Non-Linear Reduced Round Attacks Against SHA-2 Hash family

Somitra Kumar Sanadhya* and Palash Sarkar

Applied Statistics Unit,
Indian Statistical Institute,
203, B.T. Road, Kolkata,
India 700108.
somitra_r@isical.ac.in, palash@isical.ac.in

Abstract. Most of the attacks against (reduced) SHA-2 family in literature have used local collisions which are valid for linearized version of SHA-2 hash functions. Recently, at FSE '08, an attack against reduced round SHA-256 was presented by Nikolić and Biryukov which used a local collision which is valid for the actual SHA-256 function. It is a 9-step local collision which starts by introducing a modular difference of 1 in the two messages. It succeeds with probability roughly 1/3. We build on the work of Nikolić and Biryukov and provide a generalized nonlinear local collision which accepts an arbitrary initial message difference. This local collision succeeds with probability 1. Using this local collision we present attacks against 18-step SHA-256 and 18-step SHA-512 with arbitrary initial difference. Both of these attacks succeed with probability 1. We then present special cases of our local collision and show two different differential paths for attacking 20-step SHA-256 and 20-step SHA-512. One of these paths is the same as presented by Nikolić and Biryukov while the other one is a new differential path. Messages following both these differential paths can be found with probability 1. This improves on the previous result where the success probability of 20-step attack was 1/3. Finally, we present two differential paths for 21-step collisions for SHA-256 and SHA-512, one of which is a new path. The success probability of these paths for SHA-256 is roughly 2^{-15} and 2^{-17} which improves on the 21-step attack having probability 2^{-19} reported earlier. We show examples of message pairs following all the presented differential paths for up to 21-step collisions in SHA-256. We also show first real examples of colliding message pairs for up to 20-step reduced SHA-512.

1 Introduction

Cryptanalysis of hash functions has been an area of intense interest to the research community since past decade and a half. Many hash functions were broken in this time, most notable among them are MD5 [12], SHA-0 [13] and theoretical break of SHA-1 [11]. This has directed the attention of the cryptology community to the SHA-2 family of hash functions.

Known Results for the SHA-2 Family: Gilbert and Handschuh (GH) [2] were the first to study local collisions in the SHA-2 family. They reported a 9-step local collision for linearized version of SHA-256 and estimated the probability of the differential path to be 2^{-66}. This probability estimate was later improved by Hawkes et al. [3]. Sanadhya and Sarkar [7] presented 16 new 9-step local collisions for SHA-2 family of hash functions. All these local collisions are also for the linearized version of SHA-256. In [8], an algorithm for generating 18-step SHA-256 collisions was developed using one of these local collisions and many colliding message pairs for 18-step SHA-256 were obtained. The message expansion of SHA-256 was studied by Mendel et al. [4], who reported a colliding message pair for 18-step SHA-256 which was recently corrected in [5]. They used the linearized local collision from [2] in their work. Mendel et al. [4] also improved the probability estimate of the Gilbert-Handschuh local collision to values similar to those obtained in [3].

Recently, Nikolić and Biryukov [6] presented a new local collision which uses modular differences instead of the XOR differences. Since this local collision is for the actual SHA-256 (and not its linearized version), its probability is much higher than the linearized local collisions presented earlier. For the first time in the literature, the authors in [6] worked directly with modular differences for SHA-256. Using this local collision they obtained 20-step and 21-step collisions for SHA-256 with probabilities 1/3 and $1/2^{19}$ respectively.

* This author is supported by the Ministry of Information Technology, Govt. of India.

Our Contributions: We build on the work of Nikolić and Biryukov [6] and present a generalized non-linear local collision which accepts an arbitrary initial message difference. In [6], sufficient conditions for the differential path are determined and a particular local collision is obtained. We work with exact solutions of conditions imposed by the differential path and obtain general solutions of these conditions. Since we work with exact solutions of the conditions, our local collision is deterministic i.e. it holds with probability 1. Using this local collision, we obtain collisions for 18-step SHA-256 and 18-step SHA-512 with an arbitrary initial message difference. These attacks succeed with probability 1.

Then we show special instances of our generalized local collision which are suitable for finding collisions for 20-step SHA-256 and 20-step SHA-512. We present two such instances. One of these instances is a new local collision which can be realized in two different ways. The other one is the same as that presented by Nikolić and Biryukov for obtaining 20-step collision in [6]. However, unlike in [6], our 20-step attacks succeed with probability 1.

Finally, we use 20-step collisions to obtain 21-step collisions for SHA-256 as in [6]. There the probability for 21-step SHA-256 collisions is experimentally estimated to be about 2^{-19}. We improve the efficiency of the probabilistic search used in this case and obtain 21-step collisions for SHA-256 with estimated experimental probability of 2^{-15}. This is also the first time that actual collisions for SHA-512 reduced up to 20 steps are presented.

2 Notation

In this paper we use the following notation:

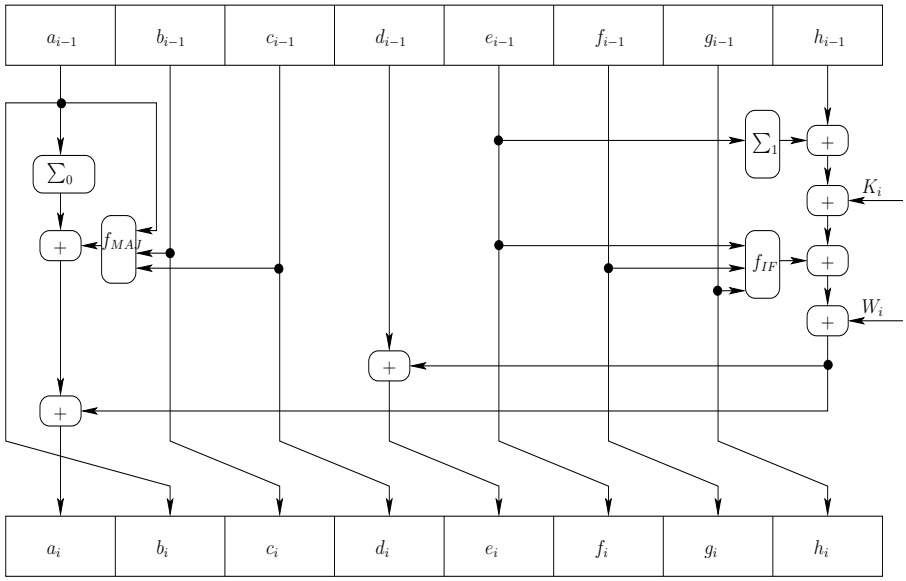
- $m_i \in \{0,1\}^n$, $W_i \in \{0,1\}^n$, $W'_i \in \{0,1\}^n$ for any i. The word size n is 32 for SHA-256 and 64 for SHA-512.
- The colliding message pair: $\{m_0, m_1, m_2, \ldots m_{15}\}$ and $\{m'_0, m'_1, m'_2, \ldots m'_{15}\}$.
- The expanded message pair: $\{W_0, W_1, W_2, \ldots W_{r-1}\}$ and $\{W'_0, W'_1, W'_2, \ldots W'_{r-1}\}$. The number of steps r is 64 for SHA-256 and 80 for SHA-512.
- The internal registers for the two message pairs in step i: $\{a_i, \ldots, h_i\}$ and $\{a'_i, \ldots, h'_i\}$.
- $\mathrm{ROTR}^k(x)$: Right rotation of an n-bit quantity x by k bits.
- $\mathrm{SHR}^k(x)$: Right shift of an n-bit quantity x by k bits.
- $\oplus$: bitwise XOR.
- $+$: addition modulo 2^n.
- $-$: subtraction modulo 2^n.
- $\delta X = X' - X$ where X is an n-bit quantity.
- $\delta\Sigma_1(e_i) = \Sigma_1(e'_i) - \Sigma_1(e_i)$.
- $\delta\Sigma_0(a_i) = \Sigma_0(a'_i) - \Sigma_0(a_i)$.
- $\delta f^i_{MAJ}(x,y,z)$: Output difference of the f_{MAJ} function in step i when its inputs differ by x, y and z. That is, $\delta f^i_{MAJ}(x,y,z) = f_{MAJ}(a_i + x, b_i + y, c_i + z) - f_{MAJ}(a_i, b_i, c_i)$.
- $\delta f^i_{IF}(x,y,z)$: Output difference of the f_{IF} function in step i when its inputs differ by x, y and z. That is, $\delta f^i_{IF}(x,y,z) = f_{IF}(e_i + x, f_i + y, g_i + z) - f_{IF}(e_i, f_i, g_i)$.

3 The SHA-2 Hash Family

The newest members of SHA family of hash functions were standardized by US NIST in 2002 [10]. There are 2 differently designed functions in this standard: the SHA-256 and SHA-512. In addition, the standard also specifies their truncated version: SHA-224 and SHA-384. The number in the name of the hash function refers to the length of message digest produced by that function. Next we describe SHA-256 and SHA-512 in detail.

