New Collision Attacks against Up to 24-Step SHA-2

(Extended Abstract)

Somitra Kumar Sanadhya* and Palash Sarkar

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

Abstract. In this work, we provide new and improved attacks against 22, 23 and 24-step SHA-2 family using a local collision given by Sanadhya and Sarkar (SS) at ACISP '08. The success probability of our 22-step attack is 1 for both SHA-256 and SHA-512. The computational efforts for the 23-step and 24-step SHA-256 attacks are respectively $2^{11.5}$ and $2^{28.5}$ calls to the corresponding step reduced SHA-256. The corresponding values for the 23 and 24-step SHA-512 attack are respectively $2^{16.5}$ and $2^{32.5}$ calls. Using a look-up table having 2^{32} (resp. 2^{64}) entries the computational effort for finding 24-step SHA-256 (resp. SHA-512) collisions can be reduced to $2^{15.5}$ (resp. $2^{22.5}$) calls. We exhibit colliding message pairs for 22, 23 and 24-step SHA-256 and SHA-512. This is the *first* time that a colliding message pair for 24-step SHA-512 is provided. The previous work on 23 and 24-step SHA-2 attacks is due to Indesteege et al. and utilizes the local collision presented by Nikolić and Biryukov (NB) at FSE '08. The reported computational efforts are 2^{18} and $2^{28.5}$ for 23 and 24-step SHA-256 respectively and $2^{43.9}$ and 2^{53} for 23 and 24-step SHA-512. The previous 23 and 24-step attacks first constructed a pseudo-collision and later converted it into a collision for the reduced round SHA-2 family. We show that this two step procedure is unnecessary. Although these attacks improve upon the existing reduced round SHA-2 attacks, they do not threaten the security of the full SHA-2 family.

Keywords: Cryptanalysis, SHA-2 hash family, reduced round attacks.

1 Introduction

Cryptanalysis of SHA-2 family has recently gained momentum due to the important work of Nikolić and Biryukov [6]. Prior work on finding collisions for step reduced SHA-256 was done in [4,5] and [8]. These earlier works used local collisions valid for the XOR linearized version of SHA-256 from [2] and [7]. On the other hand, the work [6] used a local collision which is valid for the actual SHA-256.

The authors in [6] developed techniques to handle nonlinear functions and the message expansion of SHA-2 to obtain collisions for up to 21-step SHA-256. The 21-step attack of [6] succeeded with probability 2^{-19}. Using similar

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

D.R. Chowdhury, V. Rijmen, and A. Das (Eds.): INDOCRYPT 2008, LNCS 5365, pp. 91–103, 2008.

techniques, but utilizing a different local collision, [11] showed an attack against 20-step SHA-2 which succeeds with probability one and an attack against 21-step SHA-256 which succeeds with probability 2^{-15}. Further work [9] developed collision attacks against 21-step SHA-2 family which succeeds with probability one. Very recently, Indesteege et al. [3] have developed attacks against 23 and 24 step SHA-2 family. They utilize the local collision from [6] in these attacks.

Our Contributions. Our contributions in terms of the number of steps attacked and the success probability of these attacks are as follows.

– We describe the first *deterministic* attack against 22-step SHA-256 and SHA-512.
– We describe new attacks against 23 and 24-step SHA-256 and SHA-512.
  - The complexity of the 23-step attack for both SHA-256 and SHA-512 is improved in comparison to the existing 23-step attacks of [3].
  - The complexity of 24-step SHA-512 attack is improved in comparison to the existing attack of [3]. In fact, improving the complexity to $2^{32.5}$ from the earlier reported 2^{53} allows us to provide the *first* message pair which collides for 24-step SHA-512.

Table 1. Summary of results against reduced SHA-2 family. Effort is expressed as either the probability of success or as the number of calls to the respective reduced round hash function.

Work	Hash Function	Steps	Effort		Local Collision utilized	Attack Type	Example provided
			Prob.	Calls			
[4,5]	SHA-256	18		*	GH [2]	Linear	yes
[8]	SHA-256	18		**	SS_5 [7]	”	yes
[6]	SHA-256	20	$\frac{1}{3}$		NB [6]	Non-linear	yes
		21	2^{-19}		”	”	yes
[11]	SHA-256/512	18,20	1	1	SS [11]	”	yes
	SHA-256	21	2^{-15}		”	”	yes
[9]	SHA-256/512	21	1	1	”	”	yes
[3]	SHA-256	23		2^{18}	NB [6]	”	yes
		24		$2^{28.5}$	”	”	yes
	SHA-512	23		$2^{43.9}$	”	”	yes
		24		2^{53}	”	”	**no**
This work	SHA-256/SHA-512	22	1	1	SS [11]	”	yes
	SHA-256	23		$2^{11.5}$	”	”	yes
		24		$2^{28.5}$	”	”	yes
		24		$2^{15.5}$ †	”	”	no
	SHA-512	23		$2^{16.5}$	”	”	yes
		24		$2^{32.5}$	”	”	**yes**
		24		$2^{22.5}$ ‡	”	”	no

* It is mentioned in [4,5] that the effort is 2^0 but no details are provided.
** Effort is given as running a C-program for about 30–40 minutes on a standard PC.
† A table containing 2^{32} entries, each entry of size 8 bytes, is required.
‡ A table containing 2^{64} entries, each entry of size 16 bytes, is required.

