A Combinatorial Analysis of Recent Attacks on Step Reduced SHA-2 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. We perform a combinatorial analysis of SHA-2 compression function. This analysis explains in a unified way the recent attacks against reduced round SHA-2. We start with a general class of local collisions and show that the previously used local collision by Nikolić and Biryukov (NB) and Sanadhya and Sarkar (SS) are special cases. The study also clarifies several advantages of the SS local collision over the NB local collision. Deterministic constructions of up to 22-round SHA-2 collisions are described using the SS local collision and up to 21-round SHA-2 collisions are described using the NB local collision. For 23 and 24-round SHA-2, we describe a general strategy and then apply the SS local collision to this strategy. The resulting attacks are faster than those proposed by Indesteege et al using the NB local collision. We provide colliding message pairs for 22, 23 and 24-round SHA-2. Although these attacks improve upon the existing reduced round SHA-256 attacks, they do not threaten the security of the full SHA-2 family.[1]

Keywords: SHA-2 family, reduced round collisions, cryptanalysis.

1 Introduction

Collision resistant hash functions (CRHF) are of great practical importance in cryptography. Consequently, over the years, a lot of effort has been expended in the design and analysis of such functions. The most famous families of CRHFs are the SHA-families standardized by NIST [24] of USA and are based on iterative Merkle-Damgård (MD) [14, 5] type of hash functions designed by Rivest.

A CRHF maps arbitrarily long strings to a short fixed length string. Consequently, collisions are bound to exist. Cryptanalysis of a CRHF consists of finding one such collision for the given CRHF. Since the description of function is given, one needs to carefully analyse the structure of the function in order to determine a collision. This necessitates a detailed combinatorial study of the function. One approach is to linearize the function by replacing all non-linear components with their best linear approximation. Finding a collision for such a linearized function is easy, but, the collision holds for the original function only probabilistically. One then has to look for methods to increase the probability. Alternatively, one could work directly with the nonlinear function itself. This makes the analysis more difficult, but the probability of a collision is much higher.

Cryptanalysis of the MD-family and the SHA-family has been extensively studied with major successes coming at infrequent intervals. The first major success was the cryptanalysis of MD4 by Dobbertin [6, 7] which led to actually exhibiting a colliding message pair. This was followed by partial attacks on MD5 with full cryptanalysis of MD5 and other hash functions coming recently [28, 26]. The NIST standard SHA-1 family was theoretically cryptanalysed in [27] (though, till date, a colliding message pair for SHA-1 remains to be found). Earlier, partial cryptanalysis of SHA-0 was done in [3, 1]. Following

[1] This work builds upon and subsumes previous work [22, 20, 16, 21] done by us. Whereas the previous works focussed on obtaining collisions for fixed number of rounds, the current work provides the combinatorial framework for understanding how such collisions arise.

the works in [28, 27], there have been attacks [11, 25] on MD5 with improved time complexities and/or providing collisions of structured messages.

The SHA-2 family consists of two main hash functions, SHA-256 and SHA-512, and their truncated versions SHA-224 and SHA-384. In view of the existing attacks, the only surviving family in the NIST standard is the SHA-2 family. Consequently, it is of interest to analyse the SHA-2 family. Cryptanalysis of SHA-2 family has recently gained momentum due to the important work of Nikolić and Biryukov [15]. Prior work on finding collisions for step reduced SHA-256 was done in [12, 13] and [19]. These earlier works used local collisions valid for the linearized version of SHA-256 from [8] and [17]. On the other hand, the work [15] used a local collision which is valid for the actual SHA-256.

The authors in [15] developed techniques to handle nonlinear functions and the message expansion of SHA-2 to obtain collisions for up to 21-round SHA-256. The 21-round attack of [15] succeeded with probability 2^{-19}. Very recently, Indesteege et al [9] have developed attacks against 23 and 24-round SHA-2 family. They utilize the local collision from [15] in these attacks. Following the work of [15] and partly in parallel to [9], we have published several papers [22, 20, 16, 21] on finding SHA-2 collisions for up to 24 steps with time complexities better than those obtained in [9]. The current work subsumes our previous works and provides a unified combinatorial analysis of the attacks. More details are given below.