The round function of SHA-2 hash family is shown in Figure 1. Eight registers are used in the evaluation of SHA-2. The initial value in the registers is specified by an $8 \times n$ bit IV, n=32 for SHA-256

Fig. 1. Round function of SHA-2 hash family



and 64 for SHA-512. In Step i, the 8 registers are updated from (a_{i-1}, b_{i-1}, c_{i-1}, d_{i-1}, e_{i-1}, f_{i-1}, g_{i-1}, h_{i-1}) to (a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i) according to the following equations:

$$\left.\begin{aligned} a_i &= \Sigma_0(a_{i-1}) + f_{MAJ}(a_{i-1}, b_{i-1}, c_{i-1}) + \Sigma_1(e_{i-1}) \\ &\quad + f_{IF}(e_{i-1}, f_{i-1}, g_{i-1}) + h_{i-1} + K_i + W_i \\ b_i &= a_{i-1} \\ c_i &= b_{i-1} \\ d_i &= c_{i-1} \\ e_i &= d_{i-1} + \Sigma_1(e_{i-1}) + f_{IF}(e_{i-1}, f_{i-1}, g_{i-1}) \\ &\quad + h_{i-1} + K_i + W_i \\ f_i &= e_{i-1} \\ g_i &= f_{i-1} \\ h_i &= g_{i-1} \end{aligned}\right\}$$

The f_{IF} and the f_{MAJ} are three variable boolean functions defined as:

$$\begin{aligned} f_{IF}(x, y, z) &= (x \wedge y) \oplus (\neg x \wedge z), \\ f_{MAJ}(x, y, z) &= (x \wedge y) \oplus (y \wedge z) \oplus (z \wedge x). \end{aligned}$$

For SHA-256, the functions Σ_0 and Σ_1 are defined as:

$$\begin{aligned} \Sigma_0(x) &= ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x), \\ \Sigma_1(x) &= ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x). \end{aligned}$$

For SHA-512, the corresponding functions are:

$$\begin{aligned} \Sigma_0(x) &= ROTR^{28}(x) \oplus ROTR^{34}(x) \oplus ROTR^{39}(x), \\ \Sigma_1(x) &= ROTR^{14}(x) \oplus ROTR^{18}(x) \oplus ROTR^{41}(x). \end{aligned}$$

Round i uses a n-bit word W_i which is derived from the message and a constant word K_i. There are 64 steps in SHA-256 and 80 in SHA-512. The hash function operates on a 512-bit (resp. 1024-bit)

message specified as 16 words of 32 (resp. 64) bits for SHA-256 (resp. SHA-512). Given the message words m_0, m_1, $\ldots m_{15}$, the W_i 's are computed using the equation:

$$W_i = \begin{cases} m_i & \text{for } 0 \leq i \leq 15 \\ \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16} & \text{for } 16 \leq i \leq 63 \text{ (or 80)} \end{cases} \quad (1)$$

For SHA-256, the functions σ_0 and σ_1 are defined as:

$$\sigma_0(x) = ROTR^7(x) \ \oplus ROTR^{18}(x) \oplus SHR^3(x),$$
$$\sigma_1(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x).$$

And for SHA-512, they are defined as:

$$\sigma_0(x) = ROTR^1(x) \ \oplus ROTR^8(x) \ \oplus SHR^7(x),$$
$$\sigma_1(x) = ROTR^{19}(x) \oplus ROTR^{61}(x) \oplus SHR^6(x).$$

The IV $= (a_{-1}, b_{-1}, c_{-1}, d_{-1}, e_{-1}, f_{-1}, g_{-1}, h_{-1})$ is defined as (`0x6a09e667`, `0xbb67ae85`, `0x3c6ef372`, `0xa54ff53a`, `0x510e527f`, `0x9b05688c`, `0x1f83d9ab`, `0x5be0cd19`) for SHA-256. Different IV values are defined for SHA-224, SHA-384 and SHA-512. For details, see [10].

The output hash value of a one block (512-bit for SHA-256 and 1024-bit for SHA-512) message is obtained by chaining the IV with the register values at the end of the final round as per the Merkle-Damgård construction. A similar strategy is used for multi-block messages, where the IV for next block is taken as the hash output of the previous block.

4 Collision Attacks Against the SHA-2 Hash Family

The aim of collision attacks against hash functions is to obtain two different messages which produce the same digest under that hash function. The hash functions use one word of the message in each step and process the message for multiple steps. Typically, an attacker introduces a small difference in one word of the message. Using the terminology from [1], this initial difference is called the "perturbation message difference". Next few message words are chosen to differ in such a manner that all the introduced differences cancel themselves with high probability. These later message word differences are called "correction differences".

Not all the message words used in different steps of the hash function are freely available to the attacker. Most of the hash designs have 16 words of freedom which is available in the first 16 steps of hash evaluation. Rest of the message words are computed on the basis of the first 16 words using "message expansion".

A "local collision" is a collision producing differential path (and the message differences required for this path) spanning a small number of steps in which the message expansion in not considered. Gilbert and Handschuh reported the first local collision for SHA-256 [2] using XOR differences in the message words. Sanadhya and Sarkar [7] reported 16 other local collisions, all of which used XOR differences in the message words. Very recently, Nikolić and Biryukov [6] showed a different type of local collision, which used modular differences in the message words. The XOR difference based local collisions have linear property, i.e. any number of them can be freely superimposed, whereas the modular difference based local collision can not be superimposed freely. For this reason, we refer to them as "linear local collisions" and "nonlinear local collisions" respectively.

We present our new nonlinear local collision next.

5 A General Class Of Nonlinear Local Collisions

Table 1 shows the general structure of a 9-step local collision for SHA-2 family. The perturbation message difference is taken to be x and other message differences are later computed. In Table 1, the registers $(a_{i-1}, \ldots, h_{i-1})$ and W_i are inputs to Step i of the hash evaluation and this step outputs the registers $(a_i, \ldots, h_i)$.

Table 1. A 9-step nonlinear local collision for SHA-256.

Step i	δW_i	δa_i	δb_i	δc_i	δd_i	δe_i	δf_i	δg_i	δh_i
$i-1$	0	0	0	0	0	0	0	0	0
i	x	x	0	0	0	x	0	0	0
$i+1$	δW_{i+1}	0	x	0	0	y	x	0	0
$i+2$	δW_{i+2}	0	0	x	0	z	y	x	0
$i+3$	δW_{i+3}	0	0	0	x	0	z	y	x
$i+4$	δW_{i+4}	0	0	0	0	x	0	z	y
$i+5$	δW_{i+5}	0	0	0	0	0	x	0	z
$i+6$	δW_{i+6}	0	0	0	0	0	0	x	0
$i+7$	δW_{i+7}	0	0	0	0	0	0	0	x
$i+8$	δW_{i+8}	0	0	0	0	0	0	0	0

5.1 Message Word Differences for Table 1

In Step i of SHA-2, only the registers a_i and e_i are computed. Rest of the registers are copies of the old ones. Therefore we focus on these two register evaluations only. From (1), we get:

$$\delta e_i = \delta\Sigma_1(e_{i-1}) + \delta f_{IF}(\delta e_{i-1}, \delta f_{i-1}, \delta g_{i-1}) + \delta d_{i-1} + \delta h_{i-1} + \delta W_i, \tag{2}$$

$$\begin{aligned}\delta a_i &= \delta\Sigma_0(a_{i-1}) + \delta f_{MAJ}(\delta a_{i-1}, \delta b_{i-1}, \delta c_{i-1}) + \delta\Sigma_1(e_{i-1}) + \\ &\quad \delta f_{IF}(\delta e_{i-1}, \delta f_{i-1}, \delta g_{i-1}) + \delta h_{i-1} + \delta W_i, \\ &= \delta\Sigma_0(a_{i-1}) + \delta f_{MAJ}(\delta a_{i-1}, \delta b_{i-1}, \delta c_{i-1}) + \delta e_i - \delta d_{i-1}.\end{aligned} \tag{3}$$

We now try to satisfy the restriction imposed by the differential path of Table 1 by defining suitable difference of the message words in various steps.

Step i : If $\delta W_i = x$, then this difference will propagate to both the registers a_i and e_i.

