Information Processing Letters 113 (2013) 764–770

Contents lists available at SciVerse ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Second order collision for the 42-step reduced DHA-256 hash function

Riham AlTawy, Aleksandar Kircanski, Amr Youssef*

Concordia Institute for Information Systems Engineering (CIISE), Concordia University, 1455 De Maisonneuve Blvd. W., Montreal, Quebec, H3G 1M8, Canada

ARTICLE INFO

Article history:
Received 29 April 2013
Received in revised form 16 July 2013
Accepted 16 July 2013
Available online 23 July 2013
Communicated by D. Pointcheval

Keywords:
Cryptography
Cryptanalysis
Hash functions
Second order collisions
Boomerang attack
DHA-256

ABSTRACT

At the Cryptographic Hash Workshop hosted by NIST in 2005, Lee et al. proposed the DHA-256 (Double Hash Algorithm-256) hash function. The design of DHA-256 builds upon the design of SHA-256, but introduces additional strengthening features such as optimizing the message expansion and step function against local collision attacks. Previously, DHA-256 was analyzed by J. Zhong and X. Lai, who presented a preimage attack on 35 steps of the compression function with complexity $2^{239.6}$. In addition, the IAIK Krypto Group provided evidence that there exists a 9-step local collision for the DHA-256 compression function with probability higher than previously predicted. In this paper, we analyze DHA-256 in the context of higher order differential attacks. In particular, we provide a practical distinguisher for 42 out of 64 steps and give an example of a colliding quartet to validate our results.

1. Introduction

In 2012, Keccak has been selected as the winner of the NIST hash function competition [5]. However, according to the eBASH project (ECRYPT Benchmarking of All Submitted Hashes),[1] SHA-2 outperforms Keccak on a significant number of widely used general purpose CPUs. Thus, the designs based on the SHA-2 hash may remain a compelling choice in the area of software oriented applications. On the side of attacks on the hash function following the SHA design approach, apart from the seminal attack [20] on SHA-1 by Wang, some progress has been reported in analysis of SHA-2: collisions and pseudo-collisions for variants of SHA-2 reduced to 32 and 38-step have been reported [15,14]. One of the early proposals that aim to strengthen the SHA-2 design is the DHA-256 hash function. The main innovation applied in the design of DHA-256 is tweaking the message expansion and the step function so that the possibility of repeating the patterns that potentially cause local collisions is minimal. Another change is increasing the number of modular additions in one step to 8. In this work, we revisit the security of DHA-256 hash function, assessing its security against second order differential distinguishers. Distinguishing attacks are important in the context of hash functions since compression functions are often modeled as random oracles and the potential existence of distinguishing properties can be used for disproving indifferentiability claims [6]. Furthermore, hash functions are modeled as random oracles in many schemes, e.g. the Optimal Asymmetric Encryption Padding and Probabilistic Signature Scheme [1] and the security of these schemes depends on the randomness of the underlying primitives [4]. Therefore, studying the distinguishing properties of compression functions is of importance from the perspective of proving the overall security of a particular system. Previous literature related to the higher order distinguishers on hash functions includes the pioneering works on higher order differentials by Lai [9] and the boomerang attacks by Wagner [19], both originally proposed for block ciphers. These two works have

* Corresponding author.
E-mail address: youssef@ciise.concordia.ca (A. Youssef).

[1] eBASH website: http://bench.cr.yp.to/ebash.html.

0020-0190/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ipl.2013.07.015

been adapted to hash functions independently by Biryukov et al. [3] and Lamberger and Mendel [10]. In particular, in [3], a distinguisher for the 7-round BLAKE-32 was provided, while in [10] a practical distinguisher for the 46-step SHA-2 compression function was given. The latter SHA-2 result was extended to 47 steps in [2]. Subsequently, boomerang distinguishers have been applied to other SHA-based functions. SIMD-512 compression function was analyzed by Mendel and Nad [13] and a boomerang distinguisher of theoretical complexity for the full function was presented. Sasaki et al. [17,18] provided a practical boomerang distinguisher on the full compression function of 5-pass HAVAL and also a distinguisher for the full HAS-160 compression function. Kircanski et al. [8] provided a practical zero sum for the 33-step SM3 compression function. In [12], Leurent and Roy showed that, under some conditions, three independent paths instead of two can be combined to achieve a better distinguishing complexity in a boomerang attack.

The DHA-256 hash function was first analyzed by the IAIK Krypto Group [7] where a local collision differential pattern was shown to exist with probability 2^{-63}, which is higher than it was anticipated by the designers. Later, Zhong and Lai [21] studied preimage resistance of DHA-256 using variants of meet in the middle attacks, reaching the complexity of $2^{239.6}$ function evaluations for a 35-step pseudo-preimage attack and $2^{248.8}$ for the preimage attack.

