Boomerang and Slide-Rotational Analysis of the SM3 Hash Function

Aleksandar Kircanski[1], Yanzhao Shen[2], Gaoli Wang[2,3,⋆], and Amr M. Youssef[1]

[1] Concordia University
Concordia Institute for Information Systems Engineering
Montreal, Quebec, Canada
[2] Donghua University
School of Computer Science and Technology, Shanghai, China
[3] State Key Laboratory of Information Security
Institute of Software, Chinese Academy of Sciences, Beijing, China
wanggaoli@dhu.edu.cn

Abstract. SM3 is a hash function, designed by Xiaoyun Wang *et al.* and published by the Chinese Commercial Cryptography Administration Office for the use of electronic authentication service system. The design of SM3 builds upon the design of the SHA-2 hash function, but introduces additional strengthening features. In this paper, we present boomerang distinguishers for the SM3 compression function reduced to 32 steps out of 64 steps with complexity $2^{14.4}$, 33 steps with complexity $2^{32.4}$, 34 steps with complexity $2^{53.1}$ and 35 steps with complexity $2^{117.1}$. Examples of zero-sum quartets for the 32-step and 33-step SM3 compression function are provided. We also point out a slide-rotational property of SM3-XOR, which exists due to the fact that constants used in the steps are not independent.

Keywords: Cryptanalysis, Boomerang attack, Rotational attack, Slide attack, SM3.

1 Introduction

In December of 2007, the Chinese National Cryptographic Administration Bureau released the specification of a Trusted Cryptography Module detailing a cryptoprocessor to be used within the Trusted Computing framework in China. The module specifies a set of cryptographic algorithms that includes the SMS4 block cipher, the SM2 asymmetric algorithm and SM3, a new cryptographic hash function designed by Xiaoyun Wang *et al.* [1]. The design of SM3 resembles the design of SHA-2 but includes additional fortifying features such as feeding two message-derived words into each step, as opposed to only one in the case of SHA-2.

The only previous work that we are aware of on the analysis of SM3 has been presented by Zou *et al.* [2] at ICISC 2011 where a preimage attack on step-reduced SM3 is provided. In particular, Zou *et al.* presented attacks on SM3 reduced to 30 steps, starting from the 7-th step, with time complexity 2^{249} and 28 steps, starting from the 1-st step with time complexity $2^{241.5}$.

[⋆] Corresponding author.

L.R. Knudsen and H. Wu (Eds.): SAC 2012, LNCS 7707, pp. 304–320, 2013.

The use of zero-sums as distinguishers have been introduced by Aumasson *et al.* [3]. The boomerang attack [4], originally introduced for block ciphers, has been adapted to the hash function setting independently by Biryukov *et al.* [5], and Lamberger and Mendel [6]. In particular, in [5], a distinguisher for the 7-round BLAKE-32 was provided, whereas in [6] a distinguisher for the 46-step SHA-2 compression function was provided. The latter SHA-2 result was extended to 47 steps in [7]. In [8], Mendel and Nad presented boomerang distinguishers for the SIMD-512 compression function. Sasaki [9] gave a boomerang distinguisher on the full compression function of 5-pass HAVAL. Sasaki also proposed a 2-dimension sums attack on 48-step RIPEMD-128 and 42-step RIPEMD-160 in [10]. Boomerang distinguishers have also been applied to Skein and Threefish. In [11] Aumasson *et al.* proposed a related-key boomerang distinguisher on 34-step Skein and a known-related-key boomerang distinguisher on 35-step Skein. In [12], Leurent and Roy showed that, under some conditions, three independent paths instead of two can be combined to achieve a distinguisher for the compression function with complexity 2^{114}. In [13] Chen *et al.* proposed related-key boomerang distinguishers on 32-step, 33-step and 34-step Threefish-512. Recently, Yu *et al.* [14] proposed boomerang attacks on the 32-step, 33-step and 34-step Skein-512.

Khovratovich *et al.* introduced rotational distinguishers in [15], where two words are said to be rotational if they are equal up to bit-wise rotation by some number of positions. Slide attacks were introduced by Biryukov *et al.* [16] and subsequently were applied to many cryptographic primitives.

Our Contribution. In the first part of this paper, we present a boomerang attack on the SM3 hash function reduced to 32 steps out of 64 steps with complexity $2^{14.4}$, 33 steps with complexity $2^{32.4}$, 34 steps with complexity $2^{53.1}$ and 35 steps with complexity $2^{117.1}$. Particular examples of the boomerang distinguisher for the 32-step and also the 33-step compression function are provided. The previous results and a summary of ours are given in Table 1.

In the second part of the paper, we present a slide-rotational property of SM3 and we analyze the SM3-XOR compression function, which is the SM3 compression function with the addition mod 2^{32} replaced by XOR. In particular, we show that, for SM3-XOR, one can easily construct input-output pairs satisfying a simple rotational property. Such a property exists due to the fact that, unlike in SHA-2, the constants in steps i, $i + 1$, for $i = 0, \ldots, 63$, $i \neq 15$ are computed by bitwise rotation starting from two predefined independent values. Previously, SHA2-XOR was analyzed in [17].

Paper Outline. The rest of the paper is organized as follows. In Section 2, we briefly review the specifications of the SM3 hash function and give the notation used in this paper. A brief overview of boomerang attacks is provided in Section 3. The differential characteristics, and a description of the boomerang attack process and its complexity evaluation are provided in Section 4. The slide-rotational property is explained in Section 5. Finally, our conclusion is given in Section 6.

Table 1. Summary of the attacks on the SM3 compression function (CF) and hash function (HF)

Attack	CF/HF	Steps	Complexity	Reference
Preimage attack	HF	28	$2^{241.5}$	[2]
Preimage attack	HF	30	2^{249}	[2]
Boomerang attack	CF	32	$2^{14.4}$	Section 4
Boomerang attack	CF	33	$2^{32.4}$	Section 4
Boomerang attack	CF	34	$2^{53.1}$	Section 4
Boomerang attack	CF	35	$2^{117.1}$	Section 4

2 Description of SM3 and Notation

In this section, we briefly review relevant specifications of the SM3 hash function and provide the notation used throughout the paper.

2.1 Description of SM3

The SM3 hash function compresses messages of arbitrary length into 256-bit hash values. Given any message, the algorithm first pads it into a message of length that is a multiple of 512 bits. We omit the padding method here since it is irrelevant to our attack. For our purpose, SM3 consists mainly of two parts: the message expansion and the state update transformation. In here, we briefly review the relevant specifications of these two components. For a detailed description of the hash function, we refer the reader to [1].

Message Expansion. The message expansion of SM3 splits the 512-bit message block M into 16 words m_i, $(0 \le i \le 15)$, and expands them into 68 expanded message words w_i $(0 \le i \le 67)$ and 64 expanded message words $w_i{}'(0 \le i \le 63)$ as follows:

$$w_i = \begin{cases} m_i, & 0 \le i \le 15, \\ P_1(w_{i-16} \oplus w_{i-9} \oplus (w_{i-3} \lll 15)) \oplus (w_{i-13} \lll 7) \oplus w_{i-6}, & 16 \le i \le 67, \end{cases}$$

$$w_i{}' = w_i \oplus w_{i+4}, 0 \le i \le 63.$$

The functions $P_0(X)$ which is used in the state update transformation and $P_1(X)$ which is used in message expansion are given by

$$P_0(X) = X \oplus (X \lll 9) \oplus (X \lll 17),$$
$$P_1(X) = X \oplus (X \lll 15) \oplus (X \lll 23).$$

State Update Transformation. The state update transformation starts from an initial value $IV = (A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0)$ of eight 32-bit words and updates them in 64 steps. In step $i + 1(0 \le i \le 63)$ the 32-bit words w_i and $w_i{}'$ are used to update the state variables $A_i, B_i, C_i, D_i, E_i, F_i, G_i, H_i$ as follows:

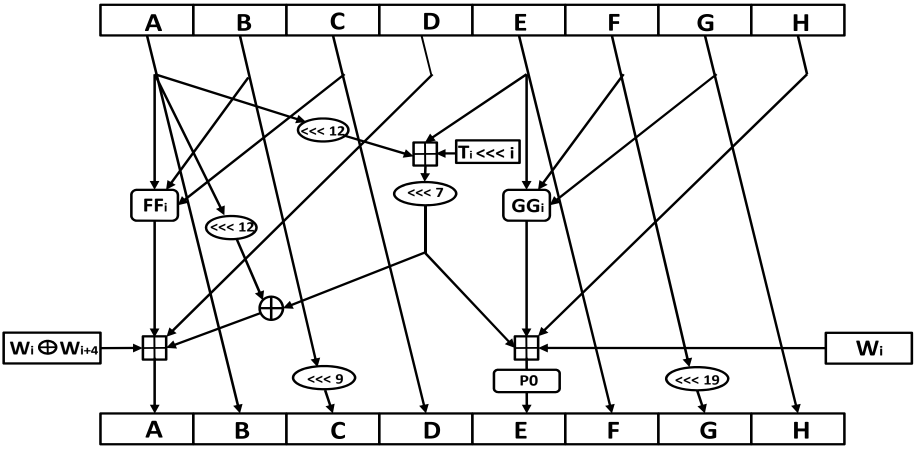

Fig. 1. One step of the SM3 hash function

$$
\begin{aligned}
SS1_i &= ((A_i \lll 12) + E_i + (T_i \lll i)) \lll 7,\\
SS2_i &= SS1_i \oplus (A_i \lll 12),\\
TT1_i &= FF_i(A_i, B_i, C_i) + D_i + SS2_i + w_i{}',\\
TT2_i &= GG_i(E_i, F_i, G_i) + H_i + SS1_i + w_i,\\
A_{i+1} &= TT1_i, B_{i+1} = A_i, C_{i+1} = (B_i \lll 9), D_{i+1} = C_i,\\
E_{i+1} &= P_0(TT2_i), F_{i+1} = E_i, G_{i+1} = (F_i \lll 19), H_{i+1} = G_i.
\end{aligned}
$$

The round constants are $T_i = 0x79cc4519$ for $i \in \{0, ..., 15\}$ and $T_i = 0x7a879d8a$, for $i \in \{16, ..., 63\}$. As for the bitwise Boolean functions $FF(X, Y, Z)$ and $GG(X, Y, Z)$ used in each step, we have

$$
FF(X, Y, Z) = \begin{cases} X \oplus Y \oplus Z, & 0 \le i \le 15,\\ (X \wedge Y) \vee (X \wedge Z) \vee (Y \wedge Z), & 16 \le i \le 63, \end{cases}
$$

$$
GG(X, Y, Z) = \begin{cases} X \oplus Y \oplus Z, & 0 \le i \le 15,\\ (X \wedge Y) \vee (\neg X \wedge Z), & 16 \le i \le 63. \end{cases}
$$

If M is the last block, then $(A_{64} \oplus A_0, B_{64} \oplus B_0, C_{64} \oplus C_0, D_{64} \oplus D_0, E_{64} \oplus D_0, F_{64} \oplus F_0, G_{64} \oplus G_0, H_{64} \oplus H_0)$ is the hash value. Otherwise $(A_{64} \oplus A_0, B_{64} \oplus B_0, C_{64} \oplus C_0, D_{64} \oplus D_0, E_{64} \oplus D_0, F_{64} \oplus F_0, G_{64} \oplus G_0, H_{64} \oplus H_0)$ constitutes the input of the next message block. One step of the SM3 compression function is depicted in Fig. 1.

2.2 Notation

Our attacks use the integer modular subtraction difference. In here, we introduce the notation used in throughout the rest of the paper.

1. X, Y, X' and Y' represent four 256-bit middle chaining values.
2. $IV_X, IV_Y, IV_{X'}$ and $IV_{Y'}$ represent four 256-bit initial values.
3. $H_X, H_Y, H_{X'}$ and $H_{Y'}$ represent four 256-bit outputs of the compression function.
4. $M_X, M_Y, M_{X'}$ and $M_{Y'}$ represent four 512-bit message blocks.
5. $w_{i,j}$ denotes the j-th bit of w_i, $0 \leq j \leq 31$.
6. $(W_t)_i^j$ (t can be X, Y, X' or Y') denotes $(j - i + 1)$ 32-bit words, w_i to w_j, where $i < j$.
7. $S_i[+j] = S_i + 2^j$ with no bit carry, $S_i[-j] = S_i - 2^j$ with no bit carry, where S can be $w, A, B, C, D, E, F, G, H$.
8. $S_i^{M_c}$ denotes the chaining value used in step i combined with the middle chaining value M_c where M_c can be X, Y, X', Y', and S can be $w, w', A, B, C, D, E, F, G, H, SS1$, or $SS2$.

3 Boomerang Distinguishers for Hash Functions

In this section, we review known-related-key boomerang attacks which can be used to distinguish a given permutation from a random oracle. We concentrate on the known-related-key boomerang attack to the compression function in the Davies-Mayer mode, i.e., $CF(M, K) = E(M, K) \oplus M$. As noted in [5,7,9,14], we can start from middle steps to construct boomerang distinguishers. Then we have

$$CF_0^{-1}(X, K_1) \oplus CF_0^{-1}(X \oplus \beta, K_2) = \alpha, \tag{1}$$

and

$$CF_1(X, K_1) \oplus CF_1(X \oplus \gamma, K_3) = \delta, \tag{2}$$

where the differential in CF_0^{-1} holds with probability p_0 and holds with probability p_1 in CF_1. Using these two differentials, we can construct the boomerang attack for the compression function CF as follows:

1. Choose a random value X, compute the corresponding value $X' = X \oplus \beta, Y = X \oplus \gamma, Y' = Y \oplus \beta$ and $K_2 = K_1 \oplus \beta_k, K_3 = K_1 \oplus \gamma_k, K_4 = K_3 \oplus \beta_k$.
2. Compute backward from $(X, K_1), (X', K_2), (Y, K_3), (Y', K_4)$ using CF_0^{-1} to obtain P, P', Q, Q'.
3. Compute forward from $(X, K_1), (X', K_2), (Y, K_3), (Y', K_4)$ using CF_1 to obtain C, C', D, D'.
4. Check whether $P \oplus P' = Q \oplus Q' = \alpha$ and $C \oplus D = C' \oplus D' = \delta$.

From (1) and (2),

$$P \oplus P' = Q \oplus Q' = \alpha \; and \; C \oplus D = C' \oplus D' = \delta, \tag{3}$$

holds with probability at least p_0^2 in the backward direction and with probability at least p_1^2 in the forward direction. Hence, assuming that the differentials are independent, the attack succeeds with probability $p_0^2 p_1^2$. The expected number of solutions to (3) is 1, if we repeat the attack about $1/(p_0^2 p_1^2)$ times.

For an n-bit random permutation, there exist 3 types of boomerang distinguishers which are summarized by Yu *et al.* in [14]. Here we recall the three distinguishers as follows.

- Type I: A quartet that satisfies $P \oplus P' = Q \oplus Q' = \alpha$, and $C \oplus D = C' \oplus D' = \delta$ where the differences α and δ are fixed. In this case, there exists a generic attack with complexity 2^n.
- Type II: A quartet that only satisfies the condition $C \oplus D = C' \oplus D'$. This type of attack is called second-order differential collision attack or zero-sum attack. In this case, we can use Wagner's generalized birthday attack [19] to obtain a quartet with the complexity $2^{n/3}$.
- Type III: A quartet satisfies the conditions $P \oplus P' = Q \oplus Q'$ and $C \oplus D = C' \oplus D'$. The complexity of this attack is about $2^{n/2}$.