- Using a table lookup, the complexity of the 24-step SHA-256 attack is improved in comparison to the existing 24-step attack of [3]. The table contains 2^{32} entries with each entry of size 8 bytes. Similary, the complexity of the 24-step SHA-512 attack is also improved using a table lookup. For this case, the table lookup has 2^{64} entries each entry of 16 bytes.
- Examples of Colliding message pairs are provided for 22, 23 and 24-step SHA-256 and SHA-512.

Our contributions to the methodology of the attacks are as follows.

- We use a different local collision for our 22, 23 and 24-step attacks. The earlier work [3] uses the local collision from [6] while we use a local collision from [11].
- The work in [3] describes 23 and 24-step collisions as a two-part procedure– first obtain a pseudo-collision and then convert it into a collision. In contrast, our analysis is direct and shows that such a two-part description is unnecessary.
- Details of a required "guess-then-determine algorithm" to solve a non-linear equation arising in the 24-step attack are provided in this work. A suggestion for a similar algorithm is given in [3] but no details are provided. There are two algorithms– one for SHA-256 and the other for SHA-512.

A summary of results on collision attacks against reduced SHA-2 family is given in Table 1.

2 Preliminaries

In this paper we use the following notation:

- Message words: $W_i \in \{0,1\}^n$, $W'_i \in \{0,1\}^n$; n is 32 for SHA-256 and 64 for SHA-512.
- Colliding message pair:$\{W_0, W_1, W_2, \ldots W_{15}\}$ & $\{W'_0, W'_1, W'_2, \ldots W'_{15}\}$.
- Expanded message pair:$\{W_0, W_1, W_2, \ldots W_{r-1}\}$ & $\{W'_0, W'_1, W'_2, \ldots W'_{r-1}\}$. The number of steps r is 64 for SHA-256 and 80 for SHA-512.
- The internal registers for the two messages at step i: $\mathrm{REG}_i = \{a_i, \ldots, h_i\}$ and $\mathrm{REG}'_i = \{a'_i, \ldots, h'_i\}$.
- $\mathrm{ROTR}^k(x)$: Right rotation of an n-bit string x by k bits.
- $\mathrm{SHR}^k(x)$: Right shift of an n-bit string x by k bits.
- $\oplus$: bitwise XOR; $+, -$: addition and subtraction modulo 2^n.
- $\delta X = X' - X$ where X is an n-bit quantity.
- $\delta\Sigma_1(x) = \Sigma_1(e'_i) - \Sigma_1(e_i) = \Sigma_1(e_i + x) - \Sigma_1(e_i)$.
- $\delta\Sigma_0(x) = \Sigma_0(a'_i) - \Sigma_0(a_i) = \Sigma_0(a_i + x) - \Sigma_0(a_i)$.
- $\delta f^i_{MAJ}(x, y, z) = f_{MAJ}(a_i + x, b_i + y, c_i + z) - f_{MAJ}(a_i, b_i, c_i)$.
- $\delta f^i_{IF}(x, y, z) = f_{IF}(e_i + x, f_i + y, g_i + z) - f_{IF}(e_i, f_i, g_i)$.

2.1 SHA-2 Hash Family

Eight registers are used in the evaluation of SHA-2. 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 $(a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i)$. For more details, see [1].

By the form of the step update function, we have the following relation.

Cross Dependence Equation (CDE)

$$e_i = a_i + a_{i-4} - \Sigma_0(a_{i-1}) - f_{MAJ}(a_{i-1}, a_{i-2}, a_{i-3}). \quad (1)$$

Later, we make extensive use of this relation. Note that a special case of this equation was first utilized in §6.1 of [11]. The equation in the form above was used in [9]. This equation can be used to show that the SHA-2 state update can be rewritten in terms of only one state variable. This fact was later observed in [3] independently.

Table 2. The 9-step Sanadhya-Sarkar local collision [11] used in the present work. Our deterministic 22-step attack and the probabilistic 23 and 24-step attacks use unequal message word differences to achieve the same differential path.

Step	δW_i		Register differences							
	I	II	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	0
i	1	1	1	0	0	0	1	0	0	0
$i+1$	-1	δW_{i+1}	0	1	0	0	-1	1	0	0
$i+2$	δW_{i+2}	0	0	0	1	0	-1	-1	1	0
$i+3$	δW_{i+3}	δW_{i+3}	0	0	0	1	0	-1	-1	1
$i+4$	0	0	0	0	0	0	1	0	-1	-1
$i+5$	0	0	0	0	0	0	0	1	0	-1
$i+6$	0	0	0	0	0	0	0	0	1	0
$i+7$	δW_{i+7}	0	0	0	0	0	0	0	0	1
$i+8$	-1	-1	0	0	0	0	0	0	0	0

3 Nonlinear Local Collision for SHA-2

We use two variations of a 9-step non-linear local collision for our attacks. This local collision was given recently by Sanadhya and Sarkar [11]. This local collision starts by introducing a perturbation message difference of 1 in the first message word. Next eight message words are chosen suitably to obtain the desired differential path. Table 2 shows the local collision used. The message word differences are different for the two variations of the local collision. Columns headed I and II under δW_i in Table 2 show the message word differences for the first and the second variations of the local collision respectively.

In the local collision, 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 $(a_i, \ldots, h_i)$.