In this paper, we provide a practical distinguisher for the 42-step-reduced DHA-256 compression function. In particular, we present two independent differential characteristics and state the conditions imposed on each step during the quartet computation. The two provided paths consist of the internal and external parts: the approach used to satisfy the internal parts of the characteristics is to enforce one path by using message modification on the two faces of the boomerang, while simultaneously verifying whether the other path is satisfied on the other two faces of the boomerang. The external parts of the paths are satisfied probabilistically. Apart from satisfying the inner state differentials, by properly choosing the message words, we maximize the probability of expanding the messages following the message difference pattern, taking into account the non-linearity of the message expansion in DHA-256.

The paper is organized as follows. In the next section, the specification of the DHA-256 function along with the notation used throughout the paper is provided. A brief overview of second order differential attacks is provided in Section 3. Afterwards, we provide detailed description of our attack, differential characteristics, and the attack's complexity in Section 4. Finally, the paper is concluded in Section 5.

2. Specification of DHA-256

DHA-256 [11] is a Merkle–Damgård based hash function. Its compression function maintains a state of eight 32-bit words, produces 256-bit digests and takes 512-bit message blocks on the input. The input message is expanded according to the message expansion function. Each expanded message word is used twice in each step which increases the message diffusion rate. The compression function inner state consists of eight 32-bit words each step. See Fig. 1.

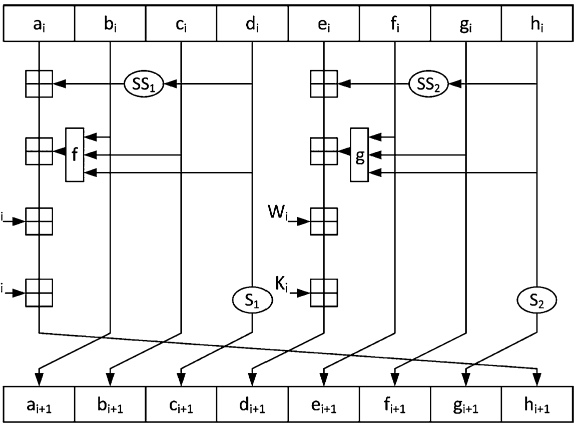

Fig. 1. The step function of DHA-256.

2.1. The state update function

The functions start by initializing the chaining variables $(a_0, b_0, c_0, \dots, h_0)$ with eight 32-bit words initial values (IVs) and updates them iteratively for 64 steps using the following operations

$$h_{i+1} = a_i + SS_1(d_i) + f(b_i, c_i, d_i) + W_i + K_i$$

$$d_{i+1} = e_i + SS_2(d_i) + g(f_i, g_i, h_i) + W_i + K_i$$

$$b_{i+1} = S_1(c_i), \qquad f_{i+1} = S_2(g_i), \qquad a_{i+1} = b_i$$

$$c_{i+1} = d_i, \qquad e_{i+1} = f_i, \qquad g_{i+1} = h_i$$

The auxiliary functions, SS_1 and SS_2, both operating on 32-bit words, and are defined by

$$SS_1(X) = X \oplus (X \lll 11) \oplus (X \lll 25)$$

$$SS_2(X) = X \oplus (X \lll 19) \oplus (X \lll 29)$$

The rotations S_1 and S_2, are defined by

$$S_1(X) = X \lll 17, \qquad S_2(X) = X \lll 2$$

The Boolean functions f and g are defined by

$$f(X, Y, Z) = (X \wedge Y) \vee (\neg X \wedge Z)$$

$$g(X, Y, Z) = (X \wedge Y) \oplus (Y \wedge Z) \oplus (X \wedge Z)$$

The step constants K_i and the IV values are available in the full function specification [11].

2.2. The message expansion

Each 512-bit message block M is split into 16 words $M_0 \| M_1 \| \cdots \| M_{15}$ and then it is expanded to 64 32-bit words $(W_0, W_1, \dots, W_{63})$ by

$$W_i = \begin{cases} M_i, & 0 \leqslant i \leqslant 15 \\ \sigma_1(W_{i-1}) + W_{i-9}, & \\ \quad + \sigma_2(W_{i-15}) + W_{i-16} & 16 \leqslant i \leqslant 63 \end{cases}$$

where the functions σ_1 and σ_2 are defined as follows

$$\sigma_1(X) = X \oplus (X \lll 7) \oplus (X \lll 22)$$

$$\sigma_2(X) = X \oplus (X \lll 13) \oplus (X \lll 27)$$

After the last step, the initial values are added to the output values of the last step (Davies–Meyer construction) resulting in the output of the compression function.