In this paper, we apply a type III attack to develop distinguishers for 32/33/34/35 steps of the SM3 compression function. Therefore, the attack is valid if $p_0^2 \cdot p_1^2 > 2^{-n/2}$.

4 Attacks on the SM3 Compression Function

In this section, we describe the proposed boomerang attack on the SM3 compression function reduced to 32 steps, and then expend our attack to 33, 34 and 35 steps. Firstly, we give a summary of the differential characteristics to be used to distinguish the target compression function from random functions. Secondly, we describe how to use the message modification technique to correct the conditions in the intermediate steps by modifying the chaining values A_{16} to H_{16}. We express our attack algorithm on 32-step SM3 compression function in the third part of this section. Then we evaluate the complexity of our attack and extend it to 33, 34 and 35 steps.

4.1 Differential Characteristics

In here, we mainly describe the differential characteristics which are used to attack 32-step SM3 compression function. In Table 2, we present a differential characteristic in the backward direction from step 16 to step 1 which holds with probability 2^{-67}, and the sufficient conditions that ensure that this characteristic holds. A differential characteristic in the forward direction from step 17 to step 32 which holds with probability 2^{-34} and its associated sufficient conditions are presented in Table 3.

Finding the differential characteristics for both backward and forward directions is an important part of the attack. We construct the differential characteristics as follows.

- The characteristic has a single bit difference in the message word w_i at some step, i, followed by 15 message words without differences. When using such characteristic, 12 steps (the ones that follow i) can be bypassed with probability 1. Because of the fast diffusion of the difference coming from the message words, any characteristic that does not follow this strategy will have a low probability.
- In the backward direction, the differences in the message words are chosen as follows: $\Delta w_2 = [+31]$, $\Delta w_i = 0$ $(0 \leq i \leq 15, i \neq 2)$. Because Δw_2= [+31] and $w_2' = w_2 \oplus w_6$, we choose $w_{6,31} = 0$ to ensure that $\Delta w_2'$= [+31]. Since $w_{18} = P_1(w_2 \oplus w_9 \oplus (w_{15} \lll 15)) \oplus (w_5 \lll 7) \oplus w_{12}$, by choosing proper $w_{12,14}, w_{12,22}$ and $w_{12,31}$, we can ensure that Δw_{18}= [+14, +22, +31] holds. Combined with $w_{14,i} = 0$

Table 2. Differential characteristic for steps 1-16 using signed bit-wise differences (32 steps)

Step	Differences of chaining values	w	w'	Pr	Sufficient conditions
	B_0:[+22] C_0:[-31] D_0:[-22,+31] F_0:[+12] G_0:[-31] H_0:[-12,+31]			2^{-6}	$A_{0,22} = C_{0,22}$, $E_{0,12} = G_{0,12}$, $D_{0,22} = 1, D_{0,31} = 0$, $H_{0,i} = 1 (i = 12, 31)$,
1	C_1:[+31] D_1:[-31] G_1:[+31] H_1:[-31]			2^{-2}	$D_{1,31} = 1$, $H_{1,31} = 1$.
2	D_2:[+31] H_2:[+31]	[+31]	[+31]	2^{-2}	$D_{2,31} = 0$, $H_{2,31} = 0$.
3				1	
⋮	⋮	⋮	⋮	⋮	⋮
14			[+14,+22,+31]	1	
15	A_{15}:[+14,+22,+31]			2^{-57}	$A_{16,i} = 0 (i = 1, 2, 9, 11, 14, 18, 22, 26, 31)$, $B_{16,i} = 0 (i = 6, 14, 21, 22, 29, 31)$, $C_{16,8} = D_{16,31}, C_{16,23} = D_{16,14}$, $C_{16,31} = D_{16,22}$, $(B_{16} \lll 12) + F_{16} = -(T_{15} \lll 15)$, $E_{16,i} = 0 (i = 1, 3, 9, 10, 18, 26, 27)$, $\bigoplus_{i \in \Lambda_j} E_{16,i} = 0,\ j \in \{1, 2, 3\}$, $\Lambda_1 = \{6, 14, 15, 16, 22, 24, 30, 31\}$, $\Lambda_2 = \{0, 6, 7, 14, 22, 23, 24, 30\}$, $\Lambda_3 = \{0, 7, 15, 16, 23, 31\}$.
16	A_{16}:[+1,+2,+9,+11, +14,+18,+22, +26,+31] B_{16}:[+14,+22,+31] E_{16}:[+1,+3,+9,+10 +18, +26, +27]				

$(i = 14, 22, 31)$, we can get $\Delta w_{14}'$= [+14, +22, +31]. So the differences of the message words in the backward direction are $\Delta w_2 =$ [+31], $\Delta w_2' =$ [+31], $\Delta w_{14}' =$ [+14, +22, +31], and all the other message words differences are zero.

If A_{15}[+14, +22, +31] holds, then we can cancel the differences $\Delta w_{14}' =$ [+14, +22, +31] in step 15, and skip 12 steps from step 15 to step 4 with probability 1. The following is the derivation for the sufficient conditions in step 16 of Table 2. The differential characteristic in step 16 is given by:

Table 3. Differential characteristic for steps 17-32 using signed bit-wise differences

Step	Differences of chaining values	w	w'	Pr	Sufficient conditions
16	A_{16}:[-6] D_{16}:[-6,-11,-13, -18,-19,-20, -22,-25,-26] E_{16}:[-28] H_{16}:[-4,-5,-11, -13,-19,-20, -22,-25,-26]	[+3,+4,+5, +11,+13, +19,+20, +22,+27]	[+3,+4,+5, +11,+13, +19,+20, +22,+27]		
17	B_{17}:[-6] F_{17}:[-28]			2^{-27}	$A_{16,i} = 1(i = 6, 23), A_{16,13} = 0$, $B_{16,6} = C_{16,6}$, $D_{16,i} = 1(i = 6, 11, 13, 18, 19, 20, 22, 25, 26)$, $E_{16,28} = 1, F_{16,28} = G_{16,28}$, $H_{16,i} = 1(i = 4, 5, 11, 13, 19, 20, 22, 25, 26)$. $SS1_{16,i} = 1(i = 3, 18, 25)$.
18	C_{18}:[-15] G_{18}:[-15]			2^{-2}	$A_{17,6} = C_{17,6}$, $E_{17,28} = 0$.
19	D_{19}:[-15] H_{19}:[-15]	[+15]	[+15]	2^{-2}	$A_{18,15} = B_{18,15}$, $E_{18,15} = 1$.
20				1	
⋮	⋮	⋮	⋮	⋮	⋮
31			[±6, +15, +30][1]	1	
32	A_{32}:[±6,+15,+30]			2^{-3}	

[1] The difference of w_2 affects $w_{31,6}$. In other words, if we choose $w^X_{31,6} - w^Y_{31,6} = +1$, then the difference $w^{X'}_{31,6} - w^{Y'}_{31,6} = -1$, and vice versa (We used " ± " to denote this fact). Note that this does not affect the XOR-differences in step-32.