4 The Deterministic 22-Step SHA-2 Attack

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. We use a similar method for our attack but this time we use the local collision of Table 2 spanning from Step 7 to Step 15. Message words are given by Column (II). The SHA-2 design has freedom of message words W_0 to W_{15}. Since the local collision spans this range only, we can deterministically satisfy all the required conditions. The message words after Step 16 are generated by message expansion. The local collision is chosen in such a way that the message expansion produces no difference in words W_i and W_i' for $i \in \{16, 17, \ldots 21\}$. This results in a deterministic 22-step attack. We explain this fact below.

First of all, note that the local collision starts from Step 7. It can be seen from the structure of the local collision that $\delta W_7 = 1$ and $\delta W_9 = \delta W_{11} = \delta W_{12} = \delta W_{13} = \delta W_{14} = 0$. In addition, δW_{15} is -1. Messages outside the span of the local collision are taken to have zero differentials. Therefore $\delta W_i = 0$ for $i \in \{0, 1, 2, 3, 4, 5, 6\}$. Consider the first 6 steps of message expansion for SHA-2 next.

$$\left.\begin{aligned}
W_{16} &= \sigma_1(W_{14}) + \ W_9 + \sigma_0(W_1) + W_0, \\
W_{17} &= \underline{\sigma_1(W_{15}) + W_{10}} + \sigma_0(W_2) + W_1, \\
W_{18} &= \sigma_1(W_{16}) + W_{11} + \sigma_0(W_3) + W_2, \\
W_{19} &= \underline{\sigma_1(W_{17})} + W_{12} + \sigma_0(W_4) + W_3, \\
W_{20} &= \underline{\sigma_1(W_{18})} + W_{13} + \sigma_0(W_5) + W_4, \\
W_{21} &= \underline{\sigma_1(W_{19})} + W_{14} + \sigma_0(W_6) + W_5.
\end{aligned}\right\} \tag{2}$$

Terms which *may have* non-zero differentials in the above equations are underlined. To obtain 22-step collisions in SHA-2, it is sufficient to ensure that $\delta\{\sigma_1(W_{15}) + W_{10}\} = 0$ so that $\delta W_{17} = 0$. This also ensures that next 4 steps of the message expansion do not produce any difference, and we have a 22-step collision. By using the local collision described earlier, it is possible to deterministically satisfy the condition $\delta\{\sigma_1(W_{15}) + W_{10}\} = 0$. Further details are available in [10].

5 A General Idea for Obtaining 23 and 24-Step SHA-2 Collisions

Obtaining deterministic collisions up to 22 steps did not require the (single) local collision to extend beyond step 15. For obtaining collisions for more number of steps, we will need to start the local collision at Step 8 (or farther) and hence the local collision will end at Step 16 (or farther). This will require us to analyze the message expansion more carefully.

For obtaining collisions up to 22 steps, we also needed to consider message expansion. But, following Nikolić and Biryukov, we ensured that there were no differences in message words from Step 16 onwards. However, now that we consider the local collision to end at Step 16 (or farther), this will necessarily mean that one or more δW_i (for $i \geq 16$) will be non-zero. This will require a modification of the Nikolić-Biryukov strategy. Instead of requiring $\delta W_i = 0$ for $i \geq 16$, we will require $\delta W_i = 0$ for a few i's after the local collision ends. So, supposing that the local collision ends at Step 16 and we want a 23-step collision, then δW_{16} is necessarily -1 and we will require $\delta W_{17} = \cdots = \delta W_{22} = 0$.

5.1 Satisfying Conditions on the Differential Path

Conditions on δW_{i+2}, δW_{i+3} and δW_{i+4} shown in Table 2 give rise to the following conditions on the values of λ, γ and μ.

$$\left.\begin{aligned} \delta W_{i+2} = \delta_1 &= -1 - \Sigma_1(\mu - 1) + \Sigma_1(\mu) - f_{IF}(\mu - 1, 0, \gamma + 1) \\ &\quad + f_{IF}(\mu, -1, \gamma + 1) \\ \delta W_{i+3} = \delta_2 &= -\Sigma_1(\lambda - 1) + \Sigma_1(\lambda) - f_{IF}(\lambda - 1, \mu - 1, 0) \\ &\quad + f_{IF}(\lambda, \mu, -1) \\ 1 &= -f_{IF}(\lambda - 1, \lambda - 1, \mu - 1) + f_{IF}(\lambda - 1, \lambda, \mu). \end{aligned}\right\} \tag{3}$$

Similar equations for the Nikolić-Biryukov differential path have been reported in [3] and a method for solving them has been discussed. The method to solve these equation is different for SHA-256 and for SHA-512. We discuss the exact details about solving them later. In describing our attacks on the SHA-2 family, we assume that some solutions to these equations have been obtained. These solutions are required to obtain colliding message pairs for the hash functions.

6 23-Step SHA-2 Collisions

We show that by suitably placing a local collision of the type described in Column (I) of Table 2 and using proper values for α, γ and μ, it is possible to obtain 23-step collisions for SHA-2.

6.1 Case $i = 8$

The local collision is started at $i = 8$ and ends at $i = 16$. Setting $\beta = \overline{\alpha}$, $u = 0$ and $\delta_1 = 0$, we need to choose a suitable value for δ_2 which is the value of $\delta W_{i+3} = \delta W_{11}$. For this case, we let $\delta = \delta_2$.