2.3. Notation

Let x be an n-bit word. The following notation will be used throughout the paper:

- x_i: the ith bit of a word x.
- $a_{j,i}, b_{j,i}, \ldots, h_{j,i}$, $0 \leqslant j \leqslant 63$ and $0 \leqslant i \leqslant 31$: the ith bit of $a, b, \ldots, h$ internal register at step j (e.g. $b_{35,10}$ is the value of the 10th bit of register b at step 35).
- m_0, m_1, m_2, m_3: the four messages that form the boomerang quartet.
- $S_i^{(m_0,m_1,m_2,m_3)}$, $0 \leqslant i \leqslant 63$: the internal state at the ith step of one of the quartet messages (m_0, m_1, m_2, m_3) executions.
- $(a, b, \ldots, h)_i^{(m_0,m_1,m_2,m_3)}$, $0 \leqslant i \leqslant 63$: the internal state register $(a, b, c, \ldots, h)$ of one of the quartet messages (m_0, m_1, m_2, m_3) executions at the ith step.
- $W_i^{(m_0,m_1,m_2,m_3)}$, $0 \leqslant i \leqslant 63$: the message word at the ith step of one of the quartet messages (m_0, m_1, m_2, m_3) executions.

To express the differences between 32-bit words, we used the usual signed bit difference notation [20].

3. Second order collision attacks on hash functions

In this section, we review higher order differential attacks on hash functions and show how the boomerang attack technique is adapted from block ciphers to hash functions to construct second order collisions. First, notions related to higher order differentials as introduced by Lai [9] are recalled.

Definition 1. Let $(S, +)$ and $(T, +)$ be abelian groups. For a function $f : S \to T$, the 1st derivative of f at a point $\alpha \in S$ is defined as

$$\Delta(\alpha) f(m) = f(m + \alpha) - f(m)$$

The ith derivative of f at $(\alpha_1, \alpha_2, \ldots, \alpha_i) = h$ is then defined recursively as

$$\Delta(\alpha_1, \ldots, \alpha_i) f(m) = \Delta(\alpha_i)\bigl(\Delta(\alpha_1, \ldots, \alpha_{i-1}) f(m)\bigr) = h$$

Accordingly, we have a second order differential collision for a function when we can find an input m and two differences (δ, α) such that

$$h(m + \delta + \alpha) - h(m + \delta) - h(m + \alpha) + h(m) = 0 \qquad (1)$$

Since (δ, α) and m can be chosen freely, the query complexity for finding a second order collision is $2^{n/3}$ [2], where n denotes the bit-size of the output of the function f. By the query complexity, we consider the number of queries required to be made to the f function oracle. On the other hand, as for the computational complexity, which would include evaluating f around $2^{n/3}$ times and finding a quartet that sums to 0, the best currently known algorithm runs in complexity no better than $2^{n/2}$. If for a particular function a second order collision is obtained with a complexity lower than $2^{n/2}$, then this hash function deviates from the random function oracle.

4. Second order distinguisher for DHA-256

The main steps of the attack can be summarised as follows: (1) properly choosing the two message/inner state differentials, (2) applying the quartet message modification in the middle steps of the compression function, and (3) randomly satisfying the remaining portions of the two differentials.

4.1. Choosing the differential characteristics

Following the boomerang approach, we have to choose two independent characteristics, which means that as long as there is no contradicting conditions in the steps at which they intersect, a change in the message differential in one characteristic should not affect the other. Due to non-linearity of the message expansion, this holds both for the chaining values characteristic and for the message characteristic. The backward differential will be denoted by C_0 and the forward differential by C_1, both for the message and the chaining values. The chaining value differentials are provided in Tables 1 and 2 and the message differentials are

- C_0: $\Delta W_4 = -31$, $\Delta W_{19} = -31, -26, -16$.
- C_1: $\Delta W_{25} = +31$, $\Delta W_{40} = +31, +26, +16$.

In the sequel, we explain the rationale behind the choice of the message and the chaining values differentials. The forward differential consists of steps 20–41, whereas the backward differential consists of steps 0–20. We differentiate between two portions in each of the two chaining values differentials: the *internal* part (steps 20 and 20–26 in C_0 and C_1, respectively) and the *external* part (steps 0–4 and 41 in C_0 and C_1, respectively). The internal part is satisfied by message modification, while the external part is satisfied randomly. Now, the message differentials were chosen so that; (1) the backward chaining value characteristic caused by the message difference has a practical probability in the external part and (2) the forward characteristic C_1 message differential does not corrupt the message differential in C_0 when the quartet expands. More precisely, we have chosen the difference at W_4, because a difference at this step leads to! an attack with practical probability 2^{-46} and can be satisfied randomly, whereas, a difference one step further would result in an unreachable attack complexity. Moreover, the difference at W_{25} keeps the independence constraint, while a difference any step further corrupts the predefined message difference in C_0. Furthermore, we have chosen the message differences in C_0 and C_1 at bit 31 to maximize the probability of the differential characteristics, because addition of the MSB

Table 1
Backward differential characteristic C_0 with probability 2^{-33}.

