Analysis of a SHA-256 Variant*

Hirotaka Yoshida[1] and Alex Biryukov[2]

[1] Systems Development Laboratory, Hitachi, Ltd.,
1099 Ohzenji, Asao-ku, Kawasaki-shi, Kanagawa-ken, 215-0013 Japan
hyoshida@sdl.hitachi.co.jp
[2] Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC,
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
abiryuko@esat.kuleuven.ac.be

Abstract. SHA-256 is a cryptographic hash function which was proposed in 2000 as a new generation of SHA functions and was adopted as FIPS standard in 2002. In this paper we will consider a SHA-256 variant and a SHACAL-2 variant in which every arithmetic addition is replaced by XOR operation. We call the SHA-256 variant SHA-2-XOR and the SHACAL-2 variant SHACAL-2-XOR respectively. We will present a differential attack on these constructions by using one-round iterative differential characteristics with probability 2^{-8} we identified. Our result shows that SHACAL-2-XOR with up to 31 rounds out of 64 has a weakness of randomness and that SHA-2-XOR with up to 34 rounds has a weakness of pseudo-collision resistance. Using the 31-round distinguisher, we present an attack on SHACAL-2-XOR with up to 32 rounds. We also show that no 2-round iterative patterns with probability higher than 2^{-16} exist.

Keywords: SHA-256,SHA-2-XOR, SHACAL-2-XOR, Differential cryptanalysis, Pseudo-collision resistance, Iterative patterns.

1 Introduction

A cryptographic hash function is an algorithm that takes input strings of arbitrary (typically very large) length and maps these to short fixed length output strings. The progress in cryptanalysis of cryptographic hash functions has been quite slow until very recently, the cryptographic community has been surprised at the progress of cryptanalysis of hash functions, such as an attack on MD5 [23] for finding collisions and an attack with a new strategy on SHA-0 [2, 3] and an attack for finding multi-collisions. However, these techniques are not applicable to SHA-256 due to its more complex message schedule and round function.

SHA-256 is a cryptographic hash function which was proposed in 2000 as a new generation of SHA functions and was adopted as FIPS standard in 2002 [18]. SHA-256 is constructed from MD(Merkle-Damgård) -construction and Davis-Meyer mode. The compression function of SHA-256 has 64 rounds, two kinds of

* This work was supported in part by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government.

B. Preneel and S. Tavares (Eds.): SAC 2005, LNCS 3897, pp. 245–260, 2006.

non-linear functions, cyclic rotations, and round-dependent constants. The hash value calculated by SHA-256 is 256 bits long.

The function obtained from the compression function of SHA-256 by removing the feed-forward operation of the Davis-Meier mode is invertible. It was proposed for use as a block cipher by Handschuh and Naccache and named SHACAL-2 [12]. The block cipher was selected as one of the NESSIE finalists. In cryptanalysis of SHACAL-2, there have been several attacks on its reduced versions [14, 22], but with time complexities around 2^{500} for 32 or less rounds.

Although several works have discussed the security of SHA-256[11] and reported interesting differential properties of several consecutive round functions [13], no weakness has been demonstrated for SHA-256 or any SHA-256 variant so far. In this paper we will consider a SHA-256 variant and a SHACAL-2 variant in both of which ADD operations are replaced by XOR operations. We call the SHA-256 variant SHA-2-XOR and the SHACAL-2 variant SHACAL-2-XOR respectively. We will present a differential attack [5] on these ciphers by identifying iterative differential characteristics. We will show how to distinguish the SHACAL-2-XOR from a random permutation. Our result will show that SHACAL-2-XOR with up to 31 rounds has a weakness of randomness and that SHA-2-XOR with up to 34 rounds has a weakness of pseudo-collision resistance. In addition to that, it will also show a property that SHA-2-XOR with up to 31 rounds has a weakness in certain collision resistance we will define.

Hereafter we introduce three kinds of resistance of hash functions for the motivation of our approach in the cryptanalysis of SHA-256: near-collision resistance, pseudo-collision resistance, and randomness.

The importance of the first two requirements is related to collision resistance. *Near-collision* resistance is resistance against attacks finding a pair of hash values which differ in only small number of bit positions. Near-collisions of the SHA-0 hash function have been found, which is an undesirable property [2] for a hash function. In fact, there has been presented a strategy to convert near-collisions into full-collisions [1]. Therefore near-collision resistance is crucial for the collision resistance. *Pseudo-collision* resistance is resistance against finding a collision obtained from more relaxed condition that different initial vectors can be chosen. Pseudo-collision resistance has a particular importance for a hash function constructed by the MD-construction because in this case pseudo-collision resistance for the hash function can be translated into collision resistance for its compression function. The theory of the MD-construction, on which the security of many popular hash functions rely, does not guarantee collision resistance for a hash function without pseudo-collision resistance for its compression function [10]. Recently, a situation where pseudo-collisions could become practical has been considered [16].

Pseudo-randomness of a function is its indistinguishability from a random function. This resistance has a particular importance in some existing applications where one of the requirements for the hash function is randomness. Recently, the strongest version of the HAVAL hash function (in encryption mode) was shown to be non-random [24].

Although in the past these three types of resistance have received less attention than the collision resistance, we expect that situation will change in the near future.

The outline of this paper is as follows. In Section 2, we give a brief description of the SHA-2 algorithm published in [18]. In Section 3 we study the known results on cryptanalysis of the SHA-256 algorithm and the SHACAL-2 algorithm. In Section 4, we present our differential attack on the SHA-2-XOR and SHACAL-2-XOR identifying iterative characteristics. Our conclusions are given in Section 5.

