 5, the probability that the intermediate difference is α before the core function is $2^{-15.5}$ on average. Thus, each subkey is suggested by a quartet with probability 2^{-31}. Similarly, the probability that the difference after the core function has an δ difference is $2^{-21.22}$ on average leading to $2^{-42.44}$ of the subkeys by the quartets. Each of the $2^{71.44}$ quartets that enter step 5 suggests $2^{56-2\times15.5-2\times21.22} = 2^{-17.44}$ subkeys, so the total number of suggested subkeys is about 2^{54}. As there are 2^{56} subkeys, the expected number of times a wrong subkey is suggested is about 2^{-2}. This means that we can find the right subkey or at least discard almost all the wrong subkeys.

Thus, the overall attack has memory complexity of 2^{53} bytes, time complexity of 2^{104} memory accesses and data complexity of 2^{63} chosen plaintexts. The expected number of right quartets is taken to be four.

4 Impossible Differential Attacks on HIGHT

In this section, we introduce improved impossible differential attack on 26-round and related-key impossible differential attack on 31-round Hight which utilize

Table 5. 26-Round impossible differential

	$\Delta X_{i,7}$	$\Delta X_{i,6}$	$\Delta X_{i,5}$	$\Delta X_{i,4}$	$\Delta X_{i,3}$	$\Delta X_{i,2}$	$\Delta X_{i,1}$	$\Delta X_{i,0}$	Subkeys			
ΔX_0	?	?	?	?	?	$e_{0,\sim}$	0	0	SK_3	SK_2	SK_1	SK_0
ΔX_1	?	?	?	?	$e_{0,\sim}$	0	0	0	SK_7	SK_6	SK_5	SK_4
ΔX_2	?	?	?	$e_{0,\sim}$	0	0	0	0	SK_{11}	SK_{10}	SK_9	SK_8
ΔX_3	?	?	$e_{0,\sim}$	0	0	0	0	0	SK_{15}	SK_{14}	SK_{13}	SK_{12}
ΔX_4	?	$e_{0,\sim}$	0	0	0	0	0	0	SK_{19}	SK_{18}	SK_{17}	SK_{16}
ΔX_5	$e_{0,\sim}$	0	0	0	0	0	0	0	SK_{23}	SK_{22}	SK_{21}	SK_{20}
ΔX_6	0	0	0	0	0	0	0	$e_{0,\sim}$	SK_{27}	SK_{26}	SK_{25}	SK_{24}
ΔX_7	0	0	0	0	0	?	$e_{0,\sim}$	0	SK_{31}	SK_{30}	SK_{29}	SK_{28}
ΔX_8	0	0	0	?	?	$e_{0,\sim}$	0	0	SK_{35}	SK_{34}	SK_{33}	SK_{32}
ΔX_9	0	?	?	?	$e_{0,\sim}$	0	0	0	SK_{39}	SK_{38}	SK_{37}	SK_{36}
ΔX_{10}	?	?	?	$e_{0,\sim}$	0	0	0	?	SK_{43}	SK_{42}	SK_{41}	SK_{40}
ΔX_{11}	?	?	$e_{0,\sim}$	0	0	?	?	?	SK_{47}	SK_{46}	SK_{45}	SK_{44}
ΔX_{12}	?	$e_{0,\sim}$	0	?	?	?	?	?	SK_{51}	SK_{50}	SK_{49}	SK_{48}
ΔX_{13}	$e_{0,\sim}$	?	?	?	?	?	?	?	SK_{55}	SK_{54}	SK_{53}	SK_{52}
ΔX_{13}	$e_{\bar{0},\sim}$	80_x	?	?	?	?	?	?	SK_{55}	SK_{54}	SK_{53}	SK_{52}
ΔX_{14}	80_x	0	?	?	?	?	?	$e_{0,\sim}$	SK_{59}	SK_{58}	SK_{57}	SK_{56}
ΔX_{15}	0	0	?	?	?	?	$e_{0,\sim}$	80_x	SK_{63}	SK_{62}	SK_{61}	SK_{60}
ΔX_{16}	0	0	?	?	?	$e_{0,\sim}$	80_x	0	SK_{67}	SK_{66}	SK_{65}	SK_{64}
ΔX_{17}	0	0	?	?	$e_{0,\sim}$	80_x	0	0	SK_{71}	SK_{70}	SK_{69}	SK_{68}
ΔX_{18}	0	0	?	$e_{2,\sim}$	80_x	0	0	0	SK_{75}	SK_{74}	SK_{73}	SK_{72}
ΔX_{19}	0	0	$e_{2,\sim}$	80_x	0	0	0	0	SK_{79}	SK_{78}	SK_{77}	SK_{76}
ΔX_{20}	0	0	80_x	0	0	0	0	0	SK_{83}	SK_{82}	SK_{81}	SK_{80}
ΔX_{21}	0	80_x	0	0	0	0	0	0	SK_{87}	SK_{86}	SK_{85}	SK_{84}
ΔX_{22}	80_x	0	0	0	0	0	0	$e_{0,\sim}$	SK_{91}	SK_{90}	SK_{89}	SK_{88}
ΔX_{23}	0	0	0	0	0	?	$e_{0,\sim}$	80_x	SK_{95}	SK_{94}	SK_{93}	SK_{92}
ΔX_{24}	0	0	0	?	?	$e_{0,\sim}$	80_x	0	SK_{99}	SK_{98}	SK_{97}	SK_{96}
ΔX_{25}	0	?	?	?	$e_{0,\sim}$	80_x	0	0	SK_{103}	SK_{102}	SK_{101}	SK_{100}
ΔX_{26}	?	?	?	$e_{\bar{0},\sim}$	80_x	0	0	?	WK_7	WK_6	WK_5	WK_4
FT	?	?	?	?	$e_{\bar{0},\sim}$	80_x	0	0				