Step	State variables	W_i	Sufficient conditions	Prob.
External part				
0	a: +29 b: −25, −7 e: +14 f: −26, −16		$a_{0,29}=0$, $b_{0,25}=1$, $b_{0,7}=1$ $c_{0,29}=0$, $d_{0,29}=1$ $d_{0,25}=c_{0,25}$, $d_{0,7}=c_{0,7}$ $e_{0,14}=0$, $f_{0,26}=1$ $f_{0,16}=1$, $h_{0,26}=g_{0,26}$ $h_{0,16}=g_{0,16}$, $h_{0,29}\neq g_{0,27}$	2^{-13}
1	a: −25, −7 d: +14 e: −26, −16 h: +29		$d_{1,14}=0$, $d_{1,7}=d_{1,28}$ $d_{1,25}=d_{1,0}$, $d_{1,3}\neq d_{1,21}$ $h_{1,29}=0$, $h_{1,16}=h_{1,19}$ $h_{1,26}=h_{1,7}$, $h_{1,10}\neq h_{1,0}$	2^{-8}
2	c: +14 g: +29		$h_{2,29}=h_{0,27}$	2^{-1}
3	b: +31 f: +31		$d_{3,31}=d_{2,31}$ $h_{3,31}=h_{2,31}$	2^{-2}
4	a: +31 e: +31	−31		1
5				1
	⋮			
18				1
Internal part				
19		−31 −26 −16	$W_{4,12}=W_{4,17}$ $W_{4,26}=W_{4,13}$ $W_{4,18}=W_{4,4}$	2^{-3}
20	d: −31, −26, −16 h: −31, −26, −16		$d_{20,31}=1$, $d_{20,26}=1$ $h_{20,31}=1$, $d_{20,16}=1$ $h_{20,26}=1$, $h_{20,16}=1$	2^{-6}

Table 2
Forward differential characteristic C_1 with probability 2^{-37}.

Step	State variables	W_i	Sufficient conditions	Prob.
Internal part				
20	b: −29 c: +22, +8 f: −14 g: +24, +14		$b_{20,29}=1$, $b_{20,21}=0$ $c_{20,22}=0$, $b_{20,8}=0$ $c_{20,8}=0$, $f_{20,14}=1$ $d_{20,29}=c_{20,29}$, $d_{20,29}=0$ $g_{20,24}=0$, $g_{20,14}=0$ $h_{20,24}=f_{20,24}$	2^{-11}
21	a: −29 b: +25, +7 e: −14 f: +26, +16		$d_{21,29}=0$, $d_{21,25}=d_{20,25}$ $d_{21,7}=d_{20,7}$, $d_{21,26}=d_{20,26}$ $h_{21,16}=h_{20,16}$ $h_{21,29}\neq h_{20,27}$	2^{-6}
22	a: +25, +7 d: −14 e: +26, +16 h: −29		$d_{22,14}=1$, $d_{22,7}=d_{22,28}$ $d_{22,25}=d_{22,0}$, $d_{22,3}\neq d_{22,21}$ $h_{22,29}=1$, $h_{22,16}=h_{22,19}$ $h_{22,26}=h_{22,7}$, $h_{22,10}\neq h_{22,0}$	2^{-8}
23	c: −14 g: −29		$h_{23,29}=h_{21,27}$	2^{-1}
24	b: −31 f: −31		$d_{24,31}=d_{23,31}$ $h_{24,31}=h_{23,31}$	2^{-2}
25	a: −31 e: −31	+31 +31		1
26				1
	⋮			
39				1
External part				
40		+31 +26 +16	$W_{4,12}=W_{4,17}$ $W_{4,26}=W_{4,13}$ $W_{4,18}=W_{4,4}$	2^{-3}
41	d: +31, +26, +16 h: +31, +26, +16		$d_{41,31}=0$, $d_{41,26}=0$ $h_{41,31}=0$, $d_{41,16}=0$ $h_{41,26}=0$, $h_{41,16}=0$	2^{-6}

causes no carry propagation and thus controlling difference diffusion requires fewer conditions. The other differences at W_{19} and W_{40} are due to the application of σ_2 function on W_4 and W_{25} respectively. Since the σ_2 function is linear and has no absorption property, we can only control the sign of the difference.