2 Description of the SHA-256 Hash Function and the SHACAL-2 Block Cipher

In this section, we give a brief description of the SHA-256 hash function and the SHACAL-2 block cipher, which is sufficient to understand the concepts introduced in this paper. For a full description of SHA-256 we refer to [18].

SHA-256 is a hash function that is based on the well-known Davies-Meyer construction of hash functions ([17], p. 341). The variable-length message M is divided into 512-bit blocks $M_0, M_1, \ldots, M_{n-1}$. The 256-bit hash value V_n is then computed as follows:

$$V_0 = IV;\ V_{s+1} = \texttt{compress}(V_s, M_s) = E_{M_s}(V_s) + V_s\ for\ 0 \leq s < n,$$

where `compress` is the compression function, IV is a fixed initial value and $E_K(X)$ is the block cipher, SHACAL-2. The function $E_K(X)$ is an iterated design that only uses simple operations on 32-bit words. The 256-bit input V_j is loaded into 8 registers (A, B, C, D, E, F, G, H) and the 512-bit message block is divided into 16 words of 32 bits $(W_0 \ldots W_{15})$ and these words are expanded to a sequence of 64 words through the message schedule:

$$\begin{aligned}
\sigma_0(X) &= ROTR^7(X) \oplus ROTR^{18}(X) \oplus SHR^3(X);\\
\sigma_1(X) &= ROTR^{17}(X) \oplus ROTR^{19}(X) \oplus SHR^{10}(X);\\
W_t &= \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}
\end{aligned}$$

where $ROTR^n$ is right rotation by n bits. SHACAL-2 encrypts the initial value using this sequence as a key.

The 8 registers are updated through a number of rounds. One round of the compression function is depicted in Fig. 1. The SHA-256 compression function consists of 64 rounds. Every round function has arithmetic addition, a round-dependent constant K_i, two linear functions Σ_0, Σ_1, and two non-linear functions CH, MJ.

$$\begin{aligned}
CH(X, Y, Z) &= (X \wedge Y) \oplus (\overline{X} \wedge Z);\\
MJ(X, Y, Z) &= (X \wedge Y) \oplus (Y \wedge Z) \oplus (Z \wedge X);\\
\Sigma_0(X) &= ROTR^2(X) \oplus ROTR^{13}(X) \oplus ROTR^{22}(X);\\
\Sigma_1(X) &= ROTR^6(X) \oplus ROTR^{11}(X) \oplus ROTR^{25}(X),
\end{aligned}$$

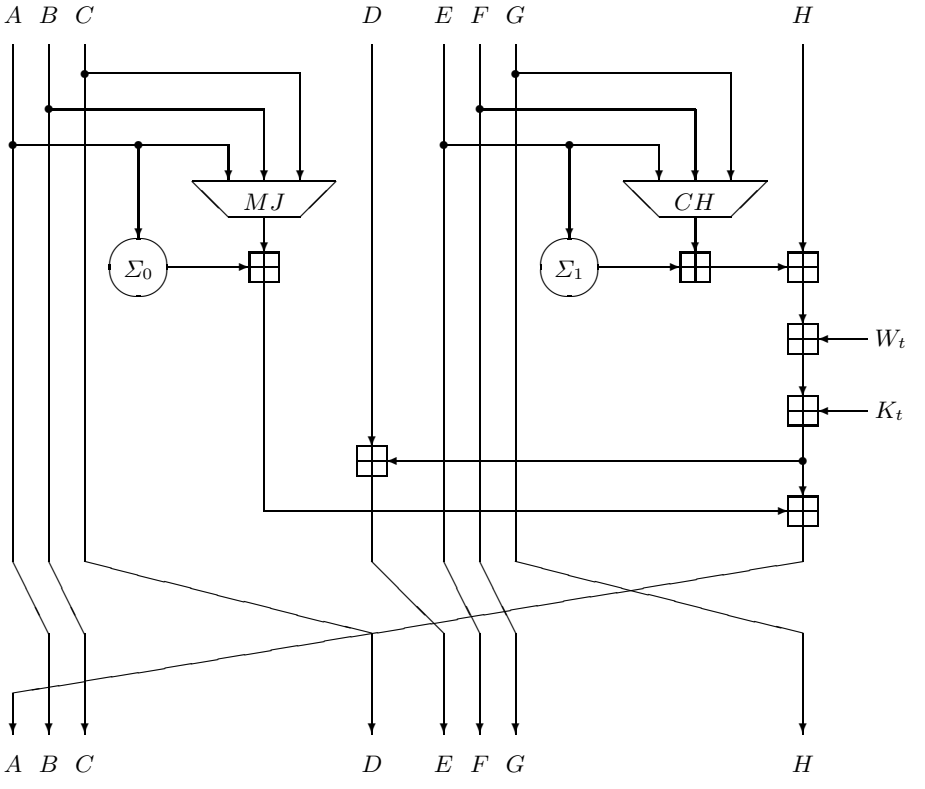

Fig. 1. Round function for SHA-256

where $\overline{X}$ is bitwise complement of X. The t-th round of the compression function updates the 8 registers using the word W_t and the constant K_i as input. The compression function updates the 8 registers according to the following algorithm:

$$\begin{aligned} T1_t(E_t, F_t, G_t, H_t, K_t, W_t) &= H_t + \Sigma_1(E_t) + CH(E_t, F_t, G_t) + K_t + W_t\,; \\ T2_t(A_t, B_t, C_t) &= \Sigma_0(A_t) + MJ(A_t, B_t, C_t)\,; \\ H_{t+1} &= G_t; G_{t+1} = F_t; F_{t+1} = E_t; E_{t+1} = D_t + T1_t\,; \\ D_{t+1} &= C_t; C_{t+1} = B_t; B_{t+1} = A_t; A_{t+1} = T1_t + T2_t. \end{aligned}$$

2.1 Our Variant of SHA-256

In our analysis, we simplify SHA-256 and SHACAL-2 by replacing all the arithmetic addition used in its round function by the XOR operation. This analysis tells us how much the carry propagation caused by the arithmetic addition affect the security of the cipher. It is also interesting for designers to investigate the security of an arithmetic-addition free hash function, because such a hash function has an advantage in its hardware implementation due to a lower gate count.