16-round impossible differential and 22-round related-key impossible differential characteristics respectively. For impossible differential attack on 26-round Hight, we use a similar characteristic as in [19] which enables us to attack 26-round of Hight with a lower complexity. However, we use better characteristic for related-key impossible differential attack.

The process of both attacks is similar. First, a data collection part is processed for the generation of necessary plaintext-ciphertext pairs. Then, to guarantee the impossible differential characteristic, pairs are filtered by checking the conditions at each intermediate rounds. At the end, the guessed key is eliminated if any one of the remaining pairs satisfies the impossible differential characteristic.

4.1 Impossible Differential Attack on HIGHT-26

We use the following 16-round impossible differential which is also given in Table 5 in detail:

$$(e_{0,\sim}, 0, 0, 0, 0, 0, 0, 0) \nrightarrow (0, 80_x, 0, 0, 0, 0, 0, 0)$$

In Table 5, the contradictory differences and the guessed subkey bytes in the attack are labeled with gray background. The differences used here are considered with respect to XOR operation and shown as hexadecimal. The propagation of the differences can easily be checked by the properties of addition and linear subround functions F_i. Contradiction is shown by miss in the middle manner at values $X_{13,7}$. This attack covers the rounds 0 - 25 and excludes the input whitening as done in [19]. Overall attack on 26-round Hight works as follows:

1.) Data Collection

(i) Choose 2^{13} structures of 2^{48} plaintexts each where the bytes $(1,0)$ have fixed values, bytes $(7,6,5,4,3)$ and most significant 7 bits of the byte (2) take all possible values.
   - Such a structure of plaintexts propose 2^{94} plaintext pairs and so we get 2^{107} pairs in total.

(ii) Obtain all the ciphertexts C^i of the plaintexts P^i. Choose only the ciphertext pairs satisfying the difference $(?,?,?,?,e_{\bar{0},\sim},80_x,0,0)$.
   - This step can be done by inserting all the ciphertexts into a hash table indexed by expected inactive bits and choosing the colliding pairs which satisfy the required difference. There is a 25-bit filtering condition over the ciphertext pairs. Therefore, 2^{82} pairs remain.

2.) Filtering and Key Elimination

We have 28 similar steps given in Table 6 to reach impossible differential characteristic and eliminate the wrong key values.

Let us look at the first step as an example. Guess MK_3 and partially encrypt every plaintext pairs by using SK_3 to obtain $(7,0)$-th bytes of X_1 (The relation between the MK values and SK values are given in the Table 9). The expected difference for the $(7,0)$-th bytes is $(?,0)$ which comes up with an eight-bit condition. Therefore, the number of total pairs is decreased to $2^{82-8}=2^{74}$ after this step. In this step, we partially encrypt 2^{82} pairs with the guessed eight-bit of the secret key. Each partial encryption is equivalent to 1/4th of a round of Hight and the overall attack is done on 26 rounds. Thus, the complexity of this step is $2\cdot 2^{82}\cdot 2^{8}\cdot\frac{1}{4}\cdot\frac{1}{26}\approx 2^{84.30}$ 26-round Hight encryptions. Remaining steps given in Table 6 follow similarly. Moreover, if the secret key byte MK is already guessed and known for the required subkey SK, it is directly used and since most of the previous conditions are preserved for the next rounds there does not exist too much conditions on the evaluation process of intermediate rounds.

In step 28, if a pair satisfies the impossible differential characteristic, we eliminate that guessed key. Since there is an eight-bit condition, every pair eliminates 2^{-8} of the keys. Therefore after the first pair, there remain $2^{112}-2^{104}=2^{112}\cdot(1-2^{-8})$ keys. After the second pair, it is expected to have $2^{112}\cdot(1-2^{-8})-2^{112}\cdot(1-2^{-8})\cdot 2^{-8}=2^{112}\cdot(1-2^{-8})^2$ remaining keys. Following that manner, after the last pair, we have $2^{112}(1-2^{-8})^{2^{11}}\approx 2^{100.46}$ remaining keys. Complexity of this step is $2\cdot 2^{112}\left\{1+(1-2^{-8})+\ldots+(1-2^{-8})^{2^{11}-1}\right\}\cdot\frac{1}{4}\cdot\frac{1}{26}\approx 2^{114.30}$ Hight encryptions.