Next, the choice of chaining value differentials is explained. The main requirement for these two characteristics is that they are compatible (i.e. do not contradict each other) in the steps where they intersect (contradictory differentials in the context of boomerang attacks have first been pointed out by Murphy [16]). The compatibility requirement can be explained as follows. Let at some steps the chaining values follow some prespecified differential for a pair of values. Now, apply the difference specified by the other differential to both instances of the inner state. If the newly obtained pair after propagating the other differential also follows the prespecified differential, then the two differentials are compatible. Naturally, the smaller number of differences the characteristics contain, the smaller will be probability of contradiction between the characteristics. To get sparse differences, we started with two linearized differentials both triggered by most significant bit in the message, the function was linearized as follows: (1) replacing addition mod 2^{32} by XOR, (2) replacing the logical functions f and g by 0-function through absorbing the inputs, except when their output can cancel another difference in the step addition operation, or when their three inputs have differences; in these cases we allow the input to pass.

In particular, to construct the differential characteristics, we propagate the difference at W_4 backward and the result is the external part of C_0 from S_4 to S_0. Also, the difference introduced by W_{19} is propagated forward, thus creating the internal part of C_0, which is S_{20}, as depicted in Table 1. Similarly, the difference at W_{25} is propagated backward to create the internal part of C_1 from S_{20} to S_{26} and the difference introduced by W_{40} is propagated forward, thus creating the external part of C_1, which is S_{41} (see Table 2). A schematic view of the position of the differences and how they are propagated between the quartet messages is provided in Fig. 2. The compatibility was confirmed by finding particular values satisfying both differentials at the intersecting step. In Tables 1 and 2, apart from the two differential characteristics, their corresponding sufficient conditions are provided.

4.2. Message modification

Message modification was performed on the quartet steps 20–26. In the sequel, we provide the exact steps of our search for the quartet conforming to the internal parts of the two paths. Let m_0, m_1, m_2 and m_3 denote the quartet $(M, M+\delta, M+\alpha, M+\delta+\alpha)$, respectively. In order to be able to produce the whole 64 words and to satisfy steps 20–26 by message modification, we want to find $W_i^{m_0,m_1,m_2,m_3}$ for $18 \leqslant i \leqslant 33$ such that $W_i^{m_0,m_1,m_2,m_3}$ for

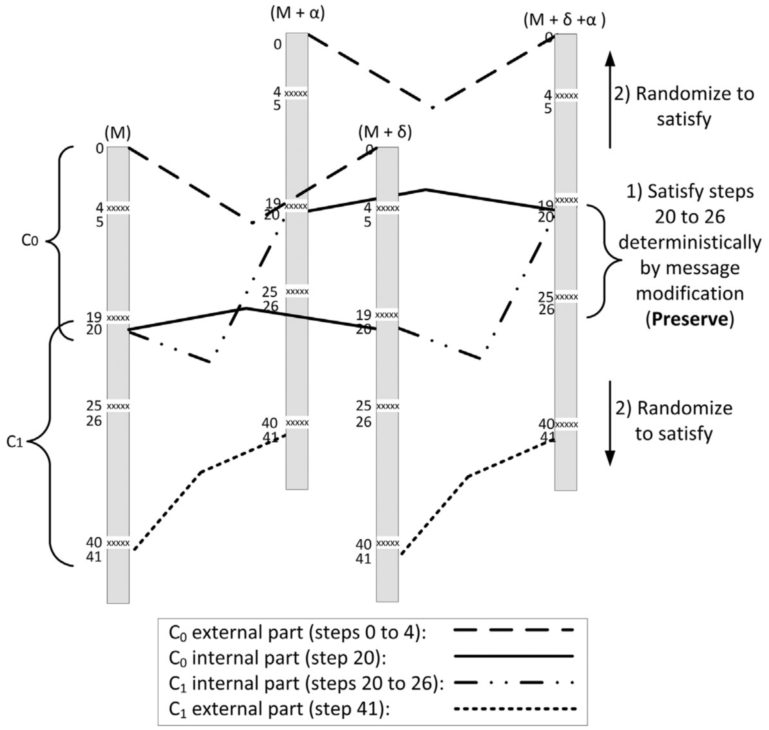
(M + α)
(M + δ + α)
(M)
(M + δ)
0
4
5
19
20
25
26
40
41
xxxxx
C0
C1
2) Randomize to satisfy
1) Satisfy steps 20 to 26 deterministically by message modification (Preserve)
2) Randomize to satisfy
C0 external part (steps 0 to 4):
C0 internal part (step 20):
C1 internal part (steps 20 to 26):
C1 external part (step 41):

Fig. 2. The differential characteristics C_0 and C_1 and how differences propagate between the quartet.