3 Previous Work

3.1 A Study on the Known Attacks on a Reduced Version of SHACAL-2

In the literature, two kinds of attacks on SHACAL-2 have been demonstrated. In [14], it was shown that the impossible differential attack [4] is applicable to the reduced 30-round SHACAL-2 with a time complexity $2^{495.1}$ and a memory complexity $2^{14.5}$. In [22] it has been shown that the differential-linear attack is applicable to the reduced 32-round SHACAL-2 with a complexity $2^{504.2}$ and a memory complexity $2^{48.4}$ which is the best attack so far. In the table 1, we list the best previous result and our result[1].

Table 1. The best previous result and our result

Attack type	#R	Data	Time	Memory
Impossible Differential attack on SHACAL-2[14]	30	744CP	$2^{495.1}$	$2^{14.5}$
Differential-linear attack on SHACAL-2[22]	32	$2^{43.4}$CP	$2^{504.2}$	$2^{48.4}$
Related-Key Rectangle attack on SHACAL-2[15]	37	$2^{43.2}$RK-CP	$2^{484.95}$	$2^{238.16}$
Distinguisher attack on SHACAL-2-XOR in this paper	31	2^{248}CP	2^{248}	
Differential attack on SHACAL-2-XOR in this paper	32	$2^{243.3}$CP	$2^{246.3}$	2^{22}

#R: # of rounds, CP: Chosen Plaintexts, RK-CP: Related-Key Chosen Plaintexts, Time: Encryption units, Memory: Bytes of memory

3.2 A Study on the Known Results on SHA-256

What has been known as results on cryptanalysis of the SHA-256 algorithm so far are several properties related to resistance of the function against the known attacks [11, 13] where none of the attacks have demonstrated any weakness in SHA-256 or any SHA-256 variant.

Hereafter we study the known results on resistance of SHA-256 against a theoretical attack on SHA-0[8] which have been very important results so far in the following sense: some strong attacks on the SHA algorithms have been developed by improving the attack. Two interesting strategies significantly reducing the complexity in the attack found collisions or near-collisions for the SHA-0 hash functions [2, 3].

We explain the procedure of the attack which is divided into two steps. This attack first finds a sequence of differences which is called local collision with a high-probability. An attacker introduces a 1-bit difference into one message word and then for the following rounds the attacker also introduced differences into the following message words so that the differences in the registers are canceled out, which results in a local collision with several rounds. As a result, the attacker has

[1] This distinguisher uses a differential characteristic for 31 rounds, it can be made more efficient by relaxing conditions of the final rounds. This is done for the differential attack which improves complexity of the attack and allows to recover the secret key bits.

obtained Recent works have given high-probabilities for the local collisions they identified, a probability 2^{-66} in [11], a better probability 2^{-39} in [13]. What we need to take into account in this step is that the attacker can choose the differences he injects whatever he likes, which means he does not care about the message schedule.

Secondly the attack analyzes the message schedule in an attempt to find two messages such that the message schedule will result in the collision obtained in the first step. However, it has been difficult to carry out this step for all the SHA-algorithms except for SHA-0 because of the large influence of the message schedule with respect to difference propagation.

4 Differential Cryptanalysis of SHA-2-XOR and SHACAL-2-XOR

4.1 Search for One-Round Iterative Differential Characteristics

We will search for one round iterative differential characteristics for SHA-2-XOR. We will first determine the constraints which an iterative characteristic should satisfy. Then we will develop an efficient algorithm to find all the differential characteristics satisfying the constraints and find one with the highest probability.

Let us denote the value in the register A at time t by A_t and the difference in this register at time t by dA_t. The t-th round changes the value A_t to A_{t+1} in the register A. The one-round iterative translates into conditions that in each register the differences at time t and at time $t+1$ are the same: $dA_{t+1} = dA_t, dB_{t+1} = dB_t, dC_{t+1} = dC_t, dD_{t+1} = dD_t, dE_{t+1} = dE_t, dF_{t+1} = dF_t, dG_{t+1} = dG_t, dH_{t+1} = dH_t$.

Our purpose here is to translate the constraints into the conditions with differences only at time t. There are 6 registers, in each of which value at time $t+1$ is determined by only one register value at time t. This builds the following simple relations between the differences at time t and the differences at time $t+1$: $dB_{t+1} = dA_t, dC_{t+1} = dB_t, dD_{t+1} = dC_t, dF_{t+1} = dE_t, dG_{t+1} = dF_t, dH_{t+1} = dG_t$. From these relations, 6 constraints $dB_{t+1} = dB_t$, $dC_{t+1} = dC_t$, $dD_{t+1} = dD_t$, $dF_{t+1} = dF_t$, $dG_{t+1} = dG_t$, $dH_{t+1} = dH_t$ are equivalent to the following conditions: $dA_t = dB_t = dC_t = dD_t, dE_t = dF_t = dG_t = dH_t$.