Since the local collision ends at Step 16, it necessarily follows that $\delta W_{16} = -1$. Consequently, we need to consider δW_{18} to ensure that it is zero. Since the collision starts at $i = 8$, all δW_j for $0 \leq j \leq 7$ are zero. Consequently, we can write $\delta W_{18} = \delta\sigma_1(W_{16}) + \delta W_{11}$, where $\delta\sigma_1(W_{16}) = \sigma_1(W_{16} - 1) - \sigma_1(W_{16})$. So, for δW_{18} to be zero, we need $\delta W_{11} = -\delta\sigma_1(W_{16})$, so that δW_{11} should be one of the values which occur in the distribution of $\sigma_1(W) - \sigma_1(W - 1)$ for some W.

Obtaining proper values for the constants only ensures that the local collision holds from Steps i to $i + 8$ as expected. It does not, however, guarantee that the reduced round collision holds. In the present case, we need to have δW_{18} to be

Table 3. Values of a and e register for the δWs given by Column (I) of Table 2 to hold. We have $\beta = \overline{\alpha}$ and using CDE, $\lambda = \beta + \alpha - \Sigma_0(\beta) - f_{MAJ}(\beta, -1, \alpha) = -\Sigma_0(\overline{\alpha})$. The value of u is either 0 or 1. Thus, the independent quantities are α, γ and μ.

index	$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$	$i+4$	$i+5$	$i+6$
a	α	α	-1	β	β				
e	γ	$\gamma+1$	-1	μ	λ	$\lambda-1$	-1	-1	$-1-u$

zero. This will happen only if W_{16} takes a value such that $\sigma_1(W_{16}-1)-\sigma_1(W_{16})$ is equal to $-\delta$. This can be ensured probabilistically in the following manner. Let the frequency of δ used in the attack be freq_δ. This means that trying approximately freq_δ possible random choices of W_0 and W_1, we expect a proper value of W_{16} and hence, a 23-step collision for SHA-2. We discuss the cases of SHA-256 and SHA-512 separately later.

Since $i = 8$, from Table 3, we see that a_6 to a_{10} get defined and e_6 to e_{14} get defined. Using CDE, the values of e_9 down to e_6 is set by fixing values of a_5 down to a_2. In other words, the values of a_2 to a_{10} are fixed. Now, consider

$$e_{14} = \Sigma_1(e_{13}) + f_{IF}(e_{13}, e_{12}, e_{11}) + a_{10} + e_{10} + K_{14} + W_{14}.$$

Note that in this equation all values other than W_{14} have already been fixed. So, W_{14} and hence $\sigma_1(W_{14})$ is also fixed. Now, from the update function of the a register, we can write

$$W_9 = a_9 - \Sigma_0(a_8) - f_{MAJ}(a_8, a_7, a_6) - \Sigma_1(e_8) - f_{IF}(e_8, e_7, e_6) - e_5 - K_9.$$

On the right hand side, all quantities other than e_5 have fixed values. Using CDE,

$$e_5 = a_5 + a_1 - \Sigma_0(a_4) - f_{MAJ}(a_4, a_3, a_2).$$

Again in the right hand side, all quantities other than a_1 have fixed values. So, we can write $W_9 = C - a_1$, where C is a fixed value. (This relation has already been observed in [3].)

Now,

$$a_1 = \Sigma_0(a_0) + f_{MAJ}(a_0, b_0, c_0) + \Sigma_1(e_0) + f_{IF}(e_0, f_0, g_0) + h_0 + K_1 + W_1$$

where a_0 and e_0 depend on W_0 whereas b_0, c_0, f_0, g_0 and h_0 depend only on IV and hence are constants. Thus, we can write $a_1 = \Phi(W_0) + W_1$, where

$$\Phi(W_0) = \Sigma_0(a_0) + f_{MAJ}(a_0, b_0, c_0) + \Sigma_1(e_0) + f_{IF}(e_0, f_0, g_0) + h_0 + K_1.$$

We write $\Phi(W_0)$ to emphasize that this depends only on W_0. At this point, we can write

$$\begin{aligned} W_{16} &= \sigma_1(W_{14}) + W_9 + \sigma_0(W_1) + W_0 \\ &= \sigma_1(W_{14}) + C - \Phi(W_0) - W_1 + \sigma_0(W_1) + W_0 \\ &= D - \Phi(W_0) - W_1 + \sigma_0(W_1) + W_0. \end{aligned}$$

Estimate of Computation Effort. Let there be freq_δ values of W_{16} for which $\sigma(W_{16}-1)-\sigma(W_{16})$ equals δ. So, we have to solve this equation for W_0 and W_1 such that W_{16} is one of these freq_δ possible values. The simplest way to do this is to try out random choices of W_0 and W_1 until W_{16} takes one of the desired values. On an average, success is obtained after freq_δ trials. Each trial corresponds to about a single step of SHA-2 computation. So, the total cost of finding suitable W_0 and W_1 is about $\frac{\mathsf{freq}_\delta}{2^{4.5}}$ tries of 23-step SHA-2 computations.

For each such solution (W_0, W_1) and an arbitrary choice of W_{15} we obtain a 23-step collision for SHA-2. Note that after W_0 and W_1 has been obtained everything else is deterministic, i.e., no random tries are required. The task of obtaining a suitable W_0 and W_1 can be viewed as a pre-computation of the type required to find the values of α, γ and μ. Then, the actual task of finding collisions becomes deterministic.