$$(A_{16}[+1, +2, +9, +11, +14, +18, +22, +26, +31], B_{16}[+14, +22, +31], C_{16}, D_{16},$$

$$E_{16}[+1, +3, +9, +10, +18, +26, +27], F_{16}, G_{16}, H_{16}) \longrightarrow$$

$$(A_{15}[+14, +22, +31], B_{15}, C_{15}, D_{15}, E_{15}, F_{15}, G_{15}, H_{15}).$$

1. Because $A_{15} = B_{16}, B_{15} = (C_{16} \ggg 9), C_{15} = D_{16}$, $E_{15} = F_{16}, F_{15} = (G_{16} \ggg 19)$ and $G_{15} = H_{16}$, we choose $(B_{16} \lll 12)+F_{16} = -(T_{15} \lll 15)$ to ensure that $\Delta SS1 = \Delta((B_{16} \lll 12)+F_{16} + (T_{15} \lll 15)) \lll 7 = [+1, +9, +18]$ holds and the conditions $B_{16,i} = 0(i = 6, 21, 29)$ ensure that $\Delta SS2 = \Delta SS1 \oplus \Delta(B_{16} \lll 12)$= $[+1, +2, +9, +11, +18, +26]$ holds.
2. The conditions $B_{15,i} = C_{15,i}$, $(i = 14, 22, 31)$, i.e., $C_{16,8}$=$D_{16,31}$, $C_{16,23}$=$D_{16,14}$ and $C_{16,31}$=$D_{16,22}$ ensure that $\Delta FF_{15}(A_{15}, B_{15}, C_{15})$= $\Delta FF_{15}(B_{16}, C_{16} \ggg 9, F_{16})$= $[+14, +22, +31]$ hold. Combined with $\Delta w_{15}' = 0$, we can get $\Delta D_{15} = \Delta A_{16}$ - $(\Delta FF_{15}(B_{16}, C_{16} \ggg 9, D_{16}) + \Delta SS2 + \Delta w_{15}') = 0$. Similarly, the conditions $\bigoplus_{i \in \Lambda_j} E_{16,i} = 0$, $j \in \{1, 2, 3\}$, $\Lambda_1 = \{6, 14, 15, 16, 22, 24, 30, 31\}$, $\Lambda_2 = \{0, 6, 7, 14, 22, 23, 24, 30\}, \Lambda_3 = \{0, 7, 15, 16, 23, 31\}$ ensure that $\Delta H_{15} = 0$ holds.

Thus the above conditions constitute a set of sufficient conditions for the differential characteristic in step 16.

Table 4. Differential characteristic for steps 1-16 using signed bit-wise differences (33 steps)

Step	Differences of chaining values	w	w'	Pr	Sufficient conditions
	B_0:[+22] C_0:[-31] D_0:[-22,+31] F_0:[+12] G_0:[-31] H_0:[-12,+31]			2^{-6}	$A_{0,22} = C_{0,22}$, $E_{0,12} = G_{0,12}$, $D_{0,22} = 1, D_{0,31} = 0$, $H_{0,i} = 1(i = 12, 31)$.
1	C_1:[+31] D_1:[-31] G_1:[+31] H_1:[-31]			2^{-2}	$D_{1,31} = 1$, $H_{1,31} = 1$.
2	D_2:[+31] H_2:[+31]	[+31]	[+31]	2^{-2}	$D_{2,31} = 0$, $H_{2,31} = 0$.
3				1	
⋮	⋮	⋮	⋮	⋮	⋮
14			[+14,+22,+31]	1	
15	A_{15}:[+14,+22,+31]			2^{-59}	$A_{16,i} = 0(i = 3, 9, 11, 14, 15, 16, 17, 22, 26, 31)$, $A_{16,1} = 1, B_{16,i} = 0(i = 6, 14, 21, 22, 29, 31)$, $C_{16,8} = D_{16,31}, C_{16,23} \neq D_{16,14}, C_{16,31} = D_{16,22}$, $(B_{16} \lll 12) + F_{16} = -(T_{15} \lll 15)$, $E_{16,i} = 0(i = 1, 3, 9, 10, 18, 26, 27)$, $\bigoplus_{i \in \Lambda_j} E_{16,i} = 0,\ j \in \{1, 2, 3\}$, $\Lambda_1 = \{6, 14, 15, 16, 22, 24, 30, 31\}$, $\Lambda_2 = \{0, 6, 7, 14, 22, 23, 24, 30\}$, $\Lambda_3 = \{0, 7, 15, 16, 23, 31\}$.
16	A_{16}:[-1,+3,+9,+11, +14,+15,+16, +17,+22, +26,+31] B_{16}:[+14,+22,+31] E_{16}:[+1,+3,+9,+10 +18, +26, +27]				

- In the forward direction, we choose the message differences as follows: $\Delta w_{19} =$ [+15], $\Delta w_i = 0(20 \leq i \leq 34)$. Since $w_{19}{}' = w_{19} \oplus w_{23}$, we choose $w_{23,15} = 0$ to ensure that $\Delta w_{19}{}' =$ [+15]. Because $\Delta w_{19} =$ [+15] and $w_{19} = P_1(w_3 \oplus w_{10} \oplus (w_{16} \lll 15)) \oplus (w_6 \lll 7) \oplus w_{13}$, let $w_3 \oplus w_{10} = 0$ and $(w_6 \lll 7) \oplus w_{13} = 0$, to get Δw_{16}= [+3, +4, +5, +11, +13, +19, +20, +22, +27]. If we choose $w_{20,i} = 0$, $(i = 3, 4, 5, 11, 13, 19, 20, 22, 27)$ and $w_{23,15} = 0$, from $w_{16}{}' = w_{16} \oplus w_{20}$, we can get $\Delta w_{16}{}' = \Delta w_{16}$. Because $w_{35} = P_1(w_{19} \oplus w_{26} \oplus (w_{32} \lll 15)) \oplus (w_{22} \lll 7) \oplus w_{29}$ and $w_{31}{}' = w_{31} \oplus w_{35}$, we choose $w_{31,i} = 0$ $(i = 6, 15, 30)$, such that $\Delta w_{31}{}' =$ $[\pm 6, +15, +30]$[1].

[1] Because $w_{31} = P_1(w_{15} \oplus w_{22} \oplus (w_{28} \lll 15)) \oplus (w_{18} \lll 7) \oplus w_{25}$ and Δw_{18}= [+14, +22, +31] in the backward direction, if we choose $w^X_{31,i} = 0$ and $w^Y_{31,i} = 1(i = 6, 15, 30)$, then the bits in $w^{X'}_{31}$ and $w^{Y'}_{31}$ are $w^{X'}_{31,i} = 0(i = 15, 30)$, $w^{X'}_{31,6} = 1$ and $w^{Y'}_{31,i} = 1(i = 15, 30)$, $w^{Y'}_{31,6} = 0$. So

Table 5. Differential characteristic for steps 17-33 using signed bit-wise differences

Step	Differences of chaining values	w	w'	Pr	Sufficient conditions
16	C_{16}:[-0,-2,-5,-6,-18, -23,-25,-30,-31] D_{16}:[-27] G_{16}:[-0,-2,-5,-6,-16, -17,-23,-25,-31] H_{16}:[-3,-8,-10,-11, -19,-26,-27]		[+27]		
17	A_{17}:[-18] D_{17}:[-0,-2,-5,-6,-18, -23,-25,-30,-31] E_{17}:[-8] H_{17}:[-0,-2,-5,-6,-16, -17,-23,-25,-31]	[+0,+2,+7, +15,+16, +17,+23, +25,+31]	[+0,+2,+7, +15,+16, +17,+23, +25,+31]	2^{-54}	$A_{16,i}$=$B_{16,i}$$(i = 0, 2, 5, 6, 23, 25, 30, 31)$, $A_{16,18} \neq B_{16,18}$, $C_{16,i} = 1(i = 0, 2, 5, 6, 18, 23, 25, 30, 31)$, $TT1_{16,8} = 0$,$D_{16,27} = 1$, $E_{16,i}$=$1(i = 0, 2, 5, 6, 16, 23, 31)$, $E_{16,i} = 0(i = 17, 25)$, $G_{16,i} = 1(i = 0, 2, 5, 6, 16, 17, 23, 25, 31)$, $H_{16,i} = 1(i = 3, 8, 10, 11, 19, 26, 27)$, $TT2_{16,i} = 1(i = 3, 8, 10, 11, 17, 19, 25, 26, 27)$.
18	B_{18}:[-18] F_{18}:[-8]			2^{-9}	$A_{17,i} = 1(i = 3, 18), A_{17,25} = 0$, $B_{17,18} = C_{17,18}$, $E_{17,8} = 1, F_{17,8} = G_{17,8}$, $SS1_{17,i} = 1(i = 5, 15, 30)$.
19	C_{19}:[-27] G_{19}:[-27]			2^{-2}	$A_{18,18} = C_{18,18}$, $E_{18,8} = 0$.
20	D_{20}:[-27] H_{20}:[-27]	[+27]	[+27]	2^{-2}	$A_{19,27} = B_{19,27}$, $E_{19,27} = 1$.
21				1	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
32			[+10,+18,+27]	1	
33	A_{32}:[+10,+18,+27]			2^{-3}	