$19 \leqslant i \leqslant 25$ generate chaining states that satisfy the conditions of the internal parts of the differential characteristics C_0 and C_1. Furthermore when these 16-word quartet is expanded, we have both message differentials satisfied. Given the message expansion equation

$$W_i = \sigma_1(W_{i-1}) + W_{i-9} + \sigma_2(W_{i-15}) + W_{i-16}$$

we have

$$W_i = W_{i+16} - \sigma_1(W_{i+15}) - W_{i+7} + \sigma_2(W_{i+1}) \qquad (2)$$

The message differential between $W_i^{m_0}$ and $W_i^{m_1}$, for $0 \leqslant i \leqslant 19$ is specified by C_0, but once the difference at step 4 starts spreading after step 19, the relation between W^{m_0} and W^{m_1} becomes unspecified by C_0. However, we use (2) to specify the relations between $W_{20}^{m_0}$ and $W_{20}^{m_1}$ to $W_{33}^{m_0}$ and $W_{33}^{m_1}$ using the preceding messages. A similar approach is used for branches m_2 and m_3. Our search procedure proceeds as follows:

1. Initialize $W_{19}^{m_0}$ randomly. Set $W_{19}^{m_1} = W_{20}^{m_0} - 0x84001000$, following step 19 difference in C_0.
2. Set $W_{19}^{m_2} = W_{19}^{m_0}$ and $W_{19}^{m_3} = W_{19}^{m_1}$, following step 19 difference in C_1.
3. Initialize $S_{20}^{m_0} = (a_{20}^{m_0}, b_{20}^{m_0}, c_{20}^{m_0}, \ldots, h_{20}^{m_0})$ randomly. Modify it to satisfy the sufficient conditions set at step 20 of the two characteristics C_0 and C_1, as shown in Tables 1 and 2,
4. Compute $S_{20}^{m_1}$, $S_{20}^{m_2}$, and $S_{20}^{m_2}$, such that both $S_{20}^{m_0}$, $S_{20}^{m_1}$ and $S_{20}^{m_2}$, $S_{20}^{m_3}$ follow the difference at step 20 C_0, and both $S_{20}^{m_0}$, $S_{20}^{m_2}$ and $S_{20}^{m_1}$, $S_{20}^{m_3}$ follow the difference at step 20 of C_1:
   - $S_{20}^{m_1}$: $d_{20}^{m_1} = d_{20}^{m_0} - 0x84010000$; $h_{20}^{m_1} = h_{20}^{m_0} - 0x84010000$; the remaining six registers are equal (no difference at C_0).
   - $S_{20}^{m_2}$: $b_{20}^{m_2} = b_{20}^{m_0} - 0x20000000$; $c_{20}^{m_2} = c_{20}^{m_0} + 0x00400100$; $f_{20}^{m_2} = f_{20}^{m_0} - 0x00004000$; $g_{20}^{m_2} = g_{20}^{m_0} + 0x01004000$; the remaining two registers are equal (no difference at C_1).
   - $S_{20}^{m_3}$: $d_{20}^{m_3} = d_{20}^{m_2} - 0x84010000$; $h_{20}^{m_3} = h_{20}^{m_2} - 0x84010000$; the remaining six registers are equal (no difference at C_0).
5. For $20 \leqslant i \leqslant 26$ do:
   - Randomly select $W_i^{m_0}$, compute $W_i^{m_1}$ using Eq. (2) to derive the relation. Compute $W_i^{m_2}$, and $W_i^{m_3}$ according to the internal part of C_1 message difference.
   - Compute $S_{i+1}^{m_0}$, $S_{i+1}^{m_1}$, $S_{i+1}^{m_2}$, and $S_{i+1}^{m_3}$, then check if they satisfy the conditions imposed by C_1, as shown in Table 2, if not go to the previous step.

To satisfy the external parts of C_0 and C_1, we randomly choose $W_{18}^{m_0}$ and $W_{29}^{m_0}$. Then, we compute the corresponding words of the other three branches until step 33. Finally we generate the internal states backward from step 20 to step 0, and forward from step 26 to 41 and check if this quartet leads to a second order collision. If not the procedure is repeated with another $W_{18}^{m_0}$ and $W_{29}^{m_0}$.