6.2 Relation to the 23-Step Collision from [3]

The NB local collision has been used in [3]. The local collision was placed from Step 9 to Step 17. In comparison, we have shown that the SS local collision gives rise to two kinds of 23-step collision. The first one is obtained by placing the local collision from Steps 8 to 16, and the second one is obtained by placing the local collision from Steps 9 to 17.

The description of the attack in [3] is quite complicated. First they consider a 23-step pseudo-collision which is next converted into 23-step collision. This two-step procedure is unnecessary. Our analysis allows us to directly describe the attacks.

7 24-Step Collisions

The local collision described in Column (I) of Table 2 is placed from Step $i = 10$ to Step $i+8 = 18$ with $u = 1$. The values of δ_1, δ_2 as well as suitable values of α, γ and μ need to be chosen.

Since, the collision ends at Step 18 and $u = 1$, we will have $\delta W_{17} = 1$ and $\delta W_{18} = -1$. As a result, to ensure $\delta W_{19} = \delta W_{20} = 0$, we need to have $\delta_1 = \delta W_{12} = -(\sigma_1(W_{17}+1) - \sigma_1(W_{17}))$ and $\delta_2 = \delta W_{13} = -(\sigma_1(W_{18}-1) - \sigma_1(W_{18}))$. Based on the differential behaviour of σ_1 described in [10], we should try to choose δ_1 and δ_2 such that $\mathsf{freq}_{-\delta_1}$ and freq_{δ_2} are as high as possible. (Here $-\delta_1$ denotes $-\delta_1 \bmod 2^n$, where n is the word size 32 or 64.) But, at the same time, the chosen δ_1 and δ_2 must be such that (3) are satisfied.

Now we consider Table 3. This table tells us what the values of the different a and e-registers need to be. Since messages up to W_{15} are free, we can set values for a and e registers up to Step 15. But, we see that $e_{16} = -1 - u = -2$. This can be achieved by setting W_{16} to

$$W_{16} = e_{16} - \Sigma_1(e_{15}) - f_{IF}(e_{15}, e_{14}, e_{13}) - a_{12} - e_{12} - K_{16}. \tag{4}$$

Since we want $e_{16} = -2$ and all other values on the right hand side are constants, we have that W_{16} is a constant value. On the other hand, W_{16} is defined by message recursion. So, we have to ensure that W_{16} takes the correct value. In addition, we need to ensure that W_{17} and W_{18} take values such that $\sigma_1(W_{17}+1)-\sigma_1(W_{17})=-\delta_1$ and $\sigma_1(W_{18}-1)-\sigma_1(W_{18})=-\delta_2$.

Since $i=10$, from Table 3, we see that a_8 to a_{12} have to be set to fixed values and e_8 to e_{16} have to be set to fixed values. Using CDE, the values of e_{11} down to e_8 are determined by a_7 to a_4. So, the values of a_0 to a_3 are free and correspondingly the choices of words W_0 to W_3 are free.

We have already seen that W_{16} is a fixed value. Note that

$$\left.\begin{aligned} W_{14} &= e_{14} - \Sigma_1(e_{13}) - f_{IF}(e_{13},e_{12},e_{11}) - a_{10} - e_{10} - K_{14} \\ W_{15} &= e_{15} - \Sigma_1(e_{14}) - f_{IF}(e_{14},e_{13},e_{12}) - a_{11} - e_{11} - K_{15}. \end{aligned}\right\} \quad (5)$$

Since for both equations, all the quantities on the right hand side are fixed values, so are W_{14} and W_{15}.

Using CDE twice, we can write

$$\left.\begin{aligned} W_9 &= -W_1 + C_4 + f_{MAJ}(a_4,a_3,a_2) - \Phi_0 \\ W_{10} &= -W_2 + C_5 + f_{MAJ}(a_5,a_4,a_3) - \Phi_1 \\ W_{11} &= -W_3 + C_6 + f_{MAJ}(a_6,a_5,a_4) - \Phi_2 \end{aligned}\right\} \quad (6)$$

where

$$\left.\begin{aligned} C_i &= e_{i+5} - \Sigma_1(e_{i+4}) - f_{IF}(e_{i+4},e_{i+3},e_{i+2}) - 2a_{i+1} - K_{i+5} \\ &\quad +\Sigma_0(a_i), \\ \Phi_i &= \Sigma_0(a_i) + f_{MAJ}(a_i,b_i,c_i) + \Sigma_1(e_i) + f_{IF}(e_i,f_i,g_i) + h_i + \\ &\quad K_{i+1}. \end{aligned}\right\} \quad (7)$$

Using the expressions for W_9, W_{10} and W_{11} we obtain the following expressions for W_{16}, W_{17} and W_{18}.

$$\left.\begin{aligned} W_{16} &= \sigma_1(W_{14}) + C_4 - W_1 + f_{MAJ}(a_4,a_3,a_2) - \Phi_0 + \sigma_0(W_1) \\ &\quad +W_0 \\ W_{17} &= \sigma_1(W_{15}) + C_5 - W_2 + f_{MAJ}(a_5,a_4,a_3) - \Phi_1 + \sigma_0(W_2) \\ &\quad +W_1 \\ W_{18} &= \sigma_1(W_{16}) + C_6 - W_3 + f_{MAJ}(a_6,a_5,a_4) - \Phi_2 + \sigma_0(W_3) \\ &\quad +W_2. \end{aligned}\right\} \quad (8)$$

We need to ensure that W_{16} has the desired value given by (4) and that W_{17} and W_{18} take values which lead to desired values for $\delta\sigma_1(W_{17})$ and $\delta\sigma_1(W_{18})$ as explained above.