Our contributions. We take a general approach to the analysis of SHA-2 family. The set of all possible 9-round local collisions using additive differentials are analysed using a general and unified framework. Simplification of the expressions are done in a systematic manner which lead us to the local collisions from [15] and [22] as special cases. We will call these NB and SS local collisions respectively.

We show that it is possible to deterministically construct up to 22-round SHA-2 collisions using the SS local collision and up to 21-round SHA-2 collisions using the NB local collision. A general method for obtaining collisions for 23 and 24-round SHA-2 are described. This method can be applied with both the NB and the SS local collisions. From the analysis it becomes clear that the SS local collision offers certain advantages over the NB local collision. Hence, we focus on the SS local collision which leads to 23 and 24-round collisions with better time complexities than those obtained using the NB local collision. A summary of results on collision attacks against reduced SHA-2 family is given in Table 1. Examples of 22, 23 and 24-round SHA-256 and SHA-512 collisions are presented in Appendix A.

We highlight the case of 23 and 24-round SHA-512 attacks from Table 1. These are considerably improved in comparison to the existing attack of [9]. While [9] describes these attacks with reported complexities of $2^{44.9}$ and 2^{53} calls to the corresponding functions, our attacks have complexities $2^{16.5}$ and $2^{32.5}$ calls. In fact, the improvement in the time complexity of the 24-round SHA-512 attack allows us to provide the *first* message pair which collides for 24-round SHA-512.

Chronology of recent attacks on SHA-2. Nikolić and Biryukov [15] started the analysis of SHA-2 using nonlinear differentials and attacked up to 21-round SHA-256. Our work was motivated by theirs. We generalize their technique and use a different local collision with certain advantages over the NB local collision. Also, we extend the number of rounds that can be attacked to 24.

The work [9] and its different versions [10] (later published as [9]) was done independently and in parallel to ours [22, 20, 16, 21]. This work used the NB local collision. The chronological sequence of our work and that of the different versions of [10] for obtaining 22 to 24-round SHA-2 collisions is the following.

1. Our work [16, 08-Mar-2008] provided the first example of colliding message pairs for 22-round SHA-2.
2. The version [10, 08-Apr-2008] provided the first examples of colliding message pairs for 23 and 24-round SHA-256.

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	Attack Type	Example
			Prob.	Calls	utilized		provided
[12, 13]	SHA-256	18		*	GH [8]	Linear	yes
[19]	SHA-256	18		**	SS_5 [17]	”	yes
[15]	SHA-256	20	$\frac{1}{3}$		NB [15]	Non-linear	yes
		21	2^{-19}		”	”	yes
[22]	SHA-256/SHA-512	18,20	1	1	SS [22]	”	yes
	SHA-256	21	2^{-15}		”	”	yes
[20]	SHA-256/SHA-512	21	1	1	”	”	yes
[9]	SHA-256	23		2^{18}	NB [15]	”	yes
		24		$2^{28.5}$	”	”	yes
	SHA-512	23		$2^{44.9}$	”	”	yes
		24		2^{53}	”	”	**no**
This work	SHA-256/SHA-512	22	1	1	SS [22]	”	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 [12, 13] 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.

3. An earlier version of the present work [18, 12-Jun-2008] provided examples of colliding message pairs for 23 and 24-round SHA-256 with improved time complexities.
4. The version [10, 14-Jul-2008] provided the first examples of colliding message pairs for 23-round SHA-512 and a theoretical attack on 24-round SHA-512 with reported time complexity of 2^{53} calls to the compression function.
5. Our paper [21] provides example of a colliding message pair for 23-round SHA-512 with improved time complexity; and the *first* example of a colliding message pair for 24-round SHA-512 (also with improved time complexity).

As mentioned earlier, the current work subsumes our earlier works [22, 20, 16, 21] providing a unified view of the attacks. More generally, the framework of our combinatorial analysis explains how things fit together.

Another important difference between [10, 9] and ours is that we provide complete details of each and every result and algorithm used in the attack. In contrast, the description of the attacks in [10, 9] is rather incomplete and the time complexities are not properly explained. They describe the attacks as two-phase procedures, where in the first phase a pseudo-collision is obtained which is then converted to a collision. Our analysis shows that such a two-phase description is unnecessary.