Step (i+1) : At this step $a'_i - a_i = e'_i - e_i = x$. We want $\delta a_{i+1} = 0$ and $\delta e_{i+1} = y$. From (3) and (2), we get:

$$\begin{aligned}\delta a_{i+1} = 0 &= \delta\Sigma_0(a_i) + \delta f^i_{MAJ}(x, 0, 0) + \delta\Sigma_1(e_i) + \delta f^i_{IF}(x, 0, 0) + \delta W_{i+1}, \\ \delta e_{i+1} = y &= \delta\Sigma_1(e_i) + \delta f^i_{IF}(x, 0, 0) + \delta W_{i+1}.\end{aligned}$$

The exact solution of the equations above is:

$$y = -\delta\Sigma_0(a_i) - \delta f^i_{MAJ}(x, 0, 0), \tag{4}$$

$$\delta W_{i+1} = y - \delta f^i_{IF}(x, 0, 0) - \delta\Sigma_1(e_i). \tag{5}$$

Step (i+2) : At this step $b'_{i+1} - b_{i+1} = f'_{i+1} - f_{i+1} = x$ and $e'_{i+1} - e_{i+1} = y$. We want $\delta a_{i+2} = 0$ and $\delta e_{i+2} = z$. From (3) and (2), we get:

$$\begin{aligned}\delta a_{i+2} = 0 &= \delta f^{i+1}_{MAJ}(0, x, 0) + \delta\Sigma_1(e_{i+1}) + \delta f^{i+1}_{IF}(y, x, 0) + \delta W_{i+2}, \\ \delta e_{i+2} = z &= \delta\Sigma_1(e_{i+1}) + \delta f^{i+1}_{IF}(y, x, 0) + \delta W_{i+2}.\end{aligned}$$

The conditions above translate to:

$$z = -\delta f^{i+1}_{MAJ}(0, x, 0), \tag{6}$$

$$\delta W_{i+2} = z - \delta f^{i+1}_{IF}(y, x, 0) - \delta\Sigma_1(e_{i+1}). \tag{7}$$

Step (i+3) : At this step $c'_{i+2} - c_{i+2} = g'_{i+2} - g_{i+2} = x$, $e'_{i+2} - e_{i+2} = z$ and $f'_{i+2} - f_{i+2} = y$. We want $\delta a_{i+3} = 0$ and $\delta e_{i+3} = 0$. From (3) and (2), we get:

$$\delta a_{i+3} = 0 = \delta f_{MAJ}^{i+2}(0,0,x) + \delta\Sigma_1(e_{i+2}) + \delta f_{IF}^{i+2}(z,y,x) + \delta W_{i+3},$$
$$\delta e_{i+3} = 0 = \delta\Sigma_1(e_{i+2}) + \delta f_{IF}^{i+2}(z,y,x) + \delta W_{i+3}.$$

The conditions above translate to:

$$\delta f_{MAJ}^{i+2}(0,0,x) = 0, \tag{8}$$

$$\delta W_{i+3} = -\delta f_{IF}^{i+2}(z,y,x) - \delta\Sigma_1(e_{i+2}). \tag{9}$$

Step (i+4) : At this step $d'_{i+3} - d_{i+3} = h'_{i+3} - h_{i+3} = x$, $f'_{i+3} - f_{i+3} = z$ and $g'_{i+3} - g_{i+3} = y$. We want $\delta a_{i+4} = 0$ and $\delta e_{i+4} = x$. From (3) and (2), we get:

$$\delta a_{i+4} = 0 = \delta f_{IF}^{i+3}(0,z,y) + x + \delta W_{i+4},$$
$$\delta e_{i+4} = x = \delta f_{IF}^{i+3}(0,z,y) + x + x + \delta W_{i+4}.$$

The conditions above translate to:

$$\delta W_{i+4} = -x - \delta f_{IF}^{i+3}(0,z,y). \tag{10}$$

Step (i+5) : At this step $e'_{i+4} - e_{i+4} = x$, $g'_{i+4} - g_{i+4} = z$ and $h'_{i+4} - h_{i+4} = y$. We want $\delta a_{i+5} = \delta e_{i+5} = 0$. From (3) and (2), we get:

$$\delta a_{i+5} = 0 = \delta\Sigma_1(e_{i+4}) + \delta f_{IF}^{i+4}(x,0,z) + y + \delta W_{i+5},$$
$$\delta e_{i+5} = 0 = \delta\Sigma_1(e_{i+4}) + \delta f_{IF}^{i+4}(x,0,z) + y + \delta W_{i+5}.$$

The conditions above translate to:

$$\delta W_{i+5} = -y - \delta f_{IF}^{i+4}(x,0,z) - \delta\Sigma_1(e_{i+4}). \tag{11}$$

Step (i+6) : At this step $f'_{i+5} - f_{i+5} = x$ and $h'_{i+5} - h_{i+5} = z$. We want $\delta a_{i+6} = \delta e_{i+6} = 0$. From (3) and (2), we get:

$$\delta a_{i+6} = 0 = \delta f_{IF}^{i+5}(0,x,0) + z + \delta W_{i+6},$$
$$\delta e_{i+6} = 0 = \delta f_{IF}^{i+5}(0,x,0) + z + \delta W_{i+6}.$$

The conditions above translate to:

$$\delta W_{i+6} = -z - \delta f_{IF}^{i+5}(0,x,0). \tag{12}$$

Step (i+7) : At this step $g'_{i+6} - g_{i+6} = x$. We want $\delta a_{i+7} = \delta e_{i+7} = 0$. From (3) and (2), we get:

$$\delta a_{i+7} = 0 = \delta f_{IF}^{i+6}(0,0,x) + \delta W_{i+7},$$
$$\delta e_{i+7} = 0 = \delta f_{IF}^{i+6}(0,0,x) + \delta W_{i+7}.$$

The conditions above translate to:

$$\delta W_{i+7} = -\delta f_{IF}^{i+6}(0,0,x). \tag{13}$$

Step (i+8) : At this step $h'_{i+7} - h_{i+7} = x$. We want $\delta a_{i+8} = \delta e_{i+8} = 0$. This will happen as desired if we have:

$$\delta W_{i+8} = -x. \tag{14}$$

5.2 Solution of Equations

To find a local collision, we need message pairs which will satisfy (4) to (14). Out of these, only (8) puts restrictions on the message pair. Rest of the equations merely define the correction message differences. For clarity, we reproduce the condition here.

$$\delta f_{MAJ}^{i+2}(0, 0, x) = 0.$$

Next we explain how to satisfy this condition easily. This is based on the technique in [6]. The f_{MAJ} function has registers (a, b, c) as inputs. The necessary condition for the two different inputs to the f_{MAJ} to not propagate the difference in the output is that :

– Registers a_{i+2} and b_{i+2} must have same value at those bit positions where registers c'_{i+2} and c_{i+2} differ.

Note that $b_{i+2} = a_{i+1}$. Although the condition above requires us to ensure equality of bit patterns in the two registers only at some places, we can strengthen this condition a little and try to make these register values exactly equal. Thus, we need to satisfy $a_{i+2} = a_{i+1}$.

Note that we have put no restriction on the message words themselves in solving earlier equations. The only restrictions are on the "difference" of messages. To ensure the equality of the registers as desired, we can now put some conditions on the actual message word W_{i+2}. When the $(i + 2)^{th}$ step of the hash evaluation is executed, the registers $\{a_{i+1}, \ldots, h_{i+1}\}$ will already be available. So we can choose W_{i+2} such that it produces a value in register a_{i+2} which is equal to the already known value a_{i+1}. This requires solving the following equations simultaneously:

$$a_{i+2} = \Sigma_0(a_{i+1}) + f_{MAJ}(a_{i+1}, b_{i+1}, c_{i+1}) + \Sigma_1(e_{i+1}) + f_{IF}(e_{i+1}, f_{i+1}, g_{i+1}) + h_{i+1} + K_{i+2} + W_{i+2},$$
$$a_{i+2} = a_{i+1}.$$

Hence, we choose W_{i+2} such that:

$$W_{i+2} = a_{i+1} - \Sigma_0(a_{i+1}) - f_{MAJ}(a_{i+1}, b_{i+1}, c_{i+1}) - \Sigma_1(e_{i+1}) - f_{IF}(e_{i+1}, f_{i+1}, g_{i+1}) - h_{i+1} - K_{i+2}. \quad (15)$$

5.3 Obtaining a Local Collision

To obtain the 9-step local collision as in Table 1, we first select the perturbation message difference δW_i as a randomly generated 32-bit (or 64-bit) quantity x. The differences δW_j for $j \in \{(i + 1), \ldots, (i + 8)\}$ are defined by (5), (7), (9), (10), (11), (12), (13) and (14). In addition, as discussed in the last section, we choose W_{i+2} such that (15) is satisfied. Rest of the message words could be any randomly chosen 32-bit (or 64-bit) words. This local collision holds with probability 1, since all the steps are deterministic and feasible.

6 Extending a Single Local Collision to Obtain 18-Step Collisions

In this section we explain how to obtain 18-step collisions using the local collision shown in this paper. We discuss three different types of differential paths depending on the value of the differential z used in δe_{i+2} to δh_{i+5} in Table 1.

For all the different cases that we describe next, we choose to span the 9-step local collision from Step 3 to Step 11. The message differentials δW_i for $i \in \{3, 4, \ldots, 11\}$ are defined by the local collision. We use a single local collision, which implies that all the other free message words are equal. That is, $\delta W_i = 0$ for $i \in \{0, 1, 2, 12, 13, 14, 15\}$.

First two steps of message expansion of SHA-2 define the message words W_{16} and W_{17} as follows:

$$W_{16} = \sigma_1(W_{14}) + W_9 + \sigma_0(W_1) + W_0$$
$$W_{17} = \sigma_1(W_{15}) + W_{10} + \sigma_0(W_2) + W_1$$

From these two equations, it is clear that if $\delta W_9 = \delta W_{10} = 0$ then the two expanded message words will be equal for Steps 17 and 18. This will result in an 18-step collision for SHA-2. Note that δW_9 and δW_{10} correspond to Steps 7 and 8 of the local collision used. Hence our target is to get differentials of the message in these two steps to vanish.