The only free quantities are W_0 to W_3 which determine a_0 to a_3. The value of C_4 depends on e_8, e_7 and e_6, where e_8 has a fixed value and e_7 and e_6 are in turn determined using CDE by a_3 and a_2. Similarly, C_5 is determined by e_9, e_8 and e_7; where e_9, e_8 have fixed values and e_7 is determined using a_3. The value of C_6 on the other hand is fixed. Coming to the Φ values, Φ_0 is determined only by W_0; Φ_1 determined by W_0 and W_1; and Φ_2 determined by W_0, W_1 and W_2. Let

$$D = W_{16} - (\sigma_1(W_{14}) + C_4 + f_{MAJ}(a_4,a_3,a_2) - \Phi_0 + W_0). \quad (9)$$

If we fix W_0 and a_3, a_2, then the value of D gets fixed and we need to find W_1 such that the following equation holds.

$$D = -W_1 + \sigma_0(W_1). \tag{10}$$

A guess-then-determine algorithm can be used to solve this equation. This algorithm will be different for SHA-256 and for SHA-512 since the σ_0 function is different for the two. The guess-then-determine algorithms for both SHA-256 and SHA-512 are described in [10].

Solving (10) Using Table Look-Up. An alternative approach would be to use a pre-computed table. For each of the 2^n possible W_1s (n is the word size 32 or 64), prepare a table of entries $(W_1, -W_1 + \sigma_0(W_1))$ sorted on the second column. Then all solutions (if there are any) for (10) can be found by a simple look-up into the table using D. The table would have 2^n entries and if a proper index structure is used, then the look-up can be done very fast. We have not implemented this method.

Given $a_1, b_1, \ldots, h_1$ and a_2 the value of W_2 gets uniquely defined; similarly, given $a_2, b_2, \ldots, h_2$ and a_3, the value of W_3 gets uniquely defined. The equations are the following.

$$\left.\begin{aligned} W_2 &= a_2 - (\Sigma_0(a_1) + f_{MAJ}(a_1, b_1, c_1) + h_1 + \Sigma_1(e_1) \\ &\quad + f_{IF}(e_1, f_1, g_1) + K_2) \\ W_3 &= a_3 - (\Sigma_0(a_2) + f_{MAJ}(a_2, b_2, c_2) + h_2 + \Sigma_1(e_2) \\ &\quad + f_{IF}(e_2, f_2, g_2) + K_3) \end{aligned}\right\} \tag{11}$$

The strategy for determining suitable $W_0, \ldots, W_3$ is the following.

1. Make random choices for W_0 and a_2, a_3.
2. Run SHA-2 with W_0 and determine Φ_0.
3. From a_3 and a_2 determine e_7 and e_6 using CDE.
4. Determine C_4 using (7) and then D using (9).
5. Solve (10) for W_1 using the guess-then-determine algorithm.
6. Run SHA-2 with W_1 to define $a_1, \ldots, h_1$.
7. Determine Φ_1 using (7) and then W_2 using (11).
8. Run SHA-2 with W_2 to define $a_2, \ldots, h_2$.
9. Determine Φ_2 using (7) and then W_3 using (11).
10. Compute W_{17} and W_{18} using (8).
11. If $\sigma_1(W_{17} + 1) - \sigma_1(W_{17}) = -\delta_1$ and $\sigma_1(W_{18} - 1) - \sigma_1(W_{18}) = \delta_2$, then return W_0, W_1, W_2 and W_3.

The values of W_0, W_1, W_2 and W_3 returned by this procedure ensure that the local collision ends properly at Step 18 and that $\delta W_j = 0$ for $j = 19, \ldots, 23$. This provides a 24-step collision.

Estimate of Computation Effort. Let Step 5 involve a computation of g operations, where each operation is much faster than a single step of SHA-2; by our assessment the time for each operation is around 2^{-4} times the cost of a single step of SHA-2. Thus, the time for Step 5 is about $\frac{g}{2^4}$ single SHA-2 steps. Further, let the success probability of the guess-then-determine attack be p. Then Step 5 needs to be repeated roughly $\frac{1}{p}$ times to obtain a solution.

By the choice of δ_1, the equality $\sigma_1(W_{17}+1) - \sigma_1(W_{17}) = -\delta_1$ holds roughly with probability $\frac{\mathsf{freq}_{\delta_1}}{2^n}$ while by the choice of δ_2 the equality $\sigma_1(W_{18}-1) - \sigma_1(W_{18}) = \delta_2$ holds roughly with probability $\frac{\mathsf{freq}_{\delta_2}}{2^n}$ and we obtain success in Step 11 with roughly $\frac{\mathsf{freq}_{\delta_1} \times \mathsf{freq}_{\delta_2}}{2^{2n}}$ probability. So, the entire procedure needs to be carried out around $\frac{2^{2n}}{\mathsf{freq}_{\delta_1} \times \mathsf{freq}_{\delta_2}}$ times to obtain a collision.