In this case, the massage word differences in the forward direction are $\Delta w_{16} = \Delta w_{16}'= [+3, +4, +5, +11, +13, +19, +20, +22, +27], \Delta w_{19} = \Delta w_{19}' = [+15], \Delta w_{31}'= [\pm 6, +15, +30]$, and all the other message words differences are zero.

The following is the derivation for the sufficient conditions for step 17 in Table 3. The differential characteristic in step 17 is given by:

$$(A_{16}[-6], B_{16}, C_{16}, D_{16}[-6, -11, -13, -18, -19, -20, -22, -25, -26], E_{16}[-28], F_{16},$$

$$G_{16}[+28], H_{16}[-4, -5, -11, -13, -19, -20, -22, -25, -26]) \longrightarrow$$

$$(A_{17}, B_{17}[-6], C_{17}, D_{17}, E_{17}, F_{17}[-28], G_{17}, H_{17}).$$

the difference of w_2 affects $w_{31,6}$. In other words, if we choose $w^X_{31,6} - w^Y_{31,6} = +1$, then the difference $w^{X'}_{31,6} - w^{Y'}_{31,6} = -1$, and vice versa (We used " ± " to denote this fact). Note that this does not affect the XOR-differences in step 32.

1. From $\varDelta A_{16} = [-6]$ and $B_{16,6} = C_{16,6}$, we get $\varDelta FF_{16}(A_{16}, B_{16}, C_{16}) = 0$. From $\varDelta E_{16} = [-28]$ and $F_{16,28} = G_{16,28}$, we get $\varDelta GG_{16}(E_{16}, F_{16}, G_{16}) = 0$.
2. From $\varDelta A_{16} = [-6]$ and $\varDelta E_{16} = [-28]$, it follows that the conditions $SS1_{16,3} = 1, SS1_{16,25} = 1$ ensure $\varDelta SS1_{16} = [-3, -25]$. Combined with the conditions $SS1_{16,18} = 1, A_{16,13} = 0$ and $A_{16,23} = 1$, we can get $\varDelta SS2_{16} = [+3, +18, -25]$. Therefore, $\varDelta A_{17} = \varDelta TT1_{16} = 0$, and $\varDelta E_{17} = \varDelta P_0(TT2_{16}) = 0$.

Thus the above conditions constitute a set of sufficient conditions for the differential characteristic in step 17.

4.2 Message Modification

We use the message modification technique which has been introduced by Wang *et al.* [18] to improve the complexity of our attack. We can modify the chaining values A_{16} to H_{16} to ensure that almost all the conditions in A_i to H_i (i=17,18,19) hold. For example, in the backward direction, we can modify $E_{16,i}(i = 6, 15)$ to make the sufficient conditions $\bigoplus_{i\in\Lambda_j} E_{16,i} = 0$, $j\in \{1, 2, 3\}$, $\Lambda_1 = \{6, 14, 15, 16, 22, 24, 30, 31\}$, $\Lambda_2 = \{0, 6, 7, 14, 22, 23, 24, 30\}, \Lambda_3 = \{0, 7, 15, 16, 23, 31\}$ hold.

In Table 3, there are 31 sufficient conditions from step 17 to step 19 in each differential. We can correct all the sufficient conditions in one differential by using message modification techniques, and the sufficient conditions $SS1_{16,i} = 1(i = 3, 18, 25)$, $A_{17,6} = C_{17,6}$, $E_{17,28} = 0$, $A_{18,15} = B_{18,15}$ and $E_{18,15} = 1$ are not corrected in another differential. So the probability of step 17 to step 19 can be improved from $2^{-2\times 31} = 2^{-62}$ to 2^{-7}. In Table 5, there are 63 sufficient conditions from step 17 to step 18 in each differential. We can correct all the sufficient conditions in one differential by using message modification techniques. However, the sufficient conditions $TT1_{16,8} = 0$, $TT2_{16,i} = 1(i = 3, 8, 10, 11, 17, 19, 25, 26, 27), A_{17,i} = 1(i = 3, 18), A_{17,25} = 0$, $E_{17,8} = 1$ and $SS1_{17,i} = 1(i = 5, 15, 30)$ are not corrected in the other differential. So the probability of step 17 to step 18 can be improved from $2^{-2\times 63} = 2^{-126}$ to 2^{-17}. Consequently, in this case, the probability of step 17 to step 20 can be improved from $2^{-2\times 67} = 2^{-134}$ to $2^{-17-2\times 4} = 2^{-25}$.

4.3 Boomerang Attacks on the 32-Step SM3 Compression Function

The attack algorithm on 32-step SM3 compression function can be summarized as follows.

1. Choose a random 512-bit message M and expand it to 36 words. Set proper message words as in section 4.1 to ensure that $\varDelta w_2 = [+31]$, $\varDelta w_2{}' = [+31]$, $\varDelta w_{14}{}'= [+14, +22, +31]$ in the backward direction, and $\varDelta w_{16} = \varDelta w_{16}{}'= [+3, +4, +5, +11, +13, +19, +20, +22, +27]$, $\varDelta w_{19} = [+15]$, $\varDelta w_{19}{}'= [+15]$, $\varDelta w_{31}{}'= [\pm 6, +21, +29]$ in the forward direction. Let $M_X = M$, $M_{X'} = M \oplus \varDelta w_2$. Expend the messages M_X and $M_{X'}$ to 36 words W_X and $W_{X'}$, respectively. Let $W_Y = W_X \oplus \varDelta w_{19}$ and $W_{Y'} = W_{X'} \oplus \varDelta w_{19}$. Then we use the 16 words $(W_Y)_{19}^{34}$ and $(W_{Y'})_{19}^{34}$ to get two 36-word $(W_Y)_0^{35}$ and $(W_{Y'})_0^{35}$ by using the message expansion algorithm. Let $M_Y = (W_Y)_0^{15}$, $M_{Y'} = (W_{Y'})_0^{15}$.