3.) Final Step

For every recorded 112 bit key at the end of Step 28, we obtain the remaining 16 bits and the original key itself with exhaustive search by checking over two plaintext-ciphertext pairs. The probability that a wrong key is suggested is approximately $2^{-64\times 2}=2^{-128}$. So, the expected number of wrong keys is about $2^{-128}\cdot 2^{116.46}=2^{-11.54}$. Thus, it is very likely that we can find the correct key.

Table 6. 26-Round impossible differential attack

	Guess Key Byte	Use	Obtain	Check Difference	Condition (In terms of bits)	Remaining Pairs	Time Complexity(HE)
1	MK_3	SK_3	$(7,0)$ of X_1	$(?,0)$	8	2^{74}	$\approx 2^{84.30}$
2	MK_1	WK_7, SK_{103}	$(7,6)$ of X_{25}	$(0,?)$	8	2^{66}	$\approx 2^{84.30}$
3	MK_2	SK_2	$(6,5)$ of X_1	-	-	2^{66}	$\approx 2^{84.30}$
4	MK_7	SK_7	$(7,0)$ of X_2	$(?,0)$	8	2^{58}	$\approx 2^{92.30}$
5	MK_0	WK_6, SK_{102}	$(5,4)$ of X_{25}	-	-	2^{58}	$\approx 2^{92.30}$
6	MK_4	SK_{98}	$(5,4)$ of X_{24}	$(0,?)$	8	2^{50}	$\approx 2^{100.30}$
7	-	WK_5, SK_{101}	$(3,2)$ of X_{25}	-	-	2^{50}	$\approx 2^{92.30}$
8	-	SK_{97}	$(3,2)$ of X_{24}	-	-	2^{50}	$\approx 2^{92.30}$
9	MK_8	SK_{93}	$(3,2)$ of X_{23}	$(0,?)$	8	2^{42}	$\approx 2^{100.30}$
10	-	SK_1	$(4,3)$ of X_1	-	-	2^{42}	$\approx 2^{92.30}$
11	MK_6	SK_6	$(6,5)$ of X_2	-	-	2^{42}	$\approx 2^{100.30}$
12	MK_{11}	SK_{11}	$(7,0)$ of X_3	$(?,0)$	8	2^{34}	$\approx 2^{108.30}$
13	-	WK_4, SK_{100}	$(1,0)$ of X_{25}	-	-	2^{34}	$\approx 2^{100.30}$
14	-	SK_{96}	$(1,0)$ of X_{24}	-	-	2^{34}	$\approx 2^{100.30}$
15	MK_{15}	SK_{92}	$(1,0)$ of X_{23}	-	-	2^{34}	$\approx 2^{108.30}$
16	-	SK_{88}	$(1,0)$ of X_{22}	$(0, e_{0,\sim})$	8	2^{26}	$\approx 2^{108.30}$
17	-	SK_0	$(2,1)$ of X_1	-	-	2^{26}	$\approx 2^{100.30}$
18	MK_5	SK_5	$(4,3)$ of X_2	-	-	2^{26}	$\approx 2^{108.30}$
19	MK_{10}	SK_{10}	$(6,5)$ of X_3	-	-	2^{26}	$\approx 2^{116.30}$
20	-	SK_{15}	$(7,0)$ of X_4	$(?,0)$	8	2^{18}	$\approx 2^{116.30}$
21	-	SK_{99}	$(7,6)$ of X_{24}	-	-	2^{18}	$\approx 2^{108.30}$
22	-	SK_{95}	$(7,6)$ of X_{23}	-	-	2^{18}	$\approx 2^{108.30}$
23	MK_{14}	SK_{91}	$(7,6)$ of X_{22}	-	-	2^{18}	$\approx 2^{116.30}$
24	-	SK_{87}	$(7,6)$ of X_{21}	$(0, 80_x)$	7	2^{11}	$\approx 2^{116.30}$
25	-	SK_4	$(2,1)$ of X_2	-	-	2^{11}	$\approx 2^{109.30}$
26	MK_9	SK_9	$(4,3)$ of X_3	-	-	2^{11}	$\approx 2^{117.30}$
27	-	SK_{14}	$(6,5)$ of X_4	-	-	2^{11}	$\approx 2^{117.30}$
28	-	SK_{19}	$(7,0)$ of X_5	$(e_{0,\sim}, 0)$	8	-	$\approx 2^{114.30}$

The total complexity of the steps given in Table 6 is $2^{119.35}$. Since we have approximately $2^{100.46}$ remaining keys before the final step, the complexity of the final step is $2^{100.46+16} = 2^{116.46}$. Therefore, the overall complexity of the attack is $2^{119.35} + 2^{116.46} = 2^{119.53}$ 26-round HIGHT evaluations, 2^{61} chosen plaintexts of data and 2^{109} bytes of memory.