The guess-then-determine step takes about $g/2^4$ single SHA-2 steps. The time for executing the entire procedure once is about $(\frac{g}{2^4}+3)$ single SHA-2 steps which is about $2^{-4.5} \times (\frac{g}{2^4}+3)$ 24-step SHA-2 computations. Since the entire process needs to be repeated many times for obtaining success, the number of 24-step SHA-2 computations till success is obtained is about $(\frac{2^{2n}}{\mathsf{freq}_{\delta_1} \times \mathsf{freq}_{\delta_2}}) \times (2^{-4.5} \times (\frac{g}{2^4}+3) \times \frac{1}{p})$.

If (10) is solved using a table look-up, then the cost estimate changes quite a lot. The cost of Step 5 reduces to about a single SHA-2 step so that the overall cost reduces to about $(\frac{2^{2n}}{\mathsf{freq}_{\delta_1} \times \mathsf{freq}_{\delta_2}}) \times (2^{-4.5} \times 3 \times \frac{1}{p})$ 24-step SHA-2 computations. The trade-off is that we need to use a look-up table having 2^n entries.

8 Exhibiting Colliding Message Pairs

The description in the previous sections provide an outline of how to obtain colliding message pairs. To actually find collisions, a lot more details are required. Due to lack of space, we are unable to provide these details here. (The reader may refer to [10] for further details.) Here we simply provide examples of actual collisions that we have found. These are given in Tables 4 to 10.

Table 4. Colliding message pair for 22-step SHA-512 with standard IV

W_1	0-3	0000000000000000	0000000000000000	c2bc8e9a85e2eb5a	6d623c5d5a2a1442
	4-7	cd38e6dee1458de7	acb73305cddb1207	148f31a512bbade5	ecd66ba86d4ab7e9
	8-11	92aafb1e9cfa1fcb	533c19b80a7c8968	e3ce7a41b11b4d75	aef3823c2a004b20
	12-15	8d41a28b0d847692	7f214e01c4e96950	0000000000000000	0000000000000000
W_2	0-3	0000000000000000	0000000000000000	c2bc8e9a85e2eb5a	6d623c5d5a2a1442
	4-7	cd38e6dee1458de7	acb73305cddb1207	148f31a512bbade5	ecd66ba86d4ab7ea
	8-11	90668fd7ec6718ee	533c19b80a7c8968	dfce7a41b11b4d76	aef3823c2a004b20
	12-15	8d41a28b0d847692	7f214e01c4e96950	0000000000000000	ffffffffffffffff

Table 5. Colliding message pair for 22-step SHA-256 with standard IV

W_1	0-7	00000000	00000000	0be293bf	99c539c9	1c672194	99b6a58a	5bf1d0ae	0a9a18d3
	8-15	0c18cf1c	329b3e6e	dc4e7a43	ab33823f	8d41a28d	7f214e03	00000000	00000000
W_2	0-7	00000000	00000000	0be293bf	99c539c9	1c672194	99b6a58a	5bf1d0ae	0a9a18d4
	8-15	07d56809	329b3e6e	dc0e7a44	ab33823f	8d41a28d	7f214e03	00000000	ffffffff

Table 6. Colliding message pair for 23-step SHA-256 with standard IV. These messages utilize a single local collision starting at Step $i = 8$.

W_1	0-7	122060e3	000f813f	d92d3fc6	ea4a475f	fb0c6581	dc4558c4	d86428b4	6e2ca576
	8-15	c8d597bf	6372d4c2	ddbd721c	79d654c4	f0064002	a894b7b6	91b7628e	3224db20
W_2	0-7	122060e3	000f813f	d92d3fc6	ea4a475f	fb0c6581	dc4558c4	d86428b4	6e2ca576
	8-15	c8d597c0	6372d4c1	ddbd721c	78d6b4c5	f0064002	a894b7b6	91b7628e	3224db20

Table 7. Colliding message pair for 23-step SHA-256 with standard IV. These messages utilize a single local collision starting at Step $i = 9$.

W_1	0-7	c201bef2	14cc32c9	3b80da44	d8212037	8987161d	a790cb4a	53b8d726	89e9a288
	8-15	3edd76e0	05f41ddc	9ebc0fc3	e099698a	2eaec58f	e7060b78	95d7030d	6bf777c0
W_2	0-7	c201bef2	14cc32c9	3b80da44	d8212037	8987161d	a790cb4a	53b8d726	89e9a288
	8-15	3edd76e0	05f41ddd	9ebc0fc2	e099c98a	2daf2590	e7060b78	95d7030d	6bf777c0

Table 8. Colliding message pair for 24-step SHA-256 with standard IV. These messages utilize a single local collision starting at Step $i = 10$.

W_1	0-7	657adf63	06c066d7	90f0b709	95a3e1d1	c3017f24	fad6c2bf	dff43685	6abff0da
	8-15	e6cfc63f	de8fb4c1	c20ca05b	f74815cc	c2e789d9	208e7105	cc08b6cf	70171840
W_2	0-7	657adf63	06c066d7	90f0b709	95a3e1d1	c3017f24	fad6c2bf	dff43685	6abff0da
	8-15	e6cfc63f	de8fb4c1	c20ca05c	f74815cb	c2e7e9d9	1f8ed106	cc08b6cf	70171840

Table 9. Colliding message pair for 23-step SHA-512 with standard IV. These messages utilize a single local collision starting at Step $i = 8$.