Table 6. Example of a quartet for 32 steps of the SM3 compression function

Message								
M_X	ecffec51	192fa6b4	6314d06c	f86f5604	ed6f140d	4597860c	3a4ccc9a	b78b2ded
	6a5b06f8	33169484	355b11c5	6b81ddd0	1e58820e	78fa46f6	742b217f	4669940f
$M_{X'}$	ecffec51	192fa6b4	e314d06c	f86f5604	ed6f140d	4597860c	3a4ccc9a	b78b2ded
	6a5b06f8	33169484	355b11c5	6b81ddd0	1e58820e	78fa46f6	742b217f	4669940f
M_Y	e8fff051	1bafa634	6b4cf854	a86ff654	6dea968e	4597060c	524cc49a	b78b25ed
	6a5b06f8	430624d4	3d5311d5	6b81ddd0	1e58020e	50fa4ef6	742b217f	4669940f
$M_{Y'}$	e8fff051	1bafa634	eb4cf854	a86ff654	6dea968e	4597060c	524cc49a	b78b25ed
	6a5b06f8	430624d4	3d5311d5	6b81ddd0	1e58020e	50fa4ef6	742b217f	4669940f
Chaining Value								
IV_X	0d548434	6f039a92	3d5fb868	01b03347	29c6a571	0d8b6217	4f2359fa	d6a363f4
$IV_{X'}$	0d548434	6f439a92	bd5fb868	81703347	29c6a571	0d8b7217	cf2359fa	56a373f4
IV_Y	792457dc	8f057732	6137fcd4	7899b663	948b29bf	d5f5a832	d9ae3751	c747e405
$IV_{Y'}$	792457dc	8f457732	e137fcd4	f859b663	948b29bf	d5f5b832	59ae3751	4747f405
H_X	a1e82b03	54b1bb42	2563b063	e514d921	a0eaf1fa	632d0eef	e2a999cf	ad4964d1
$H_{X'}$	3fd356c7	ca7f9b81	3a9694d6	31d02769	b454a3bd	c2d2dc37	45ffc720	2e319c71
H_Y	e1e8ab43	54b1bb42	2563b063	e514d921	a0eaf1fa	632d0eef	e2a999cf	ad4964d1
$H_{Y'}$	7fd3d687	ca7f9b81	3a9694d6	31d02769	b454a3bd	c2d2dc37	45ffc720	2e319c71

2. Randomly choose the chaining values A_{16}, B_{16}, C_{16}, D_{16}, E_{16}, G_{16} and H_{16} such that almost all the conditions used in step 16 and step 17[2] in Table 2 and Table 3 hold.
3. By using the message modification technique, modify $A_{16,19}$, $H_{16,28}$, $C_{16,15}$ and $G_{16,15}$ to make the sufficient conditions $A_{17,6} = C_{17,6}$, $E_{17,28} = 0$, $A_{18,15} = B_{18,15}$ and $E_{18,15} = 1$ hold in one of the differentials in the forward direction.
4. Use state update transformation process to get IV_X, IV_Y, $IV_{X'}$, $IV_{Y'}$, H_X, H_Y, $H_{X'}$ and $H_{Y'}$. Check whether $IV_X \oplus IV_{X'} = IV_Y \oplus IV_{Y'}$ and $H_X \oplus H_Y = H_{X'} \oplus H_{Y'}$ hold.
5. If a quartet is found, then a distinguisher is found. Repeat the above 4 steps with different messages and chaining values (A_{16} to H_{16}) until a distinguisher is found.

4.4 Complexity of the Attack

Using the differential characteristics and the message modification technique, we can construct the boomerang attack for the SM3 compression function reduced to 32 steps.

In the backward direction, all the sufficient conditions used in step 16 can be set in both of the differentials and the sufficient conditions used in step 3 to step 1 cannot be corrected in both of the differentials. So the differential characteristic used in the backward direction holds with probability 2^{-10}. Thus both of the differentials used in the backward direction hold with probability $2^{-10\times 2} = 2^{-20}$. In the forward direction, 7 sufficient conditions, from step 17 to step 19, are not corrected in one of the differentials by using message modifications and in step 32, non of the sufficient conditions is corrected in both of the differentials. Thus both of the differentials used in the forward direction hold with probability $2^{-7-3\times 2} = 2^{-13}$ after the message modification.

Hence, we can give a boomerang attack on 32-step SM3 with complexity $2^{20+13} = 2^{33}$. We can also use the amplified differential characteristics to improve the complexity of the attack. In this case, both of the two differentials used in the backward direction hold with probability $2^{-3.2}$ and the two differentials used in step 32 hold with probability

[2] All the conditions used in step 16 can hold in both of the differentials. In step 17 the conditions $SS1_{16,i} = 1 (i = 3, 18, 25)$ cannot be corrected in one of the differentials and all the other conditions can hold.

$2^{-4.2}$. So we can get a 32-step boomerang distinguisher with complexity $2^{3.2+7+4.2} = 2^{14.4}$. An example of a 32-step boomerang distinguisher (quartet) is presented in Table 6.

4.5 Attacks on 33/34/35 Steps SM3 Compression Function

In what follows we extend the proposed boomerang distinguisher to the 35-step SM3 compression function. First, we obtain a new 33-step boomerang distinguisher and then extend it to 35 steps. If we simply add one step in the forward differential characteristic in Table 3, this will result in some contradictions in A_{16} and E_{16} between Table 2 and Table 3. So we choose the message words differences as follows: $\varDelta w_2 = [+31], \varDelta w_i = 0$ $(0 \leq i \leq 15, i \neq 2)$ in the backward direction, and $\varDelta w_{20} = [+27], \varDelta w_i = 0 (21 \leq i \leq 35)$ in the forward direction.

We also correct $\varDelta A_{16} = [-1, +3, +9, +11, +14, +15, +16, +17, +22, +26, +31]$ and change one of the sufficient conditions $C_{16,23} = D_{16,14}$ to $C_{16,23} \neq D_{16,14}$.

In this case, the backward direction is from step 16 to step 1 and the differential characteristic which is given in Table 4 holds with probability $2^{-3.2}$. The forward direction is from step 17 to step 33 and the differential characteristic is given in Table 5 where 25 sufficient conditions are not corrected in one of the differentials by using message modifications. So the forward differential characteristic holds with probability $2^{-25-4.2} = 2^{-29.2}$. So we can get the 33-step boomerang distinguisher with complexity $2^{3.2+29.2} = 2^{32.4}$. In step 34 both of the differentials hold with probability $2^{-20.7}$ using the amplified differential characteristics. Thus we obtain a 34-step boomerang distinguisher with complexity $2^{32.4+20.7} = 2^{53.1}$. We also assume A_{35} and E_{35} all have 32-bit differences. Thus the 35-step boomerang distinguisher has a complexity $\approx 2^{53.1+2\times 32} = 2^{117.1}$.

5 A Slide-Rotational Property of SM3-XOR

In this section, we show that, in the case of the full SM3-XOR, pairs satisfying a certain rotational relation can be easily generated. An example of such a pair for the SM3-XOR is provided in Table 8. Such a property is not known to exist for SHA2-XOR [17].

The above mentioned property exists due to the fact that the constants over the 64 steps of SM3 are related. According to the SM3 specification, in steps $j \in \{0, \ldots, 15\}$, one constant rotated by j is utilized, whereas the other constant rotated by j is used in steps $j \in \{16, \ldots, 63\}$. Since operations like XOR, FF_i, GG_i, $0 \leq i < 64$, that are used in the SM3-XOR step function preserve the rotational property, it is natural to attempt a rotational attack, as provided below. We note that if instead of SM3-XOR, the original SM3 compression function is used, the addition mod 2^{32} transforms the attack into a probabilistic one, as outlined below. Due to the high number of additions per step, it appears difficult to exploit this rotational property directly and therefore the security of the SM3 compression function, at this stage of analysis, does not seem to be directly affected.