We believe that the approach we take leads to a better understanding of the combinatorial structure of the SHA-2 family.

Note. In a recent work [23], we have suggested modifications to the SHA-2 design to overcome the attacks described in this work and in the works [15, 9]. The design given in [23] also introduces the idea of multiple feedforward in the context of hash function design.

2 Preliminaries

We will 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: $\{W_0, W_1, W_2, \ldots W_{15}\}$ and $\{W'_0, W'_1, W'_2, \ldots W'_{15}\}$.
- Expanded message pair: $\{W_0, W_1, W_2, \ldots W_{N-1}\}$ and $\{W'_0, W'_1, W'_2, \ldots W'_{N-1}\}$.
  The number of steps N 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.

2.1 SHA-2 Compression Function

The complete description of the SHA-2 hash family can be found from [24]. In this work, we will need only the compression function. A description is given below.

The input to the compression function consists of 8 n-bit registers and a message block which consists of 16 n-bit words. The output consists of 8 n-bit words. For the first message block, the values of the input registers are given by 8 fixed n-bit words called the initialization vector and for later message blocks, these values are the output of the previous invocation of the compression function.

The message block is expanded from 16 n-bit words $W_0, \ldots, W_{15}$ to N n-bit words $W_0, \ldots, W_{N-1}$. A round function is applied N times. Each application updates the values of the registers. In Step i with $0 \leq i \leq N-1$, 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) as follows. (($a_{-1}, \ldots, h_{-1}$) corresponds to the initial value of the registers.)

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

The functions 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:

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

Given the message words $W_0, W_1, \ldots, W_{15}$; for $i \geq 16$, W_i is computed as follows.

$$W_i = \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16} \qquad (2)$$

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

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

And for SHA-512, they are defined as:

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

The final output of the compression function is $(a_{-1} + a_{N-1}, \ldots, h_{-1} + h_{N-1})$. Adding the initial values $(a_{-1}, \ldots, h_{-1})$ to the output of the final application of the round function is called feed-forward.

Reduced Round SHA-2. The value of N is fixed by the specification [24]. For the purpose of analysis, one may work with a lower value of N. In this paper, we will work with N up to 24. Everything else of the compression function, including the feed-forward, remain the same. Actually, we will not have to bother about the feed-forward, since we will be obtaining collisions for several steps of the round function itself.

2.2 Cross Dependence Equation (CDE)

By the form of the round update function in (1), we have the following relation.

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

Later, we make extensive use of this relation. A special case of this equation was utilized in Section 6.1 of [22]. The equation in the form above was used in [20]. 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 independently observed in [9].

The following result can be used to set registers to specific values.

Proposition 1. *Suppose that $(a_{i-1}, \ldots, h_{i-1})$ are known and α and β are any two n-bit words. Then it is possible to choose W_i such that either $a_i = \alpha$ or $e_i = \beta$. In general, however, using only W_i, it is not possible to simultaneously set both a_i to α and e_i to β.*

Proof. This is an easy consequence of (1). Consider the equation for a_i. This is given in terms of $(a_{i-1}, \ldots, h_{i-1})$ and W_i. So, if we set

$$W_i = \alpha - (\Sigma_0(a_{i-1}) + f_{MAJ}(a_{i-1}, b_{i-1}, c_{i-1}) + \Sigma_1(e_{i-1}) + f_{IF}(e_{i-1}, f_{i-1}, g_{i-1}) + h_{i-1} + K_i),$$

then clearly $a_i = \alpha$ is attained. Similarly for e_i.

Note, however, that using W_i, we cannot simultaneously set the values of both a_i and e_i. □

Even though we cannot use Proposition 1 to simultaneously set the values of a_i and e_i, there is a way out. This way is given by the CDE. Suppose, the values of $a_{i-3}, \ldots, a_i$ have already become fixed, but, a_{i-4} is still free. Then by choosing a suitable value for a_{i-4} we can attain *any* desired value for e_i. Now, using Proposition 1, we can use W_{i-4} to set a_{i-4} to the required value. So, in effect, we can use W_{i-4} to set e_i to *any* desired value. This is something nice (from a cryptanalytic point of view) and unexpected and we use this feature extensively.