Now we have two remaining constraints $dA_{t+1} = dA_t$, $dE_{t+1} = dE_t$ to transform. We introduce several functions dCH, dMJ,$dT1_t$, $dT2_t$ each of which is the output difference of a sub-function used in the round function. These functions are defined as follows:

$$
\begin{aligned}
dCH &= CH(X,Y,Z) \oplus CH(X',Y',Z'),\\
dMJ &= MJ(X,Y,Z) \oplus MJ(X',Y',Z'),\\
dT1_t &= T1_t(E_t,F_t,G_t,H_t,K_t,W_t) \oplus T1_t(E'_t,F'_t,G'_t,H'_t,K_t,W_t),\\
dT2_t &= T2_t(A_t,B_t,C_t) \oplus T2_t(A'_t,B'_t,C'_t).
\end{aligned}
$$

We rewrite the non-linear functions CH, MJ in terms of their input values and input differences. Let's denote the input differences to the non-linear functions by

$dX_t = X_t \oplus X'_t$ for two input values X_t, X'_t. dCH is calculated as the following:

$$dCH = ((Y \oplus Z) \wedge dX) \oplus (X \wedge dY) \oplus (\overline{X} \wedge dZ) \oplus (dX \wedge dY) \oplus (dX \wedge dZ) \quad (1)$$

In particular, if all the differences to CH are equal, $dX = dY = dZ$, then

$$dCH = (Y \oplus \overline{Z}) \wedge dX. \quad (2)$$

dMJ is calculated as the following:

$$dMJ = MJ(dX, dY, dZ) \oplus ((Y \oplus Z) \wedge dX) \oplus ((Z \oplus X) \wedge dY) \oplus ((X \oplus Y) \wedge dZ). \quad (3)$$

In particular, if all the differences to MJ are equal, $dX = dY = dZ$, then

$$dMJ = dX \quad (4)$$

This tells an important property on MJ that this function behaves linearly if all the input differences are equal[2].

By using the formulas (2),(4) and the constraints obtained so far, $dT1_t$, $dT2_t$, dA_{t+1}, dE_{t+1} are calculated as follows:

$$dT1_t = dE_t \oplus \Sigma_1(dE_t) \oplus ((F_t \oplus \overline{G_t}) \wedge dE_t) \quad (5)$$

$$dT2_t = \Sigma_0(dA_t) \oplus dMJ(A_t, B_t, C_t) = \Sigma_0(dA_t) \oplus dA_t \quad (6)$$

$$dA_{t+1} = dT1_t \oplus dT2_t$$

$$dE_{t+1} = dD_t \oplus dT1_t = dA_t \oplus dT1_t$$

Therefore, the remaining constraints $dA_{t+1} = dA_t$, $dE_{t+1} = dE_t$ are equivalent to the following two conditions:

$$dA_t = dT1_t \oplus dT2_t$$

$$dE_t = dA_t \oplus dT1_t.$$

We can from now on omit the time indexes of differences, e.g. $dA_t = dA$. Then these two conditions are equivalent to following conditions:

$$dA = dT1_t \oplus dE \quad (7)$$

$$dE = dT2_t. \quad (8)$$

By the formula (6), the condition (8) is calculated as follows:

$$dE = \Sigma_0(dA) \oplus dA.$$

[2] This property has been noticed previously, for example see [11].

By the formula (5), the condition (7) is calculated as follows:

$$dA = dE \oplus \Sigma_1(dE) \oplus ((F \oplus \overline{G}) \wedge dE) \oplus dE.$$

Value $F \oplus \overline{G}$ can be considered to be some random value X. This condition is equivalent to the following condition:

$$dA = \Sigma_1(dE) \oplus (X \wedge dE).$$

We have determined the conditions for the existence of iterative. We now are interested in those iterative characteristic that have high probabilities. For an iterative with differences dA, dE, if some register inputs make this condition hold, they also make the other conditions hold. Therefore, we pay a probability only for this condition to hold. We see that we have to pay probability for this equation at bit position j to hold if and only if $dE^{(j)}$ is equal to 1. In particular, an iterative where Hamming weight of dE is the smallest has the best probability. This discussion leads us to the following theorem.

Theorem 1. *For SHA-2-XOR, a differential characteristic with input differences (dA,dB,dC,dD,dE,dF,dG,dH) is a one round iterative if and only if for some 32-bit value X, the input differences dA, dE satisfy the following:*

$$dA = \Sigma_1(dE) \oplus (X \wedge dE). \tag{9}$$
$$dE = \Sigma_0(dA) \oplus dA. \tag{10}$$

If this condition holds, the other differences in the characteristic are determined by dA and dE as follows:

$$dB = dA, dC = dA, dD = dA, dF = dE, dG = dE, dH = dE.$$

Furthermore, iterative where the weight of dE is the smallest has the best probability.

4.2 The Search Algorithm

We have to design an algorithm for practical use of the theorem. By substituting the second condition into the first one, we obtain the following:

$$dA = \Sigma_1(\Sigma_0(dA) \oplus dA) \oplus (X \wedge (\Sigma_0(dA) \oplus dA)).$$

It is sufficient for us to search for dA's which make this equation solvable in terms of X. Looking at this equation per bit leads us to consider a 1-bit equation $I = X \wedge R$. We consider what is the condition on I that the equation has a solution $X = X_0$, in each of two cases, $R = 0, R = 1$. In the case of R equal to 1, there always exists a solution. In the case of R equal to 0, there exists a solution if and only if I is equal to 0. Based on this consideration, now we can develop the following algorithm shown in Table 2 where for a bit string V, its value at bit position j is denoted by $V^{(j)}$.