W_1	0-3	b9fa6fc4729ca55c	8718310e1b3590e1	1d3d530cb075b721	99166b30ecbdd705
	4-7	27ed55b66c090b62	754b2163ff6feec5	6685f40fd8ab08f8	590c1c0522f6fdfd
	8-11	b947bb4013b688c1	d9d72ca8ab1cac04	69d0e120220d4edc	30a2e93aeef24e3f
	12-15	84e76299718478b9	f11ae711647763e5	d621d2687946e862	0ee57069123ecc8b
W_2	0-3	b9fa6fc4729ca55c	8718310e1b3590e1	1d3d530cb075b721	99166b30ecbdd705
	4-7	27ed55b66c090b62	754b2163ff6feec5	6685f40fd8ab08f8	590c1c0522f6fdfd
	8-11	b947bb4013b688c2	d9d72ca8ab1cac03	69d0e120220d4edc	30a3493aeef25076
	12-15	84e76299718478b9	f11ae711647763e5	d621d2687946e862	0ee57069123ecc8b

Table 10. Colliding message pair for 24-step SHA-512 with standard IV. These messages utilize a single local collision starting at Step $i = 10$.

W_1	0-3	dedb689cfc766965	c7b8e064ff720f7c	c136883560348c9c	3747df7d0cf47678
	4-7	855e17555cfedc5f	88566babccaa63e9	5dda9777938b73cd	b17b00574a4e4216
	8-11	86f3ff48fd12ea19	cd15c6f8d6da38ce	5e2c6b7b0411e70b	36ed67e93a794e66
	12-15	1b65e96b02767821	04d0950089db6c68	5bc9b9673e38eff3	b05d879ad024d3fa
W_2	0-3	dedb689cfc766965	c7b8e064ff720f7c	c136883560348c9c	3747df7d0cf47678
	4-7	855e17555cfedc5f	88566babccaa63e9	5dda9777938b73cd	b17b00574a4e4216
	8-11	86f3ff48fd12ea19	cd15c6f8d6da38ce	5e2c6b7b0411e70c	36ed67e93a794e65
	12-15	1b66096b02767829	04d0f50089db6e9f	5bc9b9673e38eff3	b05d879ad024d3fa

Note

The submitted version of the paper contained much more details than is provided in the current version. Due to page-limit restrictions on the published version of the paper, we are unable to provide such details, which to a certain extent may affect the readability of the paper. A longer and more detailed version is available at [10].

References

1. Secure Hash Standard. Federal Information Processing Standard Publication 180-2. U.S. Department of Commerce, National Institute of Standards and Technology (NIST) (2002), http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf
2. Gilbert, H., Handschuh, H.: Security Analysis of SHA-256 and Sisters. In: Matsui, M., Zuccherato, R.J. (eds.) SAC 2003. LNCS, vol. 3006, pp. 175–193. Springer, Heidelberg (2004)
3. Indesteege, S., Mendel, F., Preneel, B., Rechberger, C.: Collisions and other Non-Random Properties for Step-Reduced SHA-256. Cryptology eprint Archive (April 2008); Selected Areas in Cryptography (accepted, 2008), http://eprint.iacr.org/2008/131
4. Mendel, F., Pramstaller, N., Rechberger, C., Rijmen, V.: Analysis of Step-Reduced SHA-256. In: Robshaw, M.J.B. (ed.) FSE 2006. LNCS, vol. 4047, pp. 126–143. Springer, Heidelberg (2006)
5. Mendel, F., Pramstaller, N., Rechberger, C., Rijmen, V.: Analysis of Step-Reduced SHA-256. Cryptology eprint Archive (March 2008), http://eprint.iacr.org/2008/130
6. Nikolić, I., Biryukov, A.: Collisions for Step-Reduced SHA-256. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 1–16. Springer, Heidelberg (2008)
7. Sanadhya, S.K., Sarkar, P.: New Local Collisions for the SHA-2 Hash Family. In: Nam, K.-H., Rhee, G. (eds.) ICISC 2007. LNCS, vol. 4817, pp. 193–205. Springer, Heidelberg (2007)
8. Sanadhya, S.K., Sarkar, P.: Attacking Reduced Round SHA-256. In: Bellovin, S., Gennaro, R. (eds.) ACNS 2008. LNCS, vol. 5037, pp. 130–143. Springer, Heidelberg (2008)
9. Sanadhya, S.K., Sarkar, P.: Deterministic Constructions of 21-Step Collisions for the SHA-2 Hash Family. In: Wu, T.-C., Lei, C.-L., Rijmen, V., Lee, D.-T. (eds.) ISC 2008. LNCS, vol. 5222. Springer, Heidelberg (2008)
10. Sanadhya, S.K., Sarkar, P.: New Collision attacks Against Up To 24-step SHA-2. Cryptology eprint Archive (September 2008), http://eprint.iacr.org/2008/270
11. Sanadhya, S.K., Sarkar, P.: Non-Linear Reduced Round Attacks Against SHA-2 Hash family. In: Mu, Y., Susilo, W. (eds.) ACISP 2008. LNCS, vol. 5107, pp. 254–266. Springer, Heidelberg (2008)