6.1 When $z = 0$ in the Local Collision

In this case the local collision looks similar to the one given in [6]. But note that that our local collision accepts any random message difference x, whereas in [6] the specific value $x = 1$ is used.

As explained above, we need to ensure that (12) and (13) give zero differences. In addition we also need $z = 0$ from (6).

To get $z = 0$, we need to have $\delta f^4_{MAJ}(0, x, 0) = 0$. Similar to the methods used in Section 5.2, we can ensure this by the sufficient condition $a_4 = c_4$ (i.e. $a_4 = a_2$) which can be deterministically satisfied by suitable choice of W_4.

Next we need two consecutive message differences zero at Steps 10 and 11 of the differential path. Equation 12 corresponding to Step 10 gives zero difference if $\delta f^8_{IF}(0, x, 0) = 0$. This can be deterministically satisfied by choosing W_8 such that $e_8 = 0$. In this case, f_{IF} selects its third argument which does not have any difference. Similarly (13) can be satisfied by choosing W_9 such that $\delta f^9_{IF}(0, 0, x) = 0$ i.e. this time we need $e_9 = -1$. Thus we can deterministically obtain 18-step collisions for SHA-2 for any random initial perturbation x. Message pairs colliding for 18-step SHA-256 and for 18-step SHA-512 with initial perturbation selected randomly are given in Section A.

6.2 When $z \neq 0$ in the Local Collision

As before, we need that (12) and (13) give zero difference in message words. Stating these equations explicitly, we require that:

$$-z = f_{IF}(e_8, f_8 + x, g_8) - f_{IF}(e_8, f_8, g_8), \tag{16}$$

$$0 = f_{IF}(e_9, f_9, g_9 + x) - f_{IF}(e_9, f_9, g_9). \tag{17}$$

Equation 17 is easy to satisfy by selecting W_9 such that $e_9 = -1$. Then f_{IF} selects its second argument which does not have any difference. However, (16) is not easily satisfied this time. This equation is easy to solve only for special z values of the type $z = 0$ or $z = \pm x$. To have an 18-step collision, we need z to take these special values. We discuss the two non-zero cases for z separately.

When $z = -x$: The value z gets defined by (6). So we need to handle this equation, which states that:

$$z = -\delta f^4_{MAJ}(0, x, 0).$$

This puts restrictions on the values of the registers $\{a_4, b_4, c_4\}$ and perturbation difference x such that the following condition holds:

$$f_{MAJ}(a_4, b_4 + x, c_4) - f_{MAJ}(a_4, b_4, c_4) = x. \tag{18}$$

Left hand side of (18) can be thought of as a function which accepts 4 words of input and returns 1 word of output. Clearly, there are many solutions to this equation. One solution to this equation is $a_4 = -1$, $c_4 = 0$ with b_4 being any arbitrary value. For any x, this will be a solution to (18). [This solution was suggested by an anonymous reviewer of ACISP 2008.] Alternately, random search of the 4 word space can be made which also quickly gives solutions for (18). The cost of finding random solutions to this equation is also negligible.

Once some values for $(a_4, b_4, c_4) = (a_4, a_3, a_2)$ and x which satisfy (18) have been selected, we need to have these register values in the differential path at the appropriate step. This can be done by choosing

W_4, W_3 and W_2 appropriately. Let the selected values of (a_4, a_3, a_2) be (α, β, γ). Then the message words should be chosen to satisfy the following equations:

$$W_2 = \gamma - \Sigma_0(a_1) - f_{MAJ}(a_1, b_1, c_1) - \Sigma_1(e_1) - f_{IF}(e_1, f_1, g_1) - h_1 - K_2, \tag{19}$$

$$W_3 = \beta - \Sigma_0(\gamma) - f_{MAJ}(\gamma, b_2, c_2) - \Sigma_1(e_2) - f_{IF}(e_2, f_2, g_2) - h_2 - K_3, \tag{20}$$

$$W_4 = \alpha - \Sigma_0(\beta) - f_{MAJ}(\beta, \gamma, c_3) - \Sigma_1(e_3) - f_{IF}(e_3, f_3, g_3) - h_3 - K_4. \tag{21}$$

We also need to satisfy (16). This is easily handled by having $e_8 = -1$ so that f_{IF} selects its middle argument and propagates the difference x. This can be done by choosing W_8 appropriately.

To summarize, we start a local collision spanning Steps 3 to 11 and choose some values of (a_4, a_3, a_2) and x such that (18) is satisfied. The differences in messages words δW_i for $i \in \{3, 4, \ldots 11\}$ are defined by the local collision. In addition, we select message words W_2, W_3 and W_4 by solving (19), (20) and (21). The local collision also requires us to choose W_5 in a particular manner (as explained in Section 5.2). Finally, we need to choose W_8 and W_9 so as to ensure that $e_8 = e_9 = -1$. Rest of the message words can be selected randomly. Note that we must first select W_0 and W_1 and then only can we solve for W_2, W_3 and W_4 etc. Further, the only cost involved in obtaining such 18-step collisions is in selecting suitable values of (a_4, b_4, c_4) and x. The 18-step collision, which is obtained after any solution of (18) is chosen, holds with probability 1.

When $z = x$: This time the majority condition takes the form:

$$f_{MAJ}(a_4, b_4 + x, c_4) - f_{MAJ}(a_4, b_4, c_4) = -x. \tag{22}$$

There are many solutions to this equation as well. In particular, one subset of solutions is given by the following choice of the variables: a_4=b_4=p and c_4=$p+x$ where p is any arbitrary 32-bit quantity and $x = 2^{31}$. This solution works because $2^{31} = -2^{31}$ in modulo 2^{32} arithmetic. The SHA-512 case is similar where we can use 2^{63} in place of 2^{31}. The cost for finding random 32-bit solutions for the above equation is experimentally found to be about 2^{24}. This means that finding a random solution for SHA-256 takes a few seconds on an ordinary PC. Few such solutions for 32-bit words are listed in Table 2.

Table 2. Example values of register (a, b, c) and x such that $f_{MAJ}(a, b+x, c) - f_{MAJ}(a, b, c) = -x$. Registers a and c can also be exchanged due to the symmetry of f_{MAJ}.

No.	a	b	c	x
1	0	0	80000000	80000000
2	44070d26	9f85286b	823480b1	7ffdfffc
3	1b1704f1	511209a2	f504556a	00000100
4	fcbeab96	a56c2117	0f94f865	fe27f002
5	a4cffbbd	8266ace3	392a62f6	fffffffa

In this case, (12) and (13) are:

$$\begin{aligned} \delta W_9 &= -x - \delta f^8_{IF}(0, x, 0) \\ \delta W_{10} &= -\delta f^9_{IF}(0, 0, x) \end{aligned}$$

The right hand side of the second equation above can be made zero by choosing $e_9 = -1$ so that the f_{IF} function chooses its middle argument. This can be achieved by suitably choosing W_9. We use random choices of words to obtain $\delta W_9 = 0$. The complexity of this step is directly related to the hamming weight of x. For a 1-bit x the probability of satisfying this step is about $1/2$ to $1/2^3$. For

20-bit x this equation gets satisfied with probability about $1/2^8$ to $1/2^{20}$. This cost is equivalent to a fraction of a second on an ordinary PC.

To summarize, we start a local collision spanning steps 3 to 11 and choose some values of (a_4, a_3, a_2) and x such that (22) is satisfied. The differences in messages words δW_i for $i \in \{3, 4, \ldots 11\}$ are defined by the local collision. In addition, we select message words W_2, W_3 and W_4 by solving (19), (20) and (21). The word W_8 is selected as explained above. Besides, W_5 is selected in the same way as in Section 5.2. Rest of the message words can be selected randomly.

There are two costs involved in obtaining such 18-step collisions: (1) Selecting suitable values of (a_4, b_4, c_4) and x satisfying (22), and (2) Satisfying $\delta W_9 = 0$. The first condition can be always satisfied by choosing suitable pre-computed values. The only probability for such 18-step collisions comes from the satisfaction of the second condition.

Message pairs colliding for 18-step SHA-256 and 18-step SHA-512 following this differential path (for both the cases $z = x$ and $z = -x$) are shown in Section A.

7 Extending a Single Local Collision to Obtain 20-Step Collisions

We follow the technique used in [6] to obtain 20-step collisions for SHA-256. This time we need to handle first 4 steps of message expansion. These steps are:

$$
\begin{aligned}
W_{16} &= \sigma_1(W_{14}) + \ W_9 + \sigma_0(W_1) + W_0 \\
W_{17} &= \sigma_1(W_{15}) + W_{10} + \sigma_0(W_2) + W_1 \\
W_{18} &= \sigma_1(W_{16}) + W_{11} + \sigma_0(W_3) + W_2 \\
W_{19} &= \sigma_1(W_{17}) + W_{12} + \sigma_0(W_4) + W_3
\end{aligned}
$$

If a single local collision spanning from Step 5 to Step 13 is used and all other messages outside the scope of this local collision are taken to have zero differentials, then $\delta W_i = 0$ for $i \in \{0, 1, 2, 3, 4, 14, 15\}$. This implies that if we can have $\delta W_9 = \delta W_{10} = \delta W_{11} = \delta W_{12} = 0$, then the differentials of the first 4 expanded message words will be zero. In this case the message expansion will not play a role and we will be able to extend a single local collision to 20 steps.