Table 2. The search algorithm

Step1: Choose a 32-bit value, dA
Step2: Compute $R = \Sigma_0(dA) \oplus dA$.
Step3: Set u to be 0.
Step4: For j=0 to 31 do:
If $R^{(j)}$ is equal to 0, do
Compute $I^{(j)} = (\Sigma_1(\Sigma_0(dA) \oplus dA) \oplus dA)^{(j)}$
If $I^{(j)}$ is equal to 1, increase u by 1.
Otherwise, do nothing.
Step5: If u is equal to 0, then output dA.
Step6: If all possible value for dA have been chosen, then end.
Otherwise go to step1.

4.3 The Best One-Round Iterative Differential Characteristics

The algorithm we designed has identified *all* one round iterative characteristics for SHA-2. The running time was 30 min. Table 3 shows all the one-round iterative differential characteristic with the best probability 2^{-8}.

Table 3. One round iterative differential characteristic with the best probability 2^{-8}

$dA = dB = dC = dD$	$dE = dF = dG = dH$
3b3b3b3b	c0c0c0c0
67676767	18181818
76767676	81818181
9d9d9d9d	60606060
b3b3b3b3	0c0c0c0c
cecececе	30303030
d9d9d9d9	06060606
ecececec	03030303

It was confirmed that one of the best iterative with $dA = b3b3b3b3$, $dE = 0c0c0c0c$ has an experimental probability $259/(2^{16})$ which is around 2^{-8}. We can theoretically tell exactly what happens in one round. The only place where probabilities are paid is the place where the CH function is applied. The difference at the input of CH, 0c0c0c0c becomes 08080808 at the output with a probability 2^{-8}, which is calculated using the following differential property of CH per bit:

$$CH(0,0,0) = 0$$
$$CH(1,1,1) = 0/1 \text{ with probability } 1/2.$$

Note that the eight iterative patterns given in Table 3 are cyclic rotations of the same pattern. In the following section, we show that no 2-round iterative patterns better than a concatenation of two best one-round iteratives exist.

4.4 Search for 2-Round Iterative Differential Characteristics

We search for two-round iterative differential characteristics for SHA-2-XOR. However, we will show that no 2-round iterative patterns with probability higher than 2^{-16}. We first determine the constraints which an iterative pattern should satisfy. Let's denote the value(the difference) in the register A at time t by $A_t(dA_t)$. The t-th first rounds change the value A_t to A_{t+1} in the register A. The constraints are translated into conditions that in each register its difference at time t and its difference at time $t+2$ are the same: $dA_{t+2} = dA_t, dB_{t+2} = dB_t, dC_{t+2} = dC_t, dD_{t+2} = dD_t, dE_{t+2} = dE_t, dF_{t+2} = dF_t, dG_{t+2} = dG_t, dH_{t+2} = dH_t$.

Our purpose here is to translate the constraints into the conditions with differences only at time t. There are 4 registers, in each of which value at time $t+1$ is determined by only one register value at time t. This builds the following simple relations between the differences at time t and the differences at time $t+1$: $dC_{t+2} = dA_t, dD_{t+2} = dB_t, dG_{t+2} = dE_t, dH_{t+2} = dF_t$. From these relations, 4 constraints $dC_{t+2} = dC_t$, $dD_{t+2} = dD_t$, , $dG_{t+2} = dG_t$, $dH_{t+2} = dH_t$ equivalent to the following conditions: $dC_t = dA_t, dD_t = dB_t, dG_t = dE_t, dH_t = dF_t$.

Now we have 4 remaining constraints $dF_{t+2} = dF_t$, $dE_{t+2} = dE_t$, $dA_{t+2} = dA_t$, $dB_{t+2} = dB_t$ from which we can derive the following four conditions:

$$dB_t \oplus \Sigma_1 dE_t = dCH(E_t, F_t, G_t), \tag{11}$$

$$dA_t \oplus \Sigma_1 dF_t = dCH(E_{t+1}, E_t, F_t), \tag{12}$$

$$dF_t \oplus \Sigma_0 dA_t = dMJ(A_t, B_t, C_t), \tag{13}$$

$$dE_t \oplus \Sigma_0 dB_t = dMJ(A_{t+1}, A_t, B_t) \tag{14}$$

In our case, the conditions $dA_{t+1} = dB_t$, $dE_{t+1} = dF_t$ hold. Therefore we need to know what is the differential property of non-linear functions with some conditions on their input differences which is given in Table 4.

Table 4. A differential property on non-linear functions

$dX = dZ$	dY	dCJ	dMJ
0	0	0	0
0	1	0/1	0/1
1	0	0/1	0/1
1	1	0/1	1

We assume that there is an iterative with differences (dA,dB,dC,dD,dE,dF, dG,dH) have a probability at least 2^{-16}. Let us define α, β as follows:

$$\alpha = dCH(E_t, F_t, G_t) \oplus dCH(E_{t+1}, E_t, F_t).$$

$$\beta = dMJ(A_t, B_t, C_t) \oplus dMJ(A_{t+1}, A_t, B_t).$$

We know $\mathrm{Ham}(\alpha) \leq 8$ by studying (11) (12) and Table 4. We can assume $\mathrm{Ham}(\alpha)$ is more than 0, otherwise the search is reduced to the search for one-round