4.2 Related-Key Impossible Differential Attack on HIGHT-31

In this section, we introduce our related-key impossible differential attack on 31-round HIGHT which utilizes a new 22-round related-key impossible differential. The differences of this attack from [19] are the used related-key impossible differential and the overall complexity which makes use of a related-key impossible differential of three more rounds. We use the following 22-round impossible differential which is given in Table 7 in detail:

$$(0,0,0,0,0,0,80_x,0) \nrightarrow (0,0,0,80_x,0,0,0,0)$$

The related-key impossible differential occurs by using the key difference $(\Delta MK_{15}, \Delta MK_{14}, \ldots, \Delta MK_0) = (80_x, 0, \ldots, 0)$. The contradiction occurs at values $X_{17,3}$ and can be shown similarly by miss in the middle manner which is given in Table 7 where the contradictory differences and the subkey bytes having nonzero differences are shown with gray background. The related-key impossible differential was found by imposing the difference 80_x to all 16 key bytes and observing the impossibility at the differentials. It can be concluded that the best

Table 7. 31-Round related-key impossible differential

	$\Delta X_{i,7}$	$\Delta X_{i,6}$	$\Delta X_{i,5}$	$\Delta X_{i,4}$	$\Delta X_{i,3}$	$\Delta X_{i,2}$	$\Delta X_{i,1}$	$\Delta X_{i,0}$	Subkeys			
ΔX_0	?	?	?	$e_{0,\sim}$	80_x	0	?	?	SK_3	SK_2	SK_1	SK_0
ΔX_1	?	?	$e_{0,\sim}$	80_x	0	0	?	?	SK_7	SK_6	SK_5	SK_4
ΔX_2	?	$e_{0,\sim}$	80_x	0	0	0	?	?	SK_{11}	SK_{10}	SK_9	SK_8
ΔX_3	$e_{0,\sim}$	80_x	0	0	0	0	?	?	SK_{15}	SK_{14}	SK_{13}	SK_{12}
ΔX_4	80_x	0	0	0	0	0	?	$e_{2,\sim}$	SK_{19}	SK_{18}	SK_{17}	SK_{16}
ΔX_5	0	0	0	0	0	0	$e_{2,\sim}$	80_x	SK_{23}	SK_{22}	SK_{21}	SK_{20}
ΔX_6	0	0	0	0	0	0	80_x	0	SK_{27}	SK_{26}	SK_{25}	SK_{24}
ΔX_7	0	0	0	0	0	0	0	0	SK_{31}	SK_{30}	SK_{29}	SK_{28}
ΔX_8	0	0	0	0	0	0	0	0	SK_{35}	SK_{34}	SK_{33}	SK_{32}
ΔX_9	0	0	0	0	0	0	0	0	SK_{39}	SK_{38}	SK_{37}	SK_{36}
ΔX_{10}	0	0	0	0	0	0	0	0	SK_{43}	SK_{42}	SK_{41}	SK_{40}
ΔX_{11}	0	0	0	80_x	0	0	0	0	SK_{47}	SK_{46}	SK_{45}	SK_{44}
ΔX_{12}	0	$e_{2,\sim}$	80_x	0	0	0	0	0	SK_{51}	SK_{50}	SK_{49}	SK_{48}
ΔX_{13}	$e_{2,\sim}$	80_x	0	0	0	0	0	?	SK_{55}	SK_{54}	SK_{53}	SK_{52}
ΔX_{14}	80_x	0	0	0	0	?	?	$e_{0,\sim}$	SK_{59}	SK_{58}	SK_{57}	SK_{56}
ΔX_{15}	0	80_x	0	?	?	?	$e_{0,\sim}$	80_x	SK_{63}	SK_{62}	SK_{61}	SK_{60}
ΔX_{16}	80_x	?	?	?	?	$e_{0,\sim}$	80_x	$e_{0,\sim}$	SK_{67}	SK_{66}	SK_{65}	SK_{64}
ΔX_{17}	?	?	?	?	$e_{0,\sim}$	?	$e_{0,\sim}$	?	SK_{71}	SK_{70}	SK_{69}	SK_{68}
ΔX_{17}	?	?	?	$e_{0,\sim}$	80_x	0	?	?	SK_{71}	SK_{70}	SK_{69}	SK_{68}
ΔX_{18}	?	?	$e_{0,\sim}$	80_x	0	0	?	?	SK_{75}	SK_{74}	SK_{73}	SK_{72}
ΔX_{19}	?	$e_{0,\sim}$	80_x	0	0	0	?	?	SK_{79}	SK_{78}	SK_{77}	SK_{76}
ΔX_{20}	$e_{0,\sim}$	80_x	0	0	0	0	?	?	SK_{83}	SK_{82}	SK_{81}	SK_{80}
ΔX_{21}	80_x	0	0	0	0	0	?	$e_{2,\sim}$	SK_{87}	SK_{86}	SK_{85}	SK_{84}
ΔX_{22}	0	0	0	0	0	0	$e_{2,\sim}$	80_x	SK_{91}	SK_{90}	SK_{89}	SK_{88}
ΔX_{23}	0	0	0	0	0	0	80_x	0	SK_{95}	SK_{94}	SK_{93}	SK_{92}
ΔX_{24}	0	0	0	0	0	0	0	0	SK_{99}	SK_{98}	SK_{97}	SK_{96}
ΔX_{25}	0	0	0	0	0	0	0	0	SK_{103}	SK_{102}	SK_{101}	SK_{100}
ΔX_{26}	0	0	0	0	0	0	0	0	SK_{107}	SK_{106}	SK_{105}	SK_{104}
ΔX_{27}	0	0	0	0	0	0	0	0	SK_{111}	SK_{110}	SK_{109}	SK_{108}
ΔX_{28}	0	0	0	80_x	0	0	0	0	SK_{115}	SK_{114}	SK_{113}	SK_{112}
ΔX_{29}	0	$e_{2,\sim}$	80_x	0	0	0	0	0	SK_{119}	SK_{118}	SK_{117}	SK_{116}
ΔX_{30}	$e_{2,\sim}$	80_x	0	0	0	0	0	?	SK_{123}	SK_{122}	SK_{121}	SK_{120}
ΔX_{31}	80_x	0	0	0	0	?	?	$e_{0,\sim}$	WK_7	WK_6	WK_5	WK_4
FT	$e_{0,\sim}$	80_x	0	0	0	0	?	?				