The local collision presented in [6] is such that the message differentials at steps $i+4$ to $i+7$ are zero for it ($i = 5$ is the starting step of the local collision). Hence it can be used to obtain 20-step collisions directly. The local collision we presented is more general but does not necessarily have 4 consecutive message differentials equal to zero. Now we find particular instances of our local collision such that we have zero differentials as desired. This time we work with sufficient conditions as in [6].

To obtain the 4 consecutive zero differentials in the local collision, we need to have differentials generated by (10), (11), (12) and (13) (corresponding to Steps 9, 10, 11 and 12 of the differential path) to be equal to zero. We next discuss the conditions put by these equations. We also need to control the values of y and z by (4) and (6). As in [6], we start the local collision by choosing $x = 1$.

Equation 4: This equation contains the term $\delta\Sigma_0(a_5) = \Sigma_0(a'_5) - \Sigma_0(a_5)$. From the differential path we know that $\delta a_5 = a'_5 - a_5 = x$. Differential behavior of the non-linear function Σ_0 is difficult to analyze. To make it tractable, we choose $\delta\Sigma_0(a_5) = x = 1$. For this case, the only solutions are $a_5 = -1 =$ `0xffffffff` and $a'_5 = 0$. We also put restriction that the f_{MAJ} term doesn't propagate any difference. This condition $f^5_{MAJ}(x, 0, 0) = 0$ implies $b_5 = c_5$, i.e. $a_4 = a_3$. Conditions on a_4 and a_5 registers can be deterministically satisfied by choosing W_4 and W_5 suitably. By the choices made above, this equation gives $y = -1$.

Equation 11: This equation contains the term $\delta\Sigma_1(e_9) = \Sigma_1(e'_9) - \Sigma_1(e_9)$. From the differential path we know that $\delta e_9 = e'_9 - e_9 = x$. Differential behaviour of the non-linear function Σ_1 is difficult to analyze. Similar to the previous equation, we choose $\delta\Sigma_1(e_9) = x = 1$. Once again, the only solutions

are $e_9 = -1 =$ `0xffffffff` and $e'_9 = 0$. This condition can be deterministically satisfied by choosing W_9 suitably. Finally, we wish to make the following difference zero:

$$\begin{aligned}\delta W_{10} &= -y - \delta f^9_{IF}(x, 0, z) - \delta\Sigma_1(e_9)\\ &= -(-1) - (f_{IF}(e_9 + 1, e_8, e_7 + z) - f_{IF}(e_9, e_8, e_7)) - (\Sigma(e_9 + 1) - \Sigma_1(e_9))\\ &= 1 - f_{IF}(0, e_8, e_7 + z) + f_{IF}(-1, e_8, e_7) - 1\\ &= e_8 - e_7 - z\end{aligned}$$

We have already chosen suitable values for x and y but z is still free. Having worked with the 18-step collisions earlier, we realize that only suitable values for z are 0, +1 and −1.

Equation 13: This equation is the easiest to satisfy. We need $\delta W_{12} = 0$. But $\delta W_{12} = \delta f^{11}_{IF}(0, 0, x)$. If the f_{IF} function chooses its middle argument then we will have the desired. Hence we need to ensure $e_{11} = -1$. This can be done deterministically by choosing W_{11} suitably.

Equation 8: This is a condition which needs to be satisfied. To get $\delta f^7_{MAJ}(0, 0, x) = 0$, it is sufficient to ensure that $a_7 = a_6$. This can be done deterministically by choosing W_7 suitably.

All the conditions are summarized in Table 3.

Table 3. Conditions put on the registers and differential path along with conditions yet to be satisfied.

1	$x = 1$, $y = -1$	5	$e_8 - z - e_7 = 0$ (11)
2	$a_4 = a_3$, $a_5 = -1$	6	$\delta f^6_{MAJ}(0, x, 0) = -z$ (6)
3	$a_7 = a_6$	7	$-x = \delta f^8_{IF}(0, z, y)$ (10)
4	$e_9 = -1$, $e_{11} = -1$	8	$\delta f^{10}_{IF}(0, x, 0) = -z$ (12)

We need to consider three choices for z: 0, 1 and −1. The middle arguments to the δf^6_{MAJ} function are $a_5 + 1$ and a_5, both of which have already been set to specific values 0 and −1 respectively (Cf. Condition 2). This causes difficulty in the satisfaction of Condition 6 in Table 3 for $z = 1$. Hence we consider the other two values for z now.

7.1 When $z = 0$

This is the same 20-step differential path considered in [6]. We now attempt to satisfy conditions 5 to 8 in Table 3.

- Taking $a_6 = a_4$ satisfies condition 6. This can be done by suitably choosing W_6.
- Taking $e_8 = e_7$ satisfies condition 5. This can be done by suitably choosing W_8.
- Taking $e_{10} = 0$ satisfies condition 8. This can be done by suitably choosing W_{10}.

The only condition remaining now is Condition 7 which is $\delta f^8_{IF}(0, 0, -1) = -1$. There is no message freedom left to satisfy this condition. In [6], this condition is let to be free and is satisfied with probability 1/3 by random choices of messages. We now show that it is possible to satisfy even this condition deterministically.

It is clear that if we have $e_8 = 0$ then f_{IF} will select its last argument which has a difference of −1. Thus the output of f_{IF} will be −1 as desired. But we have already chosen W_8 such that $e_8 = e_7$. All the earlier message words starting from W_4 have also been used to satisfy some condition or the other. We now look at the calculation of e_7:

$$\begin{aligned}e_7 &= d_6 + \Sigma_1(e_6) + f_{IF}(e_6, f_6, g_6) + h_6 + K_7 + W_7\\ &= d_6 + a_7 - \Sigma_0(a_6) - f_{MAJ}(a_6, b_6, c_6)\\ &= a_3 + a_7 - \Sigma_0(a_6) - f_{MAJ}(a_6, a_5, a_4)\\ &= a_4 + a_6 - \Sigma_0(a_6) - f_{MAJ}(a_6, -1, a_4)\end{aligned}$$

If we can ensure that $a_6 = a_4 = 0$ then $e_7 = e_8 = 0$ will be deterministic, which in turn will lead to a 20-step collision with probability 1. We used W_4 to get $a_4 = a_3$ earlier. Now we choose the free word W_3 to get $a_3 = 0$. Rest of the conditions remain the same as in [6] and we get 20-step deterministic collisions for SHA-2. Examples of colliding message pairs for 20-step SHA-256 and SHA-512 are given in Section A. The set of conditions on the registers are given as Case 1 in Table 4.

Table 4. Conditions on the registers for 20-step deterministic collisions for SHA-2. Satisfaction of these conditions lead to 20-step collisions for SHA-2 with probability 1. A condition on a_i (or e_i) can be satisfied by suitable choice of W_i. The condition on e_7 in each case gets satisfied automatically when other conditions are met.

Case 1	$x = 1,\ y = -1,$	$z = 0$	
1	$a_3 = a_4 = 0,$	$a_5 = -1,$	$a_6 = a_7 = 0$
2	$e_7 = e_8 = 0,$	$e_{10} = 0,$	$e_9 = e_{11} = -1$
Case 2-A	$x = 1,\ y = -1,$	$z = -1$	
1	$a_3 = a_4 = -1,$	$a_5 = -1,$	$a_6 = a_7 = 0$
2	$e_7 = 0, e_8 = -1,$	$e_9 = -1,$	$e_{10} = e_{11} = -1$
Case 2-B	$x = 1,\ y = -1,$	$z = -1$	
1	$a_3 = a_4 = 0,$	$a_5 = -1,$	$a_6 = a_7 = -1$
2	$e_7 = 1, e_8 = 0,$	$e_9 = -1,$	$e_{10} = e_{11} = -1$

7.2 When $z = -1$

Similar to the case $z = 0$ above, we can determine conditions for 20-step collisions in SHA-2 and deterministically satisfy all the conditions. This time we get two sets of conditions. These are listed as Case 2-A and 2-B in Table 4. Note that this case gives rise to a new 20-step differential path for SHA-2. Colliding pairs of messages satisfying these conditions are given in Section A.

8 Extending a Single Local Collision to Obtain 21-Step Collisions

Using a single local collision to obtain 21-step collisions appears difficult because initial message words start repeating in the recursion of the message expansion this time. In [6], a single local collision spanning from Step 6 to Step 14 is used and a 21-step collision for SHA-256 is obtained probabilistically. Note that the earlier 20-step collisions had the local collision spanning from Step 5 to Step 13. This time the local collision has been slid down by one step. We first describe the method used in [6].