4.3. Attack complexity

The two differentials C_0 and C_1 provided at Tables 1 and 2 have probabilities 2^{-33} and 2^{-37}, respectively. Without message modification, the complexity of the attack

Table 3
Example of a second order collision for DHA-256 42-step reduced function, where $m_1 = m_0 + \delta$, $m_2 = m_0 + \alpha$, $m_3 = m_0 + \delta + \alpha$ and $h(m_0) - h(m_1) - h(m_2) + h(m_3) = 0$.

m_0	0xea554805	0x2d8087e1	0x98663092	0xc9162379
	0xfb058245	0xfb39dd00	0x5f46c9f2	0x014d578f
	0xa9a9514b	0x3eb8e329	0xe3ef40dc	0x8cb9edb5
	0xe769f436	0x98a5ea4b	0xbd4cd7e0	0x1bf5513f
$S_0^{m_0}$	0x805bf371	0xe6fea6ce	0xe72f032d	0x66bd003
	0x2693fc3c	0xaf96d4f6	0x1b716395	0x509f78d6
δ	0x00000000	0x00000000	0x00000000	0x00000000
	0x80000000	0x00000000	0x00000000	0x00000000
	0x00000000	0x00000000	0x00000000	0x00000000
	0x00000000	0x00000000	0x00000000	0x00000000
$S_0^{m_1}$	0x605c3371	0xe4fee64e	0xe72f032d	0x066bd003
	0xe695bc3c	0xeb95d4f6	0x1b716395	0x509f78d6
α	0x39556218	0x8297a349	0x75bba37d	0xe56e2cbc
	0xacea2ebc	0x39c2103f	0xb929767d	0xa6c50925
	0x298d9c23	0xb76a30a1	0xa215ebc7	0x695a090a
	0x720282b4	0xd948ffea	0xc005e56c	0x7daf1234
$S_0^{m_2}$	0x627c0c6d	0xcca18f84	0xa7298b00	0x4f2e720a
	0xbcc8914f	0x55448371	0x4d0e0501	0x9d786d82
$S_0^{m_3}$	0x427c4c6d	0xcaa1cf04	0xa7298b00	0x4f2e720a
	0x7cca514f	0x91438371	0x4d0e0501	0x9d786d82

following from the chaining variable differentials would be $2^{-2\times 33}2^{-2\times 37} = 2^{-140}$. Taking into account the nonlinearity of the message expansion and the fact that we control the message difference expansion until step 4 backwards, but not further. Namely, the difference $W'_4 - W_4 = -31$ holds with certainty, but not $W'_3 = W_3$, since

$$W_3 = W_{19} - \sigma_1(W_{18}) - W_{10} + \sigma_2(W_4)$$

Following C_0 message differential, $W'_{18} = W_{18}$ and $W'_{10} = W_{10}$ holds with probability 1. Consequently, Eq. (3) will hold with probability $2^{2\times(-2)}$ which increases the overall probability to 2^{-144}.

$$W_{19} + \sigma_2(W_4) = W'_{19} + \sigma_2(W'_4) \tag{3}$$

Taking into account the message modification on steps 20–26, the complexity is reduced by $2^{2\times 37}$, leading to an attack complexity equal to 2^{70}. Moreover, as noted by Wang [20], when the signed difference is used in the attack and when the feed-forward is specified using modular addition the carry propagation conditions in the starting and ending steps of the function can be ignored. This decreases the complexity by $2^{2\times 12} = 2^{24}$ resulting in an attack complexity of 2^{46}. However, as previously shown in [2,12], the practical attack complexity is usually much lower due to the effect of additional characteristics with the same message differences. This was confirmed by our implementation result and we were able to compute the collision for 42-steps. The attack took about nine hours on a four core 3.0 GHz Intel Xeon processor. In Table 3, we give an example of a DHA-256 second order collision.

4.4. Tweaking the message schedule

As stated in the function specification; one out of the 64 expanded message words is fed twice in each step to update two different state variables. Subsequently, when message modification is used, we try to search for one 32-bit message word that would satisfy the conditions imposed on these two states. However, if two different message words are used in each step; one would be searching for two words that should satisfy conditions for the current and another later or previous steps, which is not guaranteed to work and may need more complicated modification techniques because of the increased probability of contradiction. Also introducing a difference at a given message word will result in two differentials at two different steps. This increases the complexity of a given characteristic and reduces the chances of finding two compatible paths.

5. Conclusion

In this paper, we analysed the security of DHA-256 hash function with respect to second order differential attacks. We devised a practical distinguisher for 42-step DHA-256 compression function and proposed a message scheduling tweak that reduces the function's vulnerability to this kind of attacks. To validate our work, we fully implemented the attack and provided particular colliding quartet for the function.

References

[1] M. Bellare, P. Rogaway, Optimal asymmetric encryption, in: A.D. Santis (Ed.), EUROCRYPT, in: Lecture Notes in Computer Science, vol. 950, Springer, 1995, pp. 92–111.

[2] A. Biryukov, M. Lamberger, F. Mendel, I. Nikolić, Second-order differential collisions for reduced SHA-256, in: D.H. Lee, X. Wang (Eds.), Advances in Cryptology – ASIACRYPT 2011, in: Lecture Notes in Computer Science, vol. 7073, Springer, 2011, pp. 270–287.