related-key impossible differential is 22 rounds and it can not be extended by the same technique[5].

Using this related-key impossible differential, we can attack up to 31 rounds of Hight. This attack covers the rounds 0 - 30 and excludes the input whitening as done in 26-round impossible differential attack [19]. We use the related-key impossible differential characteristic to attack 31-round of Hight which is detailed in Table 7. The attack can be described as follows.

1.) Data Collection

(i) Choose 2^{15} structures of 2^{48} plaintexts each where the byte (2) and the least significant seven bits of the byte (3) are fixed to certain values. The bytes $(7, 6, 5, 1, 0)$ and the most significant seven bits of the byte (7) contain every possible values.
   - There exist 2^{110} plaintext pairs in total which are encrypted by the prescribed difference in the key.

(ii) Obtain all the ciphertexts C^i of the plaintexts P^i encrypted with K^1 and ciphertext $C^{i'}$ of the plaintexts P^i encrypted with K^2 where $K^1 \oplus K^2 = (80_x, 0, \ldots, 0)$. Choose only the ciphertext pairs $(C^i, C^{j'})$ satisfying the difference $(e_{0,\sim}, 80_x, 0, 0, 0, 0, ?, ?)$.

[5] A similar attack with the same complexity can be mounted by imposing the difference 80_x to MK_9 instead of MK_{15}.