First 5 steps of message expansion for SHA-2 are:

$$\begin{aligned}
W_{16} &= \underline{\sigma_1(W_{14}) + W_9} + \sigma_0(W_1) + W_0 \\
W_{17} &= \sigma_1(W_{15}) + W_{10} + \sigma_0(W_2) + W_1 \\
W_{18} &= \underline{\sigma_1(W_{16})} + W_{11} + \sigma_0(W_3) + W_2 \\
W_{19} &= \sigma_1(W_{17}) + W_{12} + \sigma_0(W_4) + W_3 \\
W_{20} &= \underline{\sigma_1(W_{18})} + W_{13} + \sigma_0(W_5) + W_4
\end{aligned}$$

Since the chosen local collision has 4 consecutive zero message differentials within its span, we have $\delta W_i = 0$ for $i \in \{10, 11, 12, 13\}$. Further, this being the only local collision, messages outside the span of the local collision do not have any difference. Thus, we also have $\delta W_i = 0$ for $i \in \{0, 1, 2, 3, 4, 5, 15\}$. Terms which *may have* non-zero differentials in the above equations are underlined.

All these zero differentials imply that if $\delta\sigma_1(W_{14}) + \delta W_9 = 0$ then the first 5 steps of the message expansion will not produce any difference, and we will have a 21-step collision. Since both W_{14} and W_9 are random, it can be expected that they will cancel the differences in this manner. The probability

for this cancellation to happen is estimated to be about $2^{-17.5}$ in [6]. Since their local collision has probability roughly 1/3, the probability of the 21-step collision is estimated to be approximately 2^{-19}.

We use the same technique for our deterministic 20-step collisions and slide the single local collision one step to attempt a 21-step collision. We first observe that in having the 20-step collisions with probability 1, we have lost some message freedom and consequently, δW_9 is no more random for two of the three cases described in Table 4. This happens for Case 1 and Case 2-B from this table. For proof of this claim, see Section B.

To use the 20-step collision described by Case 1 in Table 4, we need to relax some of the conditions there and obtain some randomness in δW_9. An example of such a relaxation is not to enforce $a_3 = a_4 = 0$, rather only ensure $a_3 = a_4$. This also causes relaxation on the condition on e_7, and the 20-step collision becomes probabilistic now. In fact, this is exactly the same 20-step collision described in [6]. The 21-step collision can now be found for this case as described in [6]. We describe an improvement to the search for messages satisfying $\delta\sigma_1(W_{14}) + \delta W_9 = 0$ a little later.

We note that the conditions in case 2-B of Table 4 cannot be relaxed to obtain randomness in δW_9 and consequently this case can not be used for 21-step collisions. We also note that Case 2-A introduces randomness in δW_9 by default, so we do not need to relax any condition for this case. This is a good case for obtaining 21-step collisions, since it has probability 1 for all the steps other than the cancellation of δW_9 as described above. Next we describe our improved method of searching for suitable messages such that the difference in W_{14} and W_9 cancels the difference in W_{18}.

8.1 Obtaining messages satisfying $\delta\sigma_1(\delta W_{14}) + \delta W_9 = 0$

We have that $\delta W_{14} = W'_{14} - W_{14} = -1$. We expect δW_9 to be random. It is stated in [6] that by random choice of message words, the condition above can be satisfied with probability $2^{-17.5}$. This expectation seems to be based on the randomness of $\delta\sigma_1(W_{14})$. We note that the difference of two σ_1 terms when their inputs differ by -1 is highly non-random.

The choices made in the local collision make the term δW_9 biased towards values which are small in magnitude. A rough idea of the distribution of δW_9 can be had from the following example: We ran the code for 21-step collisions of [6] 5×10^5 times and observed that only 174 times the value of δW_9 came out to be larger than 1000 in magnitude. Further, there were only 334 values larger than 500, 594 values larger than 300 and 1870 values larger than 100.

At the same time, $\sigma_1(W_{14} - 1) - \sigma_1(W_{14})$ is biased towards large magnitudes for random values of W_{14}. In fact, for a large number of points $p \in \{0,1\}^{32}$ there is no solution to the equation $\sigma_1(W_{14} - 1) - \sigma_1(W_{14}) = -\delta W_9 = p$. Interestingly, this equation does not have any solution for W_{14} for even values of p. The distribution of the left hand side of this equation is so non-uniform that there are only 4 values of δW_9 in $\{-300, 300\}$ for which a solution for W_{14} exists. We list these 4 values of δW_9 and corresponding values of W_{14} in Table 5.

Table 5. Some solutions to the equation $\sigma_1(W_{14} - 1) - \sigma_1(W_{14}) + \delta W_9 = 0$ for SHA-256.

No.	δW_9	W_{14}
1	00000041	7fc00000, 80400000
2	00000101	d5000000, 81000000, 7f000000, 2b000000
3	ffffff41	4c400000, b3c00000
4	ffffff01	19000000, 4d000000, b3000000, e7000000

This analysis suggests that a specific suitable value of δW_9 should first be selected and then we should search for corresponding W_{14}. Even if this procedure is used, the probability of being able to get the correct W_{14} is of the order of 2^{-32}. This implies that the search in [6] is not over random messages,

rather a pre-computed value of W_{14} is used for a specific δW_9. From the colliding message pair given in [6], we observe that the value of δW_9 used is `ffffff01` and the corresponding W_{14} is `19000000`. This particular choice of δW_9 occurs with probability $2^{-17.5}$ which corresponds to the estimate given in [6].

We use a speed-up in the search for the correct W_{14}. First we create a list of pairs $(\sigma_1(p)-\sigma_1(p-1), p)$ for all $p \in \{0,1\}^{32}$. We sort this list on the first element. While running the code for 21-step collision, we compute δW_9 and do a binary search over this list. If this value matches with the first element of a pair in the list, then we use the second element to define W_{14}. With this improvement, we obtain a 16 fold improvement to the probability of obtaining the correct δW_9. Since W_{14} is pre-computed, the only probability is in getting the right difference δW_9.

We have extended two types of 20-step collisions to obtain 21-step collisions for SHA-256. One of the local collisions is the Case 1 of Table 4 with some conditions relaxed. As already mentioned, this is the Nikolić-Biryukov local collision [6] having probability 1/3. For this case our method succeeds in finding correct δW_9 with probability roughly $2^{-13.5}$. Thus the overall probability of the 21-step SHA-256 collision is about 2^{-15}.

The second 20-step collision we extend to 21 steps is described by Case 2-A of Table 4. For this case, we could find suitable δW_9 with probability roughly 2^{-17}. Since the probability of the 20-step collision is 1 in this case, we get the 21-step collision with probability roughly 2^{-17}.

9 Conclusions

In this paper we presented a generalized local collision for SHA-2. Using a single instance of this local collision, we obtained 18-step collisions with an arbitrary starting message difference. These collisions hold with probability 1. We then presented two different differential paths for 20-step collisions in SHA-2 both of which hold with probability 1. Finally, we improved on the search for 21-step collisions in SHA-256 increasing the probability of success 16 fold. Apart from the colliding message pairs for different cases and different number of steps for SHA-256, we also show colliding message pairs for up to 20-step SHA-512 for the first time in the literature.

Acknowledgements

We would like to thank anonymous reviewers of ACISP 2008 for giving useful suggestions.

References

1. Florent Chabaud and Antoine Joux. Differential Collisions in SHA-0. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO 1998, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*, pages 56–71. Springer, 1998.
2. Henri Gilbert and Helena Handschuh. Security Analysis of SHA-256 and Sisters. In Mitsuru Matsui and Robert J. Zuccherato, editors, *Selected Areas in Cryptography, 10th Annual International Workshop, SAC 2003, Ottawa, Canada, August 14-15, 2003, Revised Papers*, volume 3006 of *Lecture Notes in Computer Science*, pages 175–193. Springer, 2003.
3. Philip Hawkes, Michael Paddon, and Gregory G. Rose. On Corrective Patterns for the SHA-2 Family. *Cryptology eprint Archive*, August 2004. Available at `http://eprint.iacr.org/2004/207`.
4. Florian Mendel, Norbert Pramstaller, Christian Rechberger, and Vincent Rijmen. Analysis of Step-Reduced SHA-256. In Matthew J. B. Robshaw, editor, *Fast Software Encryption, 13th International Workshop, FSE 2006, Graz, Austria, March 15-17, 2006, Revised Selected Papers*, volume 4047 of *Lecture Notes in Computer Science*, pages 126–143. Springer, 2006.
5. Florian Mendel, Norbert Pramstaller, Christian Rechberger, and Vincent Rijmen. Analysis of Step-Reduced SHA-256. *Cryptology eprint Archive*, March 2008. Available at `http://eprint.iacr.org/2008/130`.
6. Ivica Nikolić and Alex Biryukov. Collisions for Step-Reduced SHA-256. In Kaisa Nyberg, editor, *Fast Software Encryption 2008*, volume Pre-proceedings version of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2008.
7. Somitra Kumar Sanadhya and Palash Sarkar. New Local Collisions for the SHA-2 Hash Family. In Kil-Hyun Nam and Gwangsoo Rhee, editors, *Information Security and Cryptology - ICISC 2007, 10th International Conference, Seoul, Korea, November 29-30, 2007, Proceedings*, volume 4817 of *Lecture Notes in Computer Science*, pages 193–205. Springer, 2007.