iterative pattens. We also know the following condition on $dE \oplus dF$ that holds for any bit position j,

$$\alpha^{(j)} = 0 \Longrightarrow (dE_t \oplus dF_t)^{(j)} = 0.$$

Hence, the number of possible values for $dE_t \oplus dF_t$ is $2^{Ham(\alpha)}$. By adding (11) and (12) we have the following:

$$dA_t \oplus dB_t = \Sigma_1(dE_t \oplus dF_t) \oplus \alpha \tag{15}$$

On the other hand, we have the following condition by adding (12) and (13),

$$dF_t \oplus dE_t \oplus \Sigma_0(dA_t \oplus dB_t) = \beta$$

Finally we obtain the following condition:

$$dF_t \oplus dE_t \oplus \Sigma_0(\Sigma_1(dE_t \oplus dF_t) \oplus \alpha) = \beta \tag{16}$$

Now we can compute $dA \oplus dB$ and β from α. From the discussion above, we also obtain the following property that holds for any bit position j,

$$(dA_t \oplus dB_t)^{(j)} = 0 \Longrightarrow \beta^{(j)} = 0.$$

However, it was confirmed that none of computed $dA_t \oplus dB_t$ and β satisfy this property. This was done with 2^{32} possible values for α. The total complexity is $2^{32+Ham(\alpha)} = 2^{40}$ elemental computations.

4.5 Pseudo-collision Attack on SHA-2-XOR Using Iterative Differential Characteristic

We present attacks on SHA-2-XOR and SHACAL-2-XOR using iterative differential characteristic we identified. We present two kinds of attacks on SHA-2-XOR.

By definition, to find a pseudo-collision, an attacker can inject differences both into the message schedule and registers. The attacker would require a complexity 2^{128} to find a pseudo-collision for a ideal hash function. We obtain a 15-round iterative with a probability 2^{-120} by concatenating one of the best one-round iterative we identified. This leads to an attack finding a pseudo-collision with a complexity 2^{120} for the 15-round SHA-2-XOR.

Our attack suggests a security model where an attacker can inject differences only into registers. Taking into account the feed-forward operation of the Davis-Meyer mode, to find a collision means to find a differential characteristic for the underlying block cipher where an input difference and an output difference of are same. In the ideal case, if both of an input difference and an output difference are fixed, then the probability that a plaintext pair with the input difference results in the output difference is 2^{-256}. However, SHA-2-XOR with 31 rounds has a probability 2^{-248} which means that 31 rounds of this hash function does not behave as a random hash function.

4.6 Differential Attack on 32-Round SHACAL-2-XOR

As for SHACAL-2-XOR, we can build a 31-round characteristic with a probability 2^{-248} concatenating one of the best iterative differential characteristics we identified. This shows SHACAL-2-XOR with 31 rounds is distinguished from a random permutation. We now attack SHACAL-2-XOR with 32 rounds by using the 30-round differential characteristic. Our goal here is to find the 32-bit key W_{31}. Let δ be the input difference in the 30-round characteristic(e.g. dA_0=dB_0=dC_0=dD_0= `3b3b3b3b`, dE_0=dF_0=dG_0=dH_0= `c0c0c0c0`). We denote a plaintext P at time t by P_t and the value of P_t in the register A by A_t. We denote the difference between a pair of plaintexts (P, P^*) at time t by Δ_t and the difference of d_t in the register A by dA_t. Let (P, P^*) be a pair of plaintexts with the difference δ:$\Delta_0 = \delta$ The pair of corresponding ciphertexts is (P_{32}, P^*_{32}). There are two steps to perform our attack, data collection step, data analysis step. In the data collection step, we encrypt $2^{240} \cdot 10$ plaintext pairs. Then we collect only $2^{16} \cdot 10$ pairs needed for the next step by checking if the corresponding ciphertexts pairs satisfy certain conditions. Let us see what this condition looks like. For the right pairs, the condition $\Delta_{30} = \delta$ holds. For the last 2 rounds, we observe how this difference behaves in case of not paying any probability. Even in this case, in the 4 registers, the differences at time 32 are determined uniquely by the differences at time 30: $dC_{32} = dA_{30}$, $dD_{32} = dB_{30}$, $dG_{32} = dE_{30}$, $dH_{32} = dF_{30}$. By studying how non-linear functions increase the uncertainty of differences, we can see there are 2^{16} candidates for the differences in the other 4 registers at time 32: (dA_{32}, dB_{32},dE_{32},dF_{32}).

In the data analysis step, we find 8bits of 32-bit key W_{31} using 2^8 counters. Each pair suggests one key therefore one counter is 2^8 in average, while the counter for the correct key bits is $2^8 \cdot 10$. This enables us to detect the correct key bits. Using another three iterative characteristics we identified, we can find another 24 bits of W_{32}.

The time complexity of this attack is $2^{246.3}$(= 2^{240}·10·2·4) 32-round SHACAL-2-XOR encryptions and the data complexity of this attack is $2^{243.3}$ chosen plaintexts which are immediately discarded leaving only 2^{17} for the analysis step.

4.7 Improvement of the Pseudo-collision Attack

In the previous section, we identified one-round iterative differential characteristics. Using the best ones with the probability 2^{-8}, we attacked 15 rounds of SHA-2-XOR regarding pseudo-collision resistance. Here we will improve this result and add more rounds.