Table 8. 31-Round related key impossible differential attack

	Guess Key Byte	Use	Obtain	Check Difference	Condition (In terms of bits)	Remaining Pairs	Time Complexity(HE)
1	MK_0	SK_0	$(2,1)$ of X_1	$(0,?)$	8 bits	2^{61}	$\approx 2^{71.05}$
2	MK_3	SK_3	$(7,0)$ of X_1	-	-	2^{61}	$\approx 2^{71.05}$
3	MK_4	SK_4	$(2,1)$ of X_2	$(0,?)$	8 bits	2^{53}	$\approx 2^{79.05}$
4	MK_9	WK_4, SK_{120}	$(1,0)$ of X_{30}	$(0,?)$	8 bits	2^{45}	$\approx 2^{79.05}$
5	MK_{12}	WK_7, SK_{123}	$(7,6)$ of X_{30}	$(e_{2,\sim}, 80_x)$	2 bits	2^{43}	$\approx 2^{79.05}$
6	-	SK_{119}	$(7,6)$ of X_{29}	$(0, e_{2,\sim})$	8 bits	2^{35}	$\approx 2^{77.05}$
7	MK_2	SK_2	$(6,5)$ of X_1	-	-	2^{35}	$\approx 2^{77.05}$
8	MK_7	SK_7	$(7,0)$ of X_2	-	-	2^{35}	$\approx 2^{85.05}$
9	MK_8	SK_8	$(2,1)$ of X_3	$(0,?)$	8 bits	2^{27}	$\approx 2^{93.05}$
10	MK_{11}	WK_6, SK_{122}	$(5,4)$ of X_{30}	-	-	2^{27}	$\approx 2^{93.05}$
11	-	SK_{118}	$(5,4)$ of X_{29}	-	-	2^{27}	$\approx 2^{93.05}$
12	-	SK_{114}	$(5,4)$ of X_{28}	$(0, 80_x)$	5 bits	2^{22}	$\approx 2^{93.05}$
13	MK_1	SK_1	$(4,3)$ of X_1	-	-	2^{22}	$\approx 2^{96.05}$
14	MK_6	SK_6	$(6,5)$ of X_2	-	-	2^{22}	$\approx 2^{104.05}$
15	-	SK_{11}	$(7,0)$ of X_3	-	-	2^{22}	$\approx 2^{104.05}$
16	-	SK_{12}	$(2,1)$ of X_4	$(0,?)$	8 bits	2^{14}	$\approx 2^{104.05}$
17	MK_5	SK_5	$(4,3)$ of X_2	-	-	2^{14}	$\approx 2^{104.05}$
18	MK_{10}	SK_{10}	$(6,5)$ of X_3	-	-	2^{14}	$\approx 2^{112.05}$
19	MK_{15}	SK_{15}	$(7,0)$ of X_4	$(80_x, e_{2,\sim})$	2 bits	2^{12}	$\approx 2^{120.05}$
20	-	SK_{16}	$(2,1)$ of X_5	$(0, e_{2,\sim})$	8 bits	2^4	$\approx 2^{118.05}$
21	-	SK_9	$(4,3)$ of X_3	-	-	2^4	$\approx 2^{110.05}$
22	MK_{14}	SK_{14}	$(6,5)$ of X_4	-	-	2^4	$\approx 2^{118.05}$
23	-	SK_{19}	$(7,0)$ of X_5	-	-	2^4	$\approx 2^{118.05}$
24	-	SK_{20}	$(2,1)$ of X_6	$(0, 80_x)$	5 bits	-	$\approx 2^{117.89}$

– This step can be done by inserting all the ciphertexts into a hash table indexed by expected inactive bits and choosing the colliding pairs which satisfy the required difference. There is 41-bit filtering condition over the ciphertext pairs. Therefore 2^{69} pairs remain.

2.) Filtering and Key Elimination

We follow the steps as in 26-round attack which is given in Table 8 to reach impossible differential characteristic.

In step 24, if a pair satisfies the impossible differential characteristic, we eliminate the corresponding guessed key. Since there is five-bit condition, each pair eliminates 2^{-5} of the keys and after the last pair there remain $2^{120}(1-2^{-5})^{2^4} \approx 2^{119.27}$ keys.

3.) Final Step:

For every recorded 120 bit key $(MK_0, \ldots, MK_{12}, MK_{14}, MK_{15})$, obtain the remaining eight bits of the key by exhaustive search by checking over three plaintext-ciphertext pairs. The probability that a wrong key is suggested is approximately $2^{-64\times 3} = 2^{-192}$. So, the expected number of wrong keys is about $2^{-192} \cdot 2^{127.27} = 2^{-64.73}$. It is very likely that we can find the correct key.

The total complexity of the steps given in Table 8 is approximately $2^{121.03}$. Since there exists $2^{119.27}$ remaining keys, the complexity of the final step is approximately $2^{127.27}$. Therefore, the complexity of this whole attack is $2^{127.28}$ 31-round Hight computations. Even if this attack is not significant compared to the exhaustive search, it is still an important result against Hight which reduces the safety margin of Hight to one round.

5 Conclusion

In this paper, we present the first related-key cryptanalysis of Present and improve the recent impossible differential attacks on Hight. Although our attacks are totally theoretical and have no practical implications, they show new results for both ciphers. The related-key attacks on Present can be seen as a new approach to see the security of the cipher. Hight, on the other hand, was shown to be secure up to 19 rounds in the original proposal and recently attacked up to 28 rounds. However, our results show that the security of Hight can be further reduced up to one round.

Acknowledgements

The authors would like to thank anonymous reviewers of ACISP 2009 for pointing out recent results on Present and Jonsung Kim for the additional information about Hight. We also like to thank Meltem Sönmez Turan, Shahram Khazaei and Martijn Stam for reviewing the previous versions of the paper. The simulations were run on the HPC Cluster of the Department of Computer Engineering, Middle East Technical University.