2.3 Differential Properties of σ_1

For the analysis of 23 and 24-round SHA-2, we will need to consider the differential properties of σ_1 with respect to modular addition. The particular property that we require is discussed in this section.

SHA-256. Consider the distribution of $\delta = \sigma_1(W) - \sigma_1(W - 1)$ as W ranges over all 2^{32} values. This distribution is highly skewed and was mentioned in Section 7.1 in [22]. Later, it has been independently observed in [9] that δ takes only 6181 values and there are several values of δ which occur for more than 2^{29} or more values of W.

Let freq_δ be the number of W such that $\delta = \sigma_1(W) - \sigma_1(W - 1)$. It is quite easy to prepare a list of $(\delta, \mathsf{freq}_\delta)$ values. For each of the 2^{32} values of W, compute $\delta = \sigma_1(W) - \sigma_1(W - 1)$. If this δ has been obtained earlier, then increment the frequency for this δ; else insert $(\delta, \mathsf{freq}_\delta = 1)$ into the list. To do this efficiently, we need a suitable index structure for searching and inserting into the list. A height balanced tree (or AVL tree) is the optimal solution; but, for the current application, a simple (data structure) hash technique is good enough and is the technique we implemented. Some values of $(\delta, \mathsf{freq}_\delta)$ are given in Table 2.

Note. Interestingly, we have observed that if freq_δ is greater than 2^{16}, then δ is always even.

Table 2. Some examples of high frequency values of $\delta = \sigma_1(W) - \sigma_1(W-1)$ for SHA-256.

δ	freq_δ	δ	freq_δ
`ffff6000`	$2^{29}+2^{26}+2^{25}$	0000a000	$2^{29}+2^{26}+2^{25}$
`ffffa000`	$2^{29}+2^{26}$	00006000	$2^{29}+2^{26}$
`ff006001`	2^{16}	`ff005fff`	2^{16}

SHA-512. In this case, $n = 64$ and it is not possible to exhaustively prepare a list of values for $\delta = \sigma_1(W) - \sigma_1(W-1)$ for all possible 2^{64} values of W. Instead, we created a list using 2^{25} randomly chosen values of W. This provides certain values of δ with certain frequencies. From these frequencies we extrapolate to estimate the actual frequency of each delta among all the 2^{64} choices of W. The extrapolation is done in the following manner. If a particular difference δ occurs κ times in 2^{25} random trials, then we expect it to have a frequency freq_δ of about $\kappa \times 2^{64}/2^{25}$. Some of the observed and the extrapolated frequencies are shown in Table 3.

Table 3. Some examples of high frequency values of $\delta = \sigma_1(W) - \sigma_1(W-1)$ for SHA-512. The column freq_O denotes the observed frequencies among 2^{25} random trials of computing δ. The column freq_δ contains the extrapolated values of the frequencies for the complete search space of 2^{64}.

δ	freq_O	freq_δ	δ	freq_O	freq_δ
200000000008	4795491	$2^{61.5}$	8e000000003a9	22	$2^{43.5}$
`ffffdffffffffff8`	4793201	$2^{61.5}$	`fff26000000000c9`	22	$2^{43.5}$
`1ffffffffff8`	4792982	$2^{61.5}$	600000000237	18	$2^{43.5}$

3 A General Non-Linear Differential Path

We use a differential technique to find a 9-round local collision. The idea is to use modular differentials which was first used for SHA-2 by Nikolić and Biryukov [15]. Given a word w, we define

$$x = -\delta\Sigma_0^i(w) - \delta f_{MAJ}^i(w,0,0); \quad y = -\delta f_{MAJ}^{i+1}(0,w,0); \quad z = -\delta f_{MAJ}^{i+2}(0,0,w). \tag{4}$$

For t-bit words α, β, γ and integer i, we use the following short-hands.