In the pseudo-collision attack model, the attacker choose any element from the set $I_{all} = \{0,1\}^{256} \times \{0,1\}^{512}$, which is taken as input to the compression function. The main idea in our improvement is to use a subset of I_{all} denoted by I_{sub} for which better probabilities for many rounds are obtained. This idea was already indicated in [19] where it is pointed out that the attacker can choose the message so that the first several rounds follow the characteristic with probability 1. It is quite natural to consider this idea in cryptanalysis of hash functions. Recently, this idea was effectively used in the attacks in [23].

To realize this idea in practice, the attacker first randomly choose an input from I_{all} and then modifies it in a way that certain condition on the register values $E_t, F_t, G_t, t = 0, 1, \ldots, 17$ in the Table 5 is satisfied. Using the resulting the set of modified inputs, we do not have to pay probability for the first 19 rounds.

Now we develop an algorithm of the input modification. Firstly, we fix one of the iterative characteristic to δ as the previous section. Let L be the constant value:`0x08080808` and J be the set of bit positions: {2, 3, 10, 11, 18, 19, 26, 27}. Studying the proof of the above theorem tells us the condition for register values at each time to result in the required difference after 19 rounds.

Table 5. The condition for register values at each time to result in the required difference after 19 rounds

$$
\begin{array}{ll}
(F_0 \oplus G_0)^{(j)} = L^{(j)} & (j \in J) \\
(E_0 \oplus F_0)^{(j)} = L^{(j)} & \\
(E_t \oplus E_{t+1})^{(j)} = L^{(j)} & (j \in J, t = 1, 2, \ldots, 17)
\end{array}
$$

Taking this condition into account, we develop the following algorithm shown in Table 6 where for a bit string V, its value at the bit position j is denoted by $V^{(j)}$.

Table 6. The input modification

Step1: Choose randomly an initial resister values: $A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0$
Step2: Choose randomly a message block of 16 words: $W_0, W_1, \ldots, W_{15}$
Step3: Replace 8 bits of $G_0^{(j)}$ and $E_0^{(j)}$ by 8 bits of $(F_0 \oplus L)^{(j)} (j \in J)$
Step4: For $t = 0$ to 15 do:
  Compute the value: $\alpha = E_{t+1} \oplus E_t \oplus L$
  Replace 8 bits of $W_t^{(j)}$ by 8bits of $\alpha^{(j)} (j \in J)$
  Apply the t-th round function with the resulting W_t
Step5: Copy the value from W_0 to the variable: W_0^{old}
Step6: Compute the value:
  $\beta = D_{16} \oplus H_{16} \oplus \Sigma_1(E_{16}) \oplus CH(E_{16}, F_{16}, G_{16}) \oplus K_{16} \oplus L \oplus E_{16}$
Step7: Compute $W_{16} = \sigma_1(W_{14}) \oplus W_9 \oplus \sigma_0(W_1) \oplus W_0$.
Step8: Replace 8 bits of $W_{16}^{(j)}$ by 8 bits of $\beta^{(j)} (j \in J)$
Step9: Replace $W_0^{(j)}$ by the value: $(W_{16} \oplus \sigma_1(W_{14}) \oplus W_9 \oplus \sigma_0(W_1))^{(j)} (j \in J)$
Step10: Replace $H_0^{(j)}$ by the value: $(H_0 \oplus W_0 \oplus W_0^{old})^{(j)} (j \in J)$

The algorithm involves a modification of 152 input bits(=19× 8 bits), that is, $E_0^{(j)}, G_0^{(j)}, H_0^{(j)}, W_0^{(j)}, W_1^{(j)}, \ldots, W_{15}^{(j)} (j \in J)$. All the modified inputs with the difference δ results in δ again after 19 rounds, which was experimentally confirmed with 2^{20} randomly chosen inputs. We use 120 input bits out of the remaining 616 bits to add 15 rounds. This leads to an attack finding a pseudo-collision with a complexity 2^{120} for the 34-round SHA-2-XOR.

4.8 An Example of a 23-Round Pseudo-collision for SHA-2-XOR

In the Table 7 Here we list an example of a pseudo-collision producing input to SHA-2-XOR with reduced rounds. Our approach found a 23-round pseudo-collision for SHA-2-XOR with a complexity 2^{32}.

Table 7. A Message and Register values producing a 23-round pseudo-collision for SHA-2-XOR

Message words $W_0, W_1, ..., W_{15}$:
0xe97ae8e7 0x695655dd 0x57e9383b 0x8c916172
0x68e61dd1 0x2bc71033 0x081dae0f 0x5546e057
0xfd1450ef 0xcb398b6a 0xa16bf40c 0xfc7bb645
0x14b17c9c 0x1b2a8265 0xa17f20c4 0xe8f96137
Register values $(A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0)$:
0x4939a45a 0x79ec4172 0xf0ef5249 0x29b5bb6f
0xd92f76e4 0x21962dfe 0xd88e64f6 0x7b624d63

4.9 The Impact on Round-Reduced Versions of the Actual SHA-256

Since our attack on SHA-2-XOR is based on one-round iterative characteristic whose Hamming weight is relatively high, it is unlikely to obtain a high probability for the same characteristic in the case of the actual the SHA-256 hash function. Therefore it is not possible to apply our attack to the actual SHA-256 in a straightforward way.

5 Conclusions

We considered a SHA-256 variant and a SHACAL-2 variant. We presented a differential attack on these ciphers. Our result shows that SHACAL-2-XOR with up to 31 rounds has a weakness of randomness and that SHA-2-XOR with up to 34 rounds has a weakness of pseudo-collision resistance. We also presented an attack on SHACAL-2-XOR with up to 32 rounds by using the 31-round distinguisher.