References

1. Daemen, J., Rijmen, V.: The Design of Rijndael. Springer, New York (2002)
2. 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, Heidelberg (2006)
3. South Korea Telecommunications Technology Associations (TTA). 64-bit Block Cipher HIGHT. Standardization Number TTAS.KO-12.0040, December 27 (2006)
4. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
5. Lim, C.H., Korkishko, T.: mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In: Song, J.-S., Kwon, T., Yung, M. (eds.) WISA 2005. LNCS, vol. 3786, pp. 243–258. Springer, Heidelberg (2006)
6. Standaert, F.-X., Piret, G., Gershenfeld, N., Quisquater, J.-J.: SEA: A Scalable Encryption Algorithm for Small Embedded Applications. In: Domingo-Ferrer, J., Posegga, J., Schreckling, D. (eds.) CARDIS 2006. LNCS, vol. 3928, pp. 222–236. Springer, Heidelberg (2006)
7. Robshaw, M.J.B.: Searching for Compact Algorithms: CGEN. In: Nguyên, P.Q. (ed.) VIETCRYPT 2006. LNCS, vol. 4341, pp. 37–49. Springer, Heidelberg (2006)
8. Leander, G., Paar, C., Poschmann, A., Schramm, K.: New Lightweight DES Variants. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 196–210. Springer, Heidelberg (2007)
9. Wheeler, D.J., Needham, R.M.: TEA, a Tiny Encryption Algorithm. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 363–366. Springer, Heidelberg (1995)

10. Wheeler, D.J., Needham, R.M.: TEA Extensions (October 1997)
11. The eSTREAM Portfolio. The eSTREAM Project (September 2008), http://www.ecrypt.eu.org/stream/
12. Wang, M.: Differential Cryptanalysis of Reduced-Round PRESENT. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 40–49. Springer, Heidelberg (2008)
13. Albrecht, M., Cid, C.: Algebraic Techniques in Differential Cryptanalysis. To appear in proceedings of FSE (2009)
14. Z'aba, M.R., Raddum, H., Henricksen, M., Dawson, E.: Bit-Pattern Based Integral Attack. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 363–381. Springer, Heidelberg (2008)
15. Collard, B., Standaert, F.-X.: A Statistical Saturation Attack against the Block Cipher PRESENT. To appear in proceedings of CT-RSA (2009)
16. Biryukov, A., Wagner, D.: Slide Attacks. In: Knudsen [26], pp. 245–259
17. Biham, E.: New Types of Cryptanalytic Attacks Using Related Keys. Journal of Cryptology 7(4), 229–246 (1994)
18. Lu, J.: Cryptanalysis of Block Ciphers. PhD thesis, Royal Holloway, University of London, England (July 2008)
19. 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, Heidelberg (2007)
20. Biham, E., Dunkelman, O., Keller, N.: Related-Key Boomerang and Rectangle Attacks. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 507–525. Springer, Heidelberg (2005)
21. Biham, E., Dunkelman, O., Keller, N.: New Combined Attacks on Block Ciphers. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 126–144. Springer, Heidelberg (2005)
22. Biham, E., Dunkelman, O., Keller, N.: The Rectangle Attack - Rectangling the Serpent. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 340–357. Springer, Heidelberg (2001)
23. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. Journal of Cryptology 18(4), 291–311 (2005)
24. Biham, E., Biryukov, A., Shamir, A.: Miss in the Middle Attacks on IDEA and Khufu. In: Knudsen [26], pp. 124–138
25. Dunkelman, O.: Techniques for Cryptanalysis of Block Ciphers. PhD thesis, Technion, Israel (February 2006)
26. Knudsen, L.R. (ed.): FSE 1999. LNCS, vol. 1636. Springer, Heidelberg (1999)

A A Brief Description of Related-Key Rectangle Attack

Let E denote the encryption function of the attacked cipher which is treated as a cascade of four subciphers as $E = E_f \circ E_1 \circ E_0 \circ E_b$ where the core $E' = E_1 \circ E_0$ is covered by additional rounds called E_b and E_f which are the subciphers before and after the core function respectively to be used in key recovery. The related-key differential is the quadruple $(\Delta X, \Delta Y, \Delta K, p)$ satisfying following under the nonlinear K-keyed function F_K,

$$Pr[F_K(P) \oplus F_{K \oplus \Delta K}(P \oplus \Delta X) = \Delta Y] = p.$$

Here, ΔX and ΔY are the corresponding input and output differences respectively, ΔK the key difference and p is the corresponding probability. We say the related-key differential characteristics $\Delta X \overset{\Delta K}{\to} \Delta Y$ holds with probability $Pr = p$.