Two 32-bit words X, Y are said to be *rotational* if $X = Y \lll n$. Let messages W and W^* satisfy $W_1^* = W_0 \lll 1, W_2^* = W_1 \lll 1, \ldots, W_{16}^* = W_{15} \lll 1$. Below, a procedure for the instant generation of pairs v, v^* such that

Table 7. Example of a quartet for 33 steps of the SM3 compression function

Message								
M_X	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
$M_{X'}$	00000000	00000000	80000000	00000000	00000000	00000000	00000000	00000000
	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
M_Y	04001c00	02800080	08582838	5000a050	80858283	00008000	68000800	00000800
	00000000	7010b050	08080010	00000000	00008000	28000800	00000000	00000000
$M_{Y'}$	04001c00	02800080	88582838	5000a050	80858283	00008000	68000800	00000800
	00000000	7010b050	08080010	00000000	00008000	28000800	00000000	00000000
Chaining Value								
IV_X	274e6355	3333edb0	14f1b3d9	7be58154	d969d138	bb60c21a	ff5909df	e92dce5d
$IV_{X'}$	274e6355	3373edb0	94f1b3d9	fba58154	d969d138	bb60d21a	7f5909df	692dde5d
IV_Y	28b7b4d8	fe5f1155	93973138	c10d3808	32d4319b	dc8de94e	ef594319	8ef80fe1
$IV_{Y'}$	28b7b4d8	fe1f1155	13973138	414d3808	32d4319b	dc8df94e	6f594319	0ef81fe1
H_X	52793642	8017615c	fbf548ba	8b05cf67	dcb79a73	e1035e10	2caefeae	701d22d9
$H_{X'}$	772427a1	b2064c80	0dd79a89	2a809122	8bc2413f	8dd6b954	bad8867b	06c59c18
H_Y	987f3286	c017e19c	fbf548ba	8b05cf67	dabd9677	e1035e10	2caefeae	701d22d9
$H_{Y'}$	bd222365	f206cc40	0dd79a89	2a809122	8dc84d3b	8dd6b954	bad8867b	06c59c18

Table 8. An example for a slide-rotational pair for the SM3-XOR compression function

$A^1, B^1, \ldots, H^1$	0x565060b7 0x125d5655 0x285c7653 0xeaf5fe1e 0xda8bd7dd 0xb8bb1904 0x43bcaf18 0x7cf88895
$W^1_0, \ldots, W^1_{15}$	0x8f450bbd 0x4a0c9922 0x73dd44f8 0x9eceaaf8 0x33b13e20 0xb59d9c33 0x6b5a5f23 0xc0d2b468 0x7a9a1e16 0xaff62878 0x3fbb01f4 0x75278787 0xac0b849e 0x498f3045 0x62687c15 0xd3498eb

$A^2, B^2, \ldots, H^2$	0x24baacaa 0x53285c76 0xd5ebfc3d 0xdf1ee2a6 0x71763209 0x2bc610ef 0xf9f1112a 0xffeb86a4
$W^2_0, \ldots, W^2_{15}$	0x7efa7542 0x1e8a177b 0x94193244 0xe7ba89f0 0x3d9d55f1 0x67627c40 0x6b3b3867 0xd6b4be46 0x81a568d1 0xf5343c2c 0x5fec50f1 0x7f7603e8 0xea4f0f0e 0x5817093d 0x931e608a 0xc4d0f82a

$$\begin{aligned} v_1^* &= v_0 \lll 1, v_2^* = v_1 \lll 8, v_3^* = v_2 \lll 1 \\ v_5^* &= v_4 \lll 1, v_6^* = v_5 \lll 18, v_7^* = v_6 \lll 1 \\ V_1^* &= V_0 \lll 1, V_2^* = V_1 \lll 8, V_3^* = V_2 \lll 1 \\ V_5^* &= V_4 \lll 1, V_6^* = V_5 \lll 18, V_7^* = V_6 \lll 1 \end{aligned} \tag{4}$$

is provided, where $V = \text{SM3-XOR}(v, W)$, $V^* = \text{SM3-XOR}(v^*, W^*)$ and v_i, V_i for $0 \leq i \leq 7$ denote i-th 32-bit word in the v and V, respectively. For a random function, a random (v, W), (v^*, W^*) satisfying the above constraints will yield the corresponding V and V^* with probability $2^{-6\times 32} = 2^{-192}$, since (4) imposes 6 32-bit conditions on V, V^*.

5.1 Constructing a Slide-Rotational Pair

In this section, step i denotes the transformation from $(A_i, B_i, C_i, D_i, E_i, F_i, G_i, H_i)$ to $(A_{i+1}, B_{i+1}, C_{i+1}, D_{i+1}, E_{i+1}, F_{i+1}, G_{i+1}, H_{i+1})$. For example by step 0 the first compression function step is denoted. We start by the following observations:

- The slide rotational messages expand to slide-rotational expanded messages with probability 1. In particular, fix $W_0, \ldots, W_{15}$ and let

$$W_1^* = W_0 \lll 1, W_2^* = W_1 \lll 1, \ldots, W_{16}^* = W_{15} \lll 1 \tag{5}$$

After expanding both W and W^*, we have $W_{i+1}^* = W_i \lll 1$, for $i = \{0, 1, \ldots, 62\}$ and also $W_{i+1}^{\prime *} = W_i' \lll 1$, for $i = \{0, 1, \ldots, 66\}$.

- We recall that T_i, $0 \leq i \leq 63$ are the step constants. If we have

$$W^*_{i+1} = W_i \lll 1, W'^*_{i+1} = W'_i \lll 1, T_{i+1} = T_i \lll 1 \tag{6}$$

$$A^*_{i+1} = A_i \lll 1, B^*_{i+1} = B_i \lll 1, \ldots, H^*_{i+1} = H_i \lll 1 \tag{7}$$

for $i = k$, then (7) will also hold for $i = k + 1$, where $k = 0, \ldots, 62$.

The observations above suggest that sliding can be introduced, as depicted in Fig. 2.

Namely, consider randomly initializing W and letting W^* satisfy (5). Moreover, $A_0, B_0 \ldots, H_0$ is chosen randomly and the inner state registers after the first step in the second instance of the hash function are initialized according to (7). Then, until step 15, due to (6), the rotational property in the inner state registers will be preserved. Once the two instances reach steps 15 and 16, respectively, a different step transformation is applied in the two instances and the rotational property may discontinue. This problem is bypassed by starting from the middle, i.e., by populating the inner states entering the critical steps 15 and 16 (see Fig. 2).

5.2 Bypassing Steps 15 and 16

The idea is to start by populating the inner states entering the critical steps 15 and 16 (see Fig. 2). In particular, a rotational pair $(A_{15}, \ldots, H_{15})$, $(A^*_{16}, \ldots, H^*_{16})$ is carefully chosen so that $(A_{16}, \ldots, H_{16})$ and $(A^*_{17}, \ldots, H^*_{17})$ satisfy relation (7). It should be noted that the rotational property may be destroyed only between A_{16} and A^*_{17} and between E_{16} and E^*_{17}, since the other registers go through identical rotational-preserving transformations in step 15 and step 16. As for A_{16} and A^*_{17}, for the purpose of tracking the possible rotational disturbance between the two registers, the equation to compute these two registers can be rewritten as

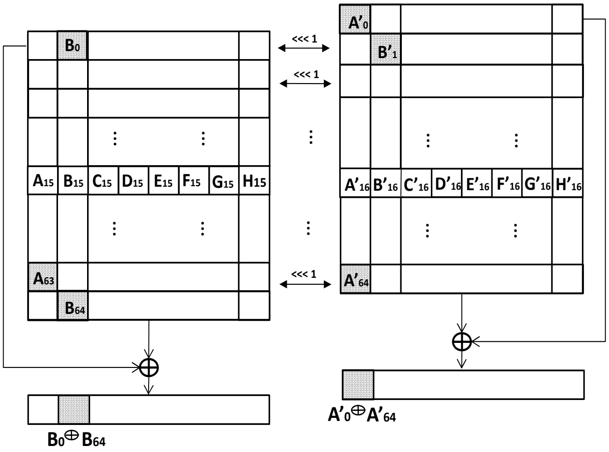

Fig. 2. The slide-rotational attack against SM3-XOR

$$A_{16} = FF_{15}(A_{15}, B_{15}, C_{15}) \oplus (T_{15} \lll 22) \oplus \alpha \tag{8}$$

$$A^*_{17} = FF_{16}(A^*_{16}, B^*_{16}, C^*_{16}) \oplus (T_{16} \lll 23) \oplus \alpha^* \tag{9}$$