$$\left.\begin{array}{r}
\delta\Sigma_1^i(\alpha) = \Sigma_1(e_i+\alpha) - \Sigma_1(e_i) = \Sigma_1(e_i') - \Sigma_1(e_i).\\
\delta\Sigma_0^i(\alpha) = \Sigma_0(a_i+\alpha) - \Sigma_0(a_i) = \Sigma_0(a_i') - \Sigma_0(a_i).\\
\delta f_{IF}^i(\alpha,\beta,\gamma) = f_{IF}(e_i+\alpha, f_i+\beta, g_i+\gamma) - f_{IF}(e_i,f_i,g_i) = f_{IF}(e_i',f_i',g_i') - f_{IF}(e_i,f_i,g_i).\\
\delta f_{MAJ}^i(\alpha,\beta,\gamma) = f_{MAJ}(a_i+\alpha, b_i+\beta, c_i+\gamma) - f_{IF}(a_i,b_i,c_i) = f_{MAJ}(a_i',b_i',c_i') - f_{IF}(a_i,b_i,c_i).\\
\delta\sigma_0(\delta W_i) = \sigma_0(W_i+\delta W_i) - \sigma_0(W_i) = \sigma_0(W_i') - \sigma_0(W_i).\\
\delta\sigma_1(\delta W_i) = \sigma_1(W_i+\delta W_i) - \sigma_1(W_i) = \sigma_1(W_i') - \sigma_1(W_i).
\end{array}\right\} \tag{5}$$

The general differential path and corresponding message differences are shown in Table 4. It can be verified that the differential path holds for the stated message differences. We show the first step of the computation, the other steps are similar. In the $(i+1)$st step we want $\delta a_{i+1} = 0$ and $\delta e_{i+1} = x$. The given values of x and δW_{i+1} ensure that these two conditions hold. Note that the values of the other registers are fixed by the values of the registers at the ith step.

$$\begin{aligned}
\delta a_{i+1} &= a'_{i+1} - a_{i+1} \\
&= (\Sigma_0(a'_i) + f_{MAJ}(a'_i, b'_i, c'_i) + \Sigma_1(e'_i) + f_{IF}(e'_i, f'_i, g'_i) + h'_i + K'_{i+1} + W'_{i+1}) \\
&\quad -(\Sigma_0(a_i) + f_{MAJ}(a_i, b_i, c_i) + \Sigma_1(e_i) + f_{IF}(e_i, f_i, g_i) + h_i + K_{i+1} + W_{i+1}) \\
&= (\Sigma_0(a'_i) - \Sigma_0(a_i)) + (f_{MAJ}(a'_i, b'_i, c'_i) - f_{MAJ}(a_i, b_i, c_i)) \\
&\quad +(\Sigma_1(e'_i) - \Sigma_1(e_i)) + (f_{IF}(e'_i, f'_i, g'_i) - f_{IF}(e_i, f_i, g_i)) + (W'_{i+1} - W_{i+1}) \\
&= \delta\Sigma_0^i(w) + \delta f_{MAJ}^i(w, 0, 0) + \delta\Sigma_1^i(w) + \delta f_{IF}^i(w, 0, 0) + \delta W_{i+1} \\
&= -x + (\delta\Sigma_1^i(w) + \delta f_{IF}^i(w, 0, 0)) + (x - \delta\Sigma_1^i(w) - \delta f_{IF}^i(w, 0, 0)) \\
&= 0 \\
\delta e_{i+1} &= e'_{i+1} - e_{i+1} \\
&= (\Sigma_1(e'_i) + f_{IF}(e'_i, f'_i, g'_i) + h'_i + K'_{i+1} + W'_{i+1}) \\
&\quad -(\Sigma_1(e_i) + f_{IF}(e_i, f_i, g_i) + h_i + K_{i+1} + W_{i+1}) \\
&= (\Sigma_1(e'_i) - \Sigma_1(e_i)) + (f_{IF}(e'_i, f'_i, g'_i) - f_{IF}(e_i, f_i, g_i)) + (W'_{i+1} - W_{i+1}) \\
&= \delta\Sigma_1^i(w) + \delta f_{IF}^i(w, 0, 0) + \delta W_{i+1} \\
&= \delta\Sigma_1^i(w) + \delta f_{IF}^i(w, 0, 0) + x - \delta\Sigma_1^i(w) - \delta f_{IF}^i(w, 0, 0) \\
&= x.
\end{aligned}$$

Table 4. General 9-round nonlinear local collision for SHA-256.