Let $\alpha \xrightarrow{\Delta K^{12}} \beta$ with probability p be the first related-key differential used for E_0 and $\gamma \xrightarrow{\Delta K^{13}} \delta$ with probability q be the second differential used for E_1. Here, (α, β) and (γ, δ) stand for the input-output differences for E_0 and E_1 respectively. We define $\hat{p}$ and $\hat{q}$ as the probabilities related to α and δ respectively as follows: $\hat{p} = \sqrt{\sum_\beta P^2_{\Delta K^{12}}(\alpha \to \beta)}$ and $\hat{q} = \sqrt{\sum_\gamma P^2_{\Delta K^{13}}(\gamma \to \delta)}$. Here, β and γ are the possible differences at the end of E_0 and at the beginning of E_1. The key differences $\Delta K^{12} = K^1 \oplus K^2 = K^3 \oplus K^4$ and $\Delta K^{13} = K^1 \oplus K^3 = K^2 \oplus K^4$ are the respective key differences for the subciphers E_0 and E_1. The subciphers before and after the core function are formed according to the α and δ differences. The basic related-key rectangle attack for the core function works as follows:

- Take a randomly chosen plaintext P_1 and form $P_2 = P_1 \oplus \alpha$.
- Obtain the corresponding ciphertexts $C_1 = E'_{K^1}(P_1)$ and $C_2 = E'_{K^2}(P_2)$, where $K^2 = K^1 \oplus \Delta K^{12}$.
- Pick another randomly chosen plaintext P_3 and form $P_4 = P_3 \oplus \alpha$.
- Obtain the corresponding ciphertexts $C_3 = E'_{K^3}(P_3)$ and $C_4 = E'_{K^4}(P_4)$, where $K^3 = K^1 \oplus \Delta K^{13}$ and $K^4 = K^3 \oplus \Delta K^{12}$.
- Check $C_1 \oplus C_3 = C_2 \oplus C_4 = \delta$.

The probability of the rectangle distinguisher is given by $Pr = 2^{-n}\hat{p}^2\hat{q}^2$ where n is the block size. If $\hat{p} \cdot \hat{q}$ is sufficiently high, the rectangle distinguisher works. As shown in [25], if the expected number of right quartets is taken to be four, then there is at least one right quartet in the data set with probability 0.982 since it is a Poisson distribution with expectation of four. Therefore, for this success rate, the number of plaintext pairs needed is $N = 2^{n/2+1}/\hat{p}\hat{q}$ that consist of $2^{n+2}/\hat{p}^2\hat{q}^2$ quartets expecting four right quartets at a time. Further details about rectangle and boomerang attacks can be found in [20,21,22,25].

B Key Schedule Properties of HIGHT

Table 9. Relations of the original key with whitening keys and subkeys

Original Key	Whitening Keys	Subkeys							
MK_{15}	WK_3	SK_{15}	SK_{24}	SK_{41}	SK_{58}	SK_{75}	SK_{92}	SK_{109}	SK_{126}
MK_{14}	WK_2	SK_{14}	SK_{31}	SK_{40}	SK_{57}	SK_{74}	SK_{91}	SK_{108}	SK_{125}
MK_{13}	WK_1	SK_{13}	SK_{30}	SK_{47}	SK_{56}	SK_{73}	SK_{90}	SK_{107}	SK_{124}
MK_{12}	WK_0	SK_{12}	SK_{29}	SK_{46}	SK_{63}	SK_{72}	SK_{89}	SK_{106}	SK_{123}
MK_{11}	WK_{15}	SK_{11}	SK_{28}	SK_{45}	SK_{62}	SK_{79}	SK_{88}	SK_{105}	SK_{122}
MK_{10}	WK_{14}	SK_{10}	SK_{27}	SK_{44}	SK_{61}	SK_{78}	SK_{95}	SK_{104}	SK_{121}
MK_9	WK_{13}	SK_9	SK_{26}	SK_{43}	SK_{60}	SK_{77}	SK_{94}	SK_{111}	SK_{120}
MK_8	WK_{12}	SK_8	SK_{25}	SK_{42}	SK_{59}	SK_{76}	SK_{93}	SK_{110}	SK_{127}
MK_7	WK_{11}	SK_7	SK_{16}	SK_{33}	SK_{50}	SK_{67}	SK_{84}	SK_{101}	SK_{118}
MK_6	WK_{10}	SK_6	SK_{23}	SK_{32}	SK_{49}	SK_{66}	SK_{83}	SK_{100}	SK_{117}
MK_5	WK_9	SK_5	SK_{22}	SK_{39}	SK_{48}	SK_{65}	SK_{82}	SK_{99}	SK_{116}
MK_4	WK_8	SK_4	SK_{21}	SK_{38}	SK_{55}	SK_{64}	SK_{81}	SK_{98}	SK_{115}
MK_3	WK_7	SK_3	SK_{20}	SK_{37}	SK_{54}	SK_{71}	SK_{80}	SK_{97}	SK_{114}
MK_2	WK_6	SK_2	SK_{19}	SK_{36}	SK_{53}	SK_{70}	SK_{87}	SK_{96}	SK_{113}
MK_1	WK_5	SK_1	SK_{18}	SK_{35}	SK_{52}	SK_{69}	SK_{86}	SK_{103}	SK_{112}
MK_0	WK_4	SK_0	SK_{17}	SK_{34}	SK_{51}	SK_{68}	SK_{85}	SK_{102}	SK_{119}