8. Somitra Kumar Sanadhya and Palash Sarkar. Attacking Reduced Round SHA-256. In Steven Bellovin and Rosario Gennaro, editors, *Applied Cryptography and Network Security - ACNS 2008, 6th International Conference, New York, NY, June 03-06, 2008, Proceedings*, volume To appear of *Lecture Notes in Computer Science*. Springer, 2008.
9. Victor Shoup, editor. *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*. Springer, 2005.
10. Secure Hash Standard. *Federal Information Processing Standard Publication 180-2*. U.S. Department of Commerce, National Institute of Standards and Technology(NIST), 2002. Available at `http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf`.
11. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding Collisions in the Full SHA-1. In Shoup [9], pages 17–36.
12. Xiaoyun Wang and Hongbo Yu. How to Break MD5 and Other Hash Functions. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer, 2005.
13. Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin. Efficient Collision Search Attacks on SHA-0. In Shoup [9], pages 1–16.

A Colliding message pairs

Table 6. Colliding message pair for 18-step SHA-256 with standard IV. These messages follow the differential path of Table 1 with $x =$ `b875622d`, $y =$ `e4bfa8a5`, $z =$ `0`.

W_1	0-7	e6f590fc	58f290f9	53ac42fa	3a7c9ee6	30dc2357	2ee1b785	0abebaa2	f61d8c82
	8-15	147e048b	501bc66b	75a3d802	3c9ca879	8f454627	8b3ff382	55a4de5a	a3e613ea
W_2	0-7	e6f590fc	58f290f9	53ac42fa	f2f20113	87b66fa8	77801baf	57d16843	9da87bd1
	8-15	9d408abf	501bc66b	75a3d802	8427464c	8f454627	8b3ff382	55a4de5a	a3e613ea

Table 7. Colliding message pair for 18-step SHA-256 with standard IV. These messages follow the differential path of Table 1 with $x =$ `60097ffe`, $y =$ `a5dba93b`, $z = x$.

W_1	0-7	9868945f	43e023b2	672e208d	d5c4df8c	294d3db9	a7bbabdc	20ff800b	76bad5a7
	8-15	1e09c4ef	e778eba6	406fc989	0f0f6380	b91e9155	7965e503	f4c4c13a	57301b93
W_2	0-7	9868945f	43e023b2	672e208d	35ce5f8a	d3a2bd52	63ff4094	c0be1992	78b4cf6e
	8-15	3407e934	e778eba6	406fc989	af05e382	b91e9155	7965e503	f4c4c13a	57301b93

Table 8. Colliding message pair for 18-step SHA-256 with standard IV. These messages follow the differential path of Table 1 with $x =$ `ec1fe92d`, $y =$ `a01beee5`, $z = -x =$ `13e016d3`.

W_1	0-7	eda26041	7ea8c572	74155b82	d4d697e9	a8c75b74	cdc3dba6	b6bc5d2f	2b2fc241
	8-15	51d8186a	416d969f	0eb5cd0c	7044ff7e	0731645f	464c0913	d7d58642	896f7bdb
W_2	0-7	eda26041	7ea8c572	74155b82	c0f68116	d75df145	9cc03075	7fcf9d26	ef2fe209
	8-15	5678aad9	416d969f	0eb5cd0c	84251651	0731645f	464c0913	d7d58642	896f7bdb

B Impossibility of some 21-step differential paths

We now show that by one step sliding of a 20-step collision for Cases 1 and 2-B of Table 4, we cannot obtain a 21-step collision for SHA-2.

Table 9. Colliding message pair for 20-step SHA-256 with standard IV. These messages follow the differential path of Table 1 with $x =$ `1`, $y =$ `-1`, $z =$ `0`. These messages satisfy Case 1 of Table 4.

W_1	0-7	17cf6aff	89e9ba13	c90b578d	b0db265f	ba7c84b0	a24899eb	980f02b7	627ec4ec
	8-15	efaf5d4e	4cb1ae36	157b67d7	3cdc84e2	d9d4c9ac	0c32f8ca	5a262489	86f0592b
W_2	0-7	17cf6aff	89e9ba13	c90b578d	b0db265f	ba7c84b0	a24899ec	93ef0235	6e9ec56e
	8-15	efaf5d4d	4cb1ae36	157b67d7	3cdc84e2	d9d4c9ac	0c32f8c9	5a262489	86f0592b

Table 10. Colliding message pair for 20-step SHA-256 with standard IV. These messages follow the differential path of Table 1 with $x =$ `1`, $y =$ `-1`, $z =$ `-1`. These messages satisfy Case 2-A of Table 4.

W_1	0-7	5a603c44	0f5fdd15	69e8c2a4	1754c271	60518701	feef6b5f	c7f50d13	fdc492ca
	8-15	d5d49f53	d4c9d37f	bf796ac4	aaf3823e	a24e8e62	8d8898c8	fc4456f3	8d557ae5
W_2	0-7	5a603c44	0f5fdd15	69e8c2a4	1754c271	60518701	feef6b60	d3d50e93	f9a49248
	8-15	d2326157	d4c9d37f	bf796ac4	aaf3823e	a24e8e62	8d8898c7	fc4456f3	8d557ae5

Table 11. Colliding message pair for 20-step SHA-256 with standard IV. These messages follow the differential path of Table 1 with $x =$ `1`, $y =$ `-1`, $z =$ `-1`. These messages satisfy Case 2-B of Table 4.

W_1	0-7	f9b685e2	4e18d30f	066c47b9	380fb811	364c2fb9	085aafac	8d999930	17532d80
	8-15	2e182279	92e6647c	2263df08	aaf3823e	46efda92	400ed683	56bba6ad	c7133d81
W_2	0-7	f9b685e2	4e18d30f	066c47b9	380fb811	364c2fb9	085aafad	917999b0	1b332dff
	8-15	323822f9	92e6647c	2263df08	aaf3823e	46efda92	400ed682	56bba6ad	c7133d81

Table 12. Colliding message pair for 21-step SHA-256 with standard IV. These messages follow the differential path of Table 1 with $x =$ `1`, $y =$ `-1`, $z =$ `0`. For these messages, $\delta W_9 =$ `ffffff41`.

W_1	0-7	f1497cd4	7fe4857c	df070eea	a035b751	ece48886	f42a8fc9	fb1fe099	2052dc45
	8-15	79f17c4b	8b1ee7ab	85da1bdc	c07222ad	3ccee34f	be164fd8	b3c00000	571b5a2f
W_2	0-7	f1497cd4	7fe4857c	df070eea	a035b751	ece48886	f42a8fc9	fb1fe09a	e233a2c1
	8-15	75d17add	8b1ee6ec	85da1bdc	c07222ad	3ccee34f	be164fd8	b3ffffff	571b5a2f

Table 13. Colliding message pair for 21-step SHA-256 with standard IV. These messages follow the differential path of Table 1 with $x =$ `1`, $y =$ `-1`, $z =$ `-1`. For these messages $\delta W_9 =$ `ffffe191`.

W_1	0-7	4158ecc7	3a3ffe61	ba7149f0	ed452440	4d9ab924	f016459f	22f5578c	c56333c1
	8-15	ff1941ff	19b8055b	fb2876ba	ca4d6044	8d41a28d	8194372b	7e100000	5240bb72
W_2	0-7	4158ecc7	3a3ffe61	ba7149f0	ed452440	4d9ab924	f016459f	22f5578d	c1433241
	8-15	fb39427d	19b7e6ec	fb2876ba	ca4d6044	8d41a28d	8194372b	7e0fffff	5240bb72

Table 14. Colliding message pair for 18-step SHA-512 with standard IV. These messages follow the differential path of Table 1 with $x =$ `373c5915a7e8cd1a`, $y =$ `bac8b5823e5656cb`, $z =$ `0`.

W_1	0-3	eb1eb59ecf0b3342	e205af0b51f81569	62759b3c1cbbfb60	d94c8594e1468081
	4-7	21c555e5eb4a74ff	466534f9e5c4dd20	2c55b4e93bd76391	0dffe3bf30abcb91
	8-11	f2051be933e8762d	57417ceddcf050ed	a7116f111de85809	5ed73acd8290c14c
	12-15	d60a9f4bbcadc128	54ea8bb9f46b36ef	3ac446634c581411	cbf82d9f9493f84d
W_2	0-3	eb1eb59ecf0b3342	e205af0b51f81569	62759b3c1cbbfb60	1088deaa892f4d9b
	4-7	537807237bcf8a95	1b8ab570b1112066	74a8eaf30dcdac7e	1e2394a74a7a9b86
	8-11	aa49e4d0d4cfdbc7	57417ceddcf050ed	a7116f111de85809	279ae1b7daa7f432
	12-15	d60a9f4bbcadc128	54ea8bb9f46b36ef	3ac446634c581411	cbf82d9f9493f84d

Table 15. Colliding message pair for 18-step SHA-512 with standard IV. These messages follow the differential path of Table 1 with $x =$ `8000000000000000`, $y =$ `7ffffff7df000000`, $z = x$.