where $\alpha = D_{15} \oplus W_{15} \oplus W_{19} \oplus (((A_{15} \lll 12) \oplus E_{15}) \lll 7) \oplus (A_{15} \lll 12)$ and $\alpha^* = D^*_{16} \oplus W^*_{16} \oplus W^*_{20} \oplus (((A^*_{16} \lll 12) \oplus E^*_{16}) \lll 7) \oplus (A^*_{16} \lll 12)$. Since (7) and (6) hold for $i = 15$, $\alpha^* = \alpha \lll 1$. Therefore, to have A_{16} and A^*_{17} be a rotational pair, it suffices to make $FF_{15}(A_{15}, B_{15}, C_{15}) \oplus (T_{15} \lll 22)$ and $FF_{16}(A^*_{16}, B^*_{16}, C^*_{16}) \oplus (T_{16} \lll 23)$ satisfy the rotational property. After expressing A^*_{16}, B^*_{16}, C^*_{16} in terms of A_{15}, B_{15}, C_{15} and using that FF_{15} and FF_{16} preserve the rotational property, the condition can be expressed in terms of A_{15}, B_{15}, C_{15} as follows:

$$FF_{15}(A_{15}, B_{15}, C_{15}) \oplus FF_{16}(A_{15}, B_{15}, C_{15}) = (T_{15} \oplus T_{16}) \lll 22 \tag{10}$$

When applied on 1-bit values X, Y and Z, the equation $FF_{15}(X, Y, Z) \oplus FF_{16}(X, Y, Z) = 0$ is satisfied for 2 out of 8 (X, Y, Z) values. Since the Hamming weight of the right-hand side of (10) is equal to 14, the number of solutions to the equation is $2^{18} \times 6^{14} = 2^{32} \times 3^{14}$. As for preserving the rotational property between E_{16} and E^*_{17}, developing the registers as in (8) and then forming the equation of the form (10) yields that the number of solutions E_{15}, F_{15} and G_{15} is $4^{32} = 2^{64}$. Therefore, the number of solutions for $(A_{15}, \ldots, H_{15})$ that pass the disturbance in steps 15 and 16 is $2^{32} \times 3^{14} \times 2^{64} \times 2^{64} \approx 2^{182.19}$, since D_{15} and H_{15} are free variables. For such pairs, it follows that relations (4) are satisfied.

When instead of SM3-XOR, the SM3 compression function is considered, this property turns into a probabilistic one. Following [15], if $p_r = P[(x \lll r) + (y \lll r) = (x + y) \lll r]$ where x and y are 32-bit words, then $p_1 = 2^{-1.415}$. Since there exists 8 additions in one SM3 step, the probability that one step and its corresponding slided step will preserve the rotational property is given by $(p_1)^8 = 2^{-11.320}$ [15].

6 Conclusions

In this paper, we have shown an application of the boomerang-style attack on the step-reduced SM3 compression function. In particular, we presented distinguishing attacks for 32 steps of the compression function with complexity $2^{14.4}$, 33 steps with complexity $2^{32.4}$, 34 steps with complexity $2^{53.1}$ and 35 steps with complexity $2^{117.1}$. Our results suggest that 35-step SM3 compression does not behave randomly. In the second part of the paper, a slide-rotational property of SM3-XOR function is exposed and an example of a slide-rotational pair for SM3-XOR compression function is given.

Acknowledgment. The authors would like to thank the anonymous reviewers for valuable comments that helped improve the presentation of the paper. The work of Gaoli Wang and Yanzhao Shen is supported by the National Natural Science Foundation of China (No. 61103238), Chen Guang project (supported by Shanghai Municipal Education Commission and Shanghai Education Development Foundation), the Fundamental Research Funds for the Central Universities. The work of Aleksandar Kircanski and Amr Youssef is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

1. Specification of SM3 cryptographic hash function (in Chinese), http://www.oscca.gov.cn/UpFile/20101222141857786.pdf/
2. Zou, J., Wu, W., Wu, S., Su, B., Dong, L.: Preimage Attacks on Step-Reduced SM3 Hash Function. In: Kim, H. (ed.) ICISC 2011. LNCS, vol. 7259, pp. 375–390. Springer, Heidelberg (2012)
3. Aumasson, J.P.: Zero-sum Distinguishers. Rump session talk at CHES 2009 (2009), http://131002.net/data/papers/AM09.pdf
4. Wagner, D.: The Boomerang Attack. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 156–170. Springer, Heidelberg (1999)
5. Biryukov, A., Nikolić, I., Roy, A.: Boomerang Attacks on BLAKE-32. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 218–237. Springer, Heidelberg (2011)
6. Lamberger, M., Mendel, F.: Higher-Order Differential Attack on Reduced SHA-256. Cryptology ePrint Archive: Report 2011/037, http://eprint.iacr.org/
7. Biryukov, A., Lamberger, M., Mendel, F., Nikolić, I.: Second-Order Differential Collisions for Reduced SHA-256. In: Lee, D.H. (ed.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 270–287. Springer, Heidelberg (2011)
8. Mendel, F., Nad, T.: Boomerang Distinguisher for the SIMD-512 Compression Function. In: Bernstein, D.J., Chatterjee, S. (eds.) INDOCRYPT 2011. LNCS, vol. 7107, pp. 255–269. Springer, Heidelberg (2011)
9. Sasaki, Y.: Boomerang Distinguishers on MD4-Family: First Practical Results on Full 5-Pass HAVAL. In: Miri, A., Vaudenay, S. (eds.) SAC 2011. LNCS, vol. 7118, pp. 1–18. Springer, Heidelberg (2012)
10. Sasaki, Y., Wang, L.: 2-Dimension Sums: Distinguishers Beyond Three Rounds of RIPEMD-128 and RIPEMD-160, http://eprint.iacr.org/2012/049.pdf
11. Aumasson, J.-P., Çalık, Ç., Meier, W., Özen, O., Phan, R.C.-W., Varıcı, K.: Improved Cryptanalysis of Skein. In: Matsui, M. (ed.) ASIACRYPT 2009. LNCS, vol. 5912, pp. 542–559. Springer, Heidelberg (2009)
12. Leurent, G., Roy, A.: Boomerang Attacks on Hash Function Using Auxiliary Differentials. In: Dunkelman, O. (ed.) CT-RSA 2012. LNCS, vol. 7178, pp. 215–230. Springer, Heidelberg (2012)
13. Chen, J., Jia, K.: Improved Related-Key Boomerang Attacks on Round-Reduced Threefish-512. In: Kwak, J., Deng, R.H., Won, Y., Wang, G. (eds.) ISPEC 2010. LNCS, vol. 6047, pp. 1–18. Springer, Heidelberg (2010)
14. Yu, H., Chen, J., Wang, X.: The Boomerang Attacks on the Round-Reduced Skein-512. In: Knudsen, L.R., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 288–304. Springer, Heidelberg (2012)
15. Khovratovich, D., Nikolić, I.: Rotational Cryptanalysis of ARX. In: Hong, S., Iwata, T. (eds.) FSE 2010. LNCS, vol. 6147, pp. 333–346. Springer, Heidelberg (2010)
16. Biryukov, A., Wagner, D.: Slide Attacks. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 245–259. Springer, Heidelberg (1999)
17. Yoshida, H., Biryukov, A.: Analysis of a SHA-256 Variant. In: Preneel, B., Tavares, S. (eds.) SAC 2005. LNCS, vol. 3897, pp. 245–260. Springer, Heidelberg (2006)
18. Wang, X., Yu, H.: How to Break MD5 and Other Hash Functions. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 19–35. Springer, Heidelberg (2005)
19. Wagner, D.: A Generalized Birthday Problem. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 288–303. Springer, Heidelberg (2002)