Differential Path										Message Word Differences
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	$\delta W_i = w;$ $\delta W_{i+1} = x - \delta\Sigma_1^i(w) - \delta f_{IF}^i(w, 0, 0);$ $\delta W_{i+2} = y - \delta\Sigma_1^{i+1}(x) - \delta f_{IF}^{i+1}(x, w, 0);$ $\delta W_{i+3} = z - \delta\Sigma_1^{i+2}(y) - \delta f_{IF}^{i+2}(y, x, w);$ $\delta W_{i+4} = -w - \delta\Sigma_1^{i+3}(z) - \delta f_{IF}^{i+3}(z, y, x);$ $\delta W_{i+5} = -x - \delta\Sigma_1^{i+4}(w) - \delta f_{IF}^{i+4}(w, z, y);$ $\delta W_{i+6} = -y - \delta f_{IF}^{i+5}(0, w, z);$ $\delta W_{i+7} = -z - \delta f_{IF}^{i+6}(0, 0, w);$ $\delta W_{i+8} = -w.$
i	w	w	0	0	0	w	0	0	0	
$i+1$	δW_{i+1}	0	w	0	0	x	w	0	0	
$i+2$	δW_{i+2}	0	0	w	0	y	x	w	0	
$i+3$	δW_{i+3}	0	0	0	w	z	y	x	w	
$i+4$	δW_{i+4}	0	0	0	0	w	z	y	x	
$i+5$	δW_{i+5}	0	0	0	0	0	w	z	y	
$i+6$	δW_{i+6}	0	0	0	0	0	0	w	z	
$i+7$	δW_{i+7}	0	0	0	0	0	0	0	w	
$i+8$	δW_{i+8}	0	0	0	0	0	0	0	0	

The important thing to note about the differential path shown in Table 4 is that it puts no restrictions on the actual message words $W_i, \ldots, W_{i+8}$. Starting at any value for the registers a_i to h_i, and using any given non-zero w, and any $W_i, \ldots, W_{i+8}$, we simply run the compression function step-by-step and define the words x, y, z, the respective δW_is and consequently the respective W'_is. All the steps are deterministic and hence with probability one, we obtain W'_is which collide with W_is. This gives rise to a local collision.

Note. We have defined $\delta X = X' - X$ and so $\delta W_i = w$ means $W'_i = W_i + w$; if we had defined δX to be $X - X'$, then W'_i would have been $W_i - w$. Consequently, without loss of generality one can assume $w > 0$.

Specifying the values of (w, x, y, z) completely specifies message differences as well as the differences in the register values at all the steps. Two special cases for (w, x, y, z) have been used.

Nikolić-Biryukov (NB) [15]. $(w, x, y, z) = (1, -1, 0, 0)$.
Sanadhya-Sarkar (SS) [22]. $(w, x, y, z) = (1, -1, -1, 0)$.

The NB local collision was the first to be proposed and has been used for finding collisions in both [15] and [10, 9]. The SS local collision was proposed later and was motivated by the analysis done in [15]. But, it turns out that the SS local collision is actually more attractive than the NB local collision. This is due to the fact that the time complexities of collision attacks using the SS local collision is lesser than the time complexities of collision attacks using the NB local collision. To understand why this is so, one needs to go through the detailed combinatorial analysis of the SHA-2 round function carried out in this work.

3.1 Simplifications

The differential path by itself is not useful for obtaining longer round collisions. To do this, we need to simplify the expressions and obtain conditions. This is done using several rules which are actually sufficient conditions. The rules and their consequences are described below.

Simplifying $\delta\Sigma_0$. There is only one occurrence of Σ_0 in all the expressions and that is in the expression for x. In both SHA-256 and SHA-512, Σ_0 is a linear function which is invariant only on 0 and -1. Note that $-1 =$ `ffffffff` for SHA-256 and $-1 =$ `ffffffffffffffff` for SHA-512. Since $\delta\Sigma_0^i(w) = \Sigma_0(a_i + w) - \Sigma_0(a_i)$ an easy way to satisfy this is to ensure that both a_i and $a_i + w$ are either 0 or -1.