W_1	0-3	d5d231bd0aee1913	988d1c29544b4e23	77641612867ae0ba	c3b0ce9aee99e947
	4-7	4a65130318dcc860	2ffd17efc9d7826d	8773e9f2c175c1c7	d8dcc93460a556ba
	8-11	f0f055a560f90591	e586e628eca6fdaa	74763eadbd1b619b	63faa21560edc065
	12-15	f97c799fa4a01d9f	d119e52a631aa6ec	16e76e09c000af74	8a32144bfd97630e
W_2	0-3	d5d231bd0aee1913	988d1c29544b4e23	77641612867ae0ba	43b0ce9aee99e947
	4-7	4a62f2faf79cc860	b3ceb7df69bee62d	877609f2a2b5c1c7	d8dcc92c40a556ba
	8-11	70ee75ad82390591	6586e628eca6fdaa	74763eadbd1b619b	e3faa21560edc065
	12-15	f97c799fa4a01d9f	d119e52a631aa6ec	16e76e09c000af74	8a32144bfd97630e

Table 16. Colliding message pair for 18-step SHA-512 with standard IV. These messages follow the differential path of Table 1 with $x =$ `c0145fc22e2f8106`, $y =$ `70df70d99098ebeb`, $z = -x =$ `3feba03dd1d07efa`.

W_1	0-3	649447f9cd22cbc1	b56e3ca4d7d16a57	5bd5d12d24969ab4	0e2ea85d485ad0f9
	4-7	87afbac32285a4a7	69bf436266be288e	46aa45bd104ef93c	370586b96422ce9b
	8-11	d7534fa56ee15811	423b664e4392c00e	50133367aa291e21	09691402f481d4b4
	12-15	06a12448353c4575	358db4301a231c4c	f5d1794c82015a66	c1464f23262776b4
W_2	0-3	649447f9cd22cbc1	b56e3ca4d7d16a57	5bd5d12d24969ab4	ce43081f768a51ff
	4-7	e4ac8281951462fe	97b78dd2e69b6cec	114335dbcd070889	0e21b1394492621b
	8-11	9a5eb6e1de5fd1a0	423b664e4392c00e	50133367aa291e21	4954b440c65253ae
	12-15	06a12448353c4575	358db4301a231c4c	f5d1794c82015a66	c1464f23262776b4

Table 17. Colliding message pair for 20-step SHA-512 with standard IV. These messages follow the differential path of Table 1 with $x =$ `1`, $y =$ `-1`, $z =$ `0`. These messages satisfy Case 1 of Table 4.

W_1	0-3	6058ceb9a1077eb2	a4cf55c2b1bb8fce	784193965385ff3b	7463839e2fe1d369
	4-7	88168bd7f18e72a7	2c4bba75ff7d74e6	2aebc8365586a02d	c3506e0db562134a
	8-11	73a70156e11e07c2	9947f674a891d76c	1023901ef5eace3b	b258c2dde4e508ac
	12-15	f644b8df45fbe4d8	bb87a43dc0674b95	61d9c1b117244b44	e2264ccbf7bf427e
W_2	0-3	6058ceb9a1077eb2	a4cf55c2b1bb8fce	784193965385ff3b	7463839e2fe1d369
	4-7	88168bd7f18e72a7	2c4bba75ff7d74e7	2af008365506a02d	c34c2e0db5e2134a
	8-11	73a70156e11e07c1	9947f674a891d76c	1023901ef5eace3b	b258c2dde4e508ac
	12-15	f644b8df45fbe4d8	bb87a43dc0674b94	61d9c1b117244b44	e2264ccbf7bf427e

Table 18. Colliding message pair for 20-step SHA-512 with standard IV. These messages follow the differential path of Table 1 with $x =$ `1`, $y =$ `-1`, $z =$ `-1`. These messages satisfy Case 2-A of Table 4.

W_1	0-3	1c99041525eeeeb3	7dfc74f74bab1a89	aaca442cddb37351	21d1684a782a5b87
	4-7	3d374aed94c9d766	296c28f080eced7a	62f73e6df90ce266	d4c85286272c52c1
	8-11	e2d8e832fb623115	5c43e3fc9bee94c3	5ef6f726192a4213	aaf3823c2a004b1f
	12-15	fa18ffe92868d117	8584328bd3146ed0	c3ce87104858e6cb	6dc9cd6519344c6a
W_2	0-3	1c99041525eeeeb3	7dfc74f74bab1a89	aaca442cddb37351	21d1684a782a5b87
	4-7	3d374aed94c9d766	296c28f080eced7b	62fafe6df88ce264	d4cc928628ac52c0
	8-11	f73a261982122135	5c43e3fc9bee94c3	5ef6f726192a4213	aaf3823c2a004b1f
	12-15	fa18ffe92868d117	8584328bd3146ecf	c3ce87104858e6cb	6dc9cd6519344c6a

Table 19. Colliding message pair for 20-step SHA-512 with standard IV. These messages follow the differential path of Table 1 with $x =$ `1`, $y =$ `-1`, $z =$ `-1`. These messages satisfy Case 2-B of Table 4.

W_1	0-3	7f446c831ae44cd8	fe2fdbf87099c0da	5d260ebc8025368b	2c24db0985d910d7
	4-7	71ec2db073b48f6a	4c95a6faaa6dd1a5	1f12885da19643e6	3ac1f1ef5ef38304
	8-11	03cf1b75849b5222	1d5c1436e6417e2a	1b619cf7e4dfde50	aaf3823c2a004b1f
	12-15	f3fd487aea68fbd9	fcf6a431bae731ff	aba4536a50179e3d	837c2afdff067b28
W_2	0-3	7f446c831ae44cd8	fe2fdbf87099c0da	5d260ebc8025368b	2c24db0985d910d7
	4-7	71ec2db073b48f6a	4c95a6faaa6dd1a6	1f1e485d9f1643e7	3abe31ef5f7382ff
	8-11	03d35b75851b5221	1d5c1436e6417e2a	1b619cf7e4dfde50	aaf3823c2a004b1f
	12-15	f3fd487aea68fbd9	fcf6a431bae731fe	aba4536a50179e3d	837c2afdff067b28

First note that the cases described in Table 4 are for a local collision spanning from Steps 5 to 13. Now that we have shifted the local collision by one step to span it from Step 6 to Step 14, all the conditions of Table 4 also need to be shifted by one index. Hence a condition on a_i in this table will become a condition on a_{i+1} for our present case.

B.1 Case 1: ($x = 1$, $y = -1$, $z = 0$):

We have $a_4 = a_5 = a_7 = a_8 = 0$, $a_6 = -1$, $e_8 = e_9 = e_{11} = 0$, $e_{10} = e_{12} = -1$. From (9), $\delta W_9 = -\delta f^8_{IF}(0, -1, 1) - \delta\Sigma_1(e_8)$. Simplifying this we get:

$$\begin{aligned}\delta W_9 &= f_{IF}(e_8, e_7, e_6) - f_{IF}(e_8, e_7 - 1, e_6 + 1) + \Sigma_1(e_8) - \Sigma_1(e_8)\\ &= f_{IF}(0, e_7, e_6) - f_{IF}(0, e_7 - 1, e_6 + 1) + \Sigma_1(0) - \Sigma_1(0)\\ &= e_6 - (e_6 + 1) = -1\end{aligned}$$

We now need a pair of message words W_{14} and $W'_{14} = W_{14} - 1$ such that $\delta W_{14} = -\delta W_9 = 1$. We note that there does not exist any 32-bit word W_{14} which can satisfy this condition.

B.2 Case 2-B: ($x = 1$, $y = -1$, $z = -1$):

We have $a_4 = a_5 = 0$, $a_6 = a_7 = a_8 = -1$, $e_8 = 1$, $e_9 = 0$, $e_{10} = e_{11} = e_{12} = -1$. From (9), $\delta W_9 = -\delta f^8_{IF}(-1, -1, 1) - \delta\Sigma_1(e_8)$. Simplifying this we get:

$$\begin{aligned}\delta W_9 &= f_{IF}(e_8, e_7, e_6) - f_{IF}(e_8 - 1, e_7 - 1, e_6 + 1) + \Sigma_1(e_8) - \Sigma_1(e_8 - 1)\\ &= f_{IF}(1, e_7, e_6) - f_{IF}(0, e_7 - 1, e_6 + 1) + \Sigma_1(1) - \Sigma_1(0)\\ &= f_{IF}(1, e_7, e_6) - f_{IF}(0, e_7 - 1, e_6 + 1) + \Sigma_1(1)\end{aligned}$$

The f_{IF} function selects its output bit from either its second or third argument. Since the first arguments of the two f_{IF} terms differ only at the lowest bit, the output from the difference of the two f_{IF} terms can only be +1 or 0 or -1. The last term $\Sigma_1(1)$ is a constant quantity. We now need a pair of message words W_{14} and $W'_{14} = W_{14} - 1$ such that $\delta W_{14} = -\delta W_9$. We note that there does not exist any 32-bit word W_{14} which can satisfy this condition.