Acknowledgements

The authors would like to thank Bart Preneel for his suggestions towards this analysis. We also would like to thank Joseph Lano and Souradyuti Paul for helpful comments and useful discussions. We are also grateful to the anonymous referees for their valuable remarks.

References

1. E. Biham, "New Results on SHA-0 and SHA-1," Invited talk presented at SAC 2004.
2. E. Biham, R. Chen "Near-Collision of SHA-0," in *Proceedings of CRYPT 2004*, LNCS 3152, M. Franklin, Ed., pp.290–305, 2004.

3. E. Biham, R. Chen, A. Joux, P. Carribault, C. Lemuet, and W. Jalby, "Collisions of SHA-0 and Reduced SHA-1," in *Proceedings of Eurocrypt 2005*, LNCS 3494, R. Cramer, Ed., Springer-Verlag, pp. 36–57, 2005.
4. E. Biham, A. Biryukov, A. Shamir, "Cryptanalysis of SkipJack Reduced to 31 Rounds Using Impossible Differentials," in *Proceedings of Eurocrypt'99*, LNCS 1592, pp.12–23, 1999.
5. E. Biham, A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
6. A. Biryukov, D. Wagner, "Advanced slide attacks," in *Proceedings of Eurocrypt 2000*, LNCS 1807, B. Preneel, Ed., Springer-Verlag, pp. 589–606, 2000.
7. B. D. Boer, A. Bosselaers, "Collisions for the compression function of MD5," in *Proceedings of Eurocrypt 1993*, LNCS 765, T. Helleseth, Ed., Springer-Verlag, pp. 293–304, 1993.
8. F. Chabaud and A. Joux, "Differential Collisions in SHA-0," in *Proceedings of CRYPTO'98*, LNCS 1462, H. Krawczyk, Ed., pp.56-71, Springer-Verlag, 1998.
9. I. Damgård, "A design principle for hash functions," in *Proceedings of Crypto'89*, LNCS 435, G. Brassard, Ed., Springer-Verlag, pp. 416–427, 1990.
10. H. Dobbertin, "The status of MD5 after a recent attack,", Cryptobytes, Vol. 2, No. 2, pp. 1–6, Summer 1996.
11. H. Gilbert, H. Handschuh, "Security Analysis of SHA-256 and Sisters," in *Proceedings of SAC 2003*, LNCS 3006, M. Matsui and R. Zuccherato, Eds., Springer-Verlag, pp. 175–193, 2004.
12. H. Handschuh, D. Naccache, "SHACAL," Submission to the NESSIE project, 2000. Available from `http://www.gemplus.com/smart/r_d/publications/pdf/HN00shac.pdf`.
13. P. Hawkes, M. Paddon, and G.G. Rose, "On Corrective Patterns for the SHA-2 Family," Cryptology ePrint Archive August 2004. Available from `http://eprint.iacr.org/`.
14. S. Hong, J. Kim, G. Kim, J. Sung, C. Lee, and S. Lee, "Impossible Differential Attack on 30-Round SHACAL-2," in *Proceedings of INDOCRYPT 2003*, LNCS 2904, T. Johansson and S. Maitra, Ed., Springer-Verlag, pp. 97–106, 2003.
15. J. Kim, G. Kim , S. Lee, J. Lim, and J. Song, "Related-Key Attacks on Reduced Rounds of SHACAL-2,"in *Proceedings of INDOCRYPT 2004* , LNCS 3348, A. Canteaut and K. Viswanathan Ed., Springer-Verlag, pp. 175–189 2004.
16. L. R. Knudsen and J. E. Mathiassen, "Preimage and collision attacks on MD2," in *Proceedings of FSE 2005*, LNCS 3557, H. Gilbert and H. Handschuh Ed., Springer-Verlag, pp. 255–267, 2005.
17. A. Menezes, P. van Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.
18. National Institute of Standards and Technology, FIPS-180-2: "Secure Hash Standard (SHS)," August 2002.
19. V. Rijmen, B. Preneel, "Improved characteristics for differential cryptanalysis of hash functions based on block ciphers," Fast Software Encryption, Lecture Notes in Computer Science 1008, B. Preneel, Ed., Springer-Verlag, 1995, pp. 242-248.
20. R. Rivest, "The MD5 message-digest algorithm," Request for Comments (RFC) 1321, Internet Activities Board, Internet Privacy Task Force, April 1992.
21. M. Saarinen, "Cryptanalysis of Block Ciphers Based on SHA-1 and MD5," in *Proceedings of FSE 2003*, LNCS 2887, T. Johansson, Ed., Springer-Verlag, pp. 36–44, 2003.

22. Y. Shin, J. Kim, G. Kim, S. Hong, and S. Lee, "Differential-Linear Type Attacks on Reduced Rounds of SHACAL-2," in *Proceedings of ACISP 2004*, LNCS 3108, H. Wang, J. Pieprzyk, and V. Varadharajan, Ed., Springer-Verlag, pp. 110–122, 2004.
23. X. Wang, X. Lai, D. Feng, H. Chen, and X. Yu, "Cryptanalysis of the Hash Functions MD4 and RIPEMD," in *Proceedings of Eurocrypt 2005*, LNCS 3494, R. Cramer, Ed., Springer-Verlag, pp. 1–18, 2005.
24. H. Yoshida, A. Biryukov, C. D. Cannière, J. Lano, and B. Preneel, "Non-randomness of the Full 4 and 5-pass HAVAL," in *Proceedings of SCN 2004*, LNCS 3352, C. Blundo and S. Climato, Ed., Springer-Verlag, pp. 324–336, 2005.