Rule 1: Ensure that $\delta\Sigma_0^i(w) = w$ by putting $w = 1$ and $a_i = -1$.

Table 5. Different cases for (w, x, y, z).

(I)	(II)	(III)	(IV)
$(w, -w, 0, 0)$	$(w, -w, 0, -w)$	$(w, -w, -w, 0)$	$(w, -w, -w, -w)$
(V)	(VI)	(VII)	(VIII)
$(w, -2w, 0, 0)$	$(w, -2w, 0, -w)$	$(w, -2w, -w, 0)$	$(w, -2w, -w, -w)$

Simplifying Majority. If two of the inputs are equal, then the output of $f_{MAJ}()$ is equal to this input. Based on this observation, we have the following rule.

Rule 2: Simplify each occurrence of f_{MAJ} by making two of the inputs equal.

This rule has several consequences. The function f_{MAJ} is used only in the definitions of x, y and z. Consider, for example x which, after the application of Rule 1, is equal to

$$x = -w - f_{MAJ}(a_i + w, a_{i-1}, a_{i-2}) + f_{MAJ}(a_i, a_{i-1}, a_{i-2}).$$

There are three ways to apply Rule 2 to this occurrence of f_{MAJ}. These are:

1. Set $a_{i-1} = a_{i-2}$ which implies $x = -w$;
2. set $a_{i-1} = a_i + w$, $a_i = a_{i-2}$ which implies that $x = -2w$;
3. set $a_{i-2} = a_i + w$, $a_i = a_{i-1}$ which also implies that $x = -2w$.

So applying Rule 2 to x implies that either $x = -w$ (in which case $a_{i-1} = a_{i-2}$) or $x = -2w$ (in which case either ($a_{i-1} = a_i + w$ and $a_i = a_{i-2}$) or ($a_{i-2} = a_i + w$ and $a_i = a_{i-1}$).

Similar reasoning applies to the expressions for y and z. Now, if we simultaneously apply Rule 2 to all the three occurrences of f_{MAJ}, then there are eight possible values of (w, x, y, z) which are listed as Cases (I) to (VIII) in Table 5. The related sufficient conditions are given in Table 6.

These sufficient conditions specify certain values for the registers $(a_{i-2}, a_{i-1}, a_i, a_{i+1}, a_{i+2})$ and (e_{i+1}, e_{i+2}). Actually, the conditions on the a-register values are independent and the conditions on the e-register values are obtained from these values using the CDE. Using Proposition 1, it is possible to set the values of $(W_{i-2}, \ldots, W_{i+2})$ to ensure that the $(a_{i-2}, \ldots, a_{i+2})$ obtain the required values. Consequently, we can ensure that any of the cases in Table 6 can be made to hold with probability one.

Note. If $w = 1$, then Case (I) of Table 5 corresponds to the NB local collision and Case (III) of Table 5 corresponds to the SS local collision. As we proceed, we will see that the other cases become unusable.

Table 6. Result of applying Rules 1 and 2. For this table, we have $w = 1$ and $a_i = -1$.

Case	a_{i-2}	a_{i-1}	a_i	a_{i+1}	a_{i+2}	e_{i+2}	e_{i+1}
I	α	α	-1	α	α	$-\Sigma_0(\alpha) + \alpha$	$1 + a_{i-3}$
II(a)	0	0	-1	0	-1	-1	$1 + a_{i-3}$
II(b)	-1	-1	-1	-1	0	1	$1 + a_{i-3}$
III(a)	-1	-1	-1	0	0	0	$2 + a_{i-3}$
III(b)	0	0	-1	-1	-1	1	a_{i-3}
IV(a)	-1	-1	-1	0	-1	-1	$2 + a_{i-3}$
IV(b)	0	0	-1	-1	0	2	a_{i-3}
V(a)	-1	0	-1	0	0	-1	$2 + a_{i-3}$
V(b)	0	-1	-1	-1	-1	1	$1 + a_{i-3}$
VI(a)	-1	0	-1	0	-1	-2	$2 + a_{i-3}$
VI(b)	0	-1	-1	-1	0	2	$1 + a_{i-3