[3] A. Biryukov, I. Nikolić, A. Roy, Boomerang attacks on BLAKE-32, in: A. Joux (Ed.), Fast Software Encryption, in: Lecture Notes in Computer Science, vol. 6733, Springer, 2011, pp. 218–237.

[4] R. Canetti, O. Goldreich, S. Halevi, The random oracle methodology, revisited, Journal of the ACM (JACM) 51 (July 2004) 557–594.

[5] S. Chang, R. Perlner, W.E. Burr, M. Turan, J. Kelsey, S. Paul, L.E. Bassham, Third-round report of the SHA-3 cryptographic hash algorithm competition, http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7896.pdf, 2012.

[6] J.-S. Coron, J. Patarin, Y. Seurin, The random oracle model and the ideal cipher model are equivalent, in: D. Wagner (Ed.), CRYPTO, in: Lecture Notes in Computer Science, vol. 5157, Springer, 2008, pp. 1–20.

[7] IAIK Krypto Group, Preliminary analysis of DHA-256, Cryptology ePrint Archive, Report 2005/398, 2005, http://eprint.iacr.org/2005/398.

[8] A. Kircanski, Y. Shen, G. Wang, A.M. Youssef, Boomerang and slide-rotational analysis of the SM3 hash function, in: L.R. Knudsen, H. Wu (Eds.), Selected Areas in Cryptography, in: Lecture Notes in Computer Science, vol. 7707, Springer, 2013, pp. 304–320.

[9] X. Lai, Higher order derivatives and differential cryptanalysis, in: R. Blahut, D. Costello Jr., U. Maurer, T. Mittelholzer (Eds.), Communications and Cryptography, Kluwer, 1994, pp. 227–233.

[10] M. Lamberger, F. Mendel, Higher-order differential attack on reduced SHA-256, Cryptology ePrint Archive, Report 2011/037, 2011, http://eprint.iacr.org/2011/037.

[11] J. Lee, D. Chang, H. Kim, E. Lee, D. Hong, A new 256-bit hash function DHA-256 – enhancing the security of SHA-256, NIST - First Cryptographic Hash Workshop, November 2005.

[12] G. Leurent, A. Roy, Boomerang attacks on hash function using auxiliary differentials, in: O. Dunkelman (Ed.), CT-RSA, in: Lecture Notes in Computer Science, vol. 7178, Springer, 2012, pp. 215–230.

[13] F. Mendel, T. Nad, Boomerang distinguisher for the SIMD-512 compression function, in: D.J. Bernstein, S. Chatterjee (Eds.), INDOCRYPT, in: Lecture Notes in Computer Science, vol. 7107, Springer, 2011, pp. 255–269.

[14] F. Mendel, T. Nad, M. Schläffer, Finding SHA-2 characteristics: Searching through a minefield of contradictions, in: D.H. Lee, X. Wang (Eds.), ASIACRYPT, in: Lecture Notes in Computer Science, vol. 7073, Springer, 2011, pp. 288–307.

[15] F. Mendel, T. Nad, M. Schläffer, Improving local collisions: New attacks on reduced SHA-256, in: T. Johansson, P.Q. Nguyen (Eds.), EUROCRYPT, in: Lecture Notes in Computer Science, vol. 7881, Springer, 2013, pp. 262–278.

[16] S. Murphy, The return of the cryptographic boomerang, IEEE Transactions on Information Theory 57 (4) (2011) 2517–2521.

[17] Y. Sasaki, Boomerang distinguishers on MD4-based hash functions: First practical results on full 5-pass HAVAL, in: A. Miri, S. Vaudenay (Eds.), Selected Areas of Cryptography, in: Lecture Notes in Computer Science, vol. 7118, Springer, 2011, pp. 1–18.

[18] Y. Sasaki, L. Wang, Y. Takasaki, K. Sakiyama, K. Ohta, Boomerang distinguishers for full HAS-160 compression function, in: G. Hanaoka, T. Yamauchi (Eds.), Advances in Information and Computer Security, in: Lecture Notes in Computer Science, vol. 7631, Springer, 2012, pp. 156–169.

[19] D. Wagner, The boomerang attack, in: L.R. Knudsen (Ed.), Fast Software Encryption, in: Lecture Notes in Computer Science, vol. 1636, Springer, 1999, pp. 156–170.

[20] X. Wang, Y.L. Yin, H. Yu, Finding collisions in the full SHA-1, in: V. Shoup (Ed.), CRYPTO, in: Lecture Notes in Computer Science, vol. 3621, Springer, 2005, pp. 17–36.

[21] J. Zhong, X. Lai, Preimage attack on reduced DHA-256, Journal of Information Science and Engineering 27 (2011) 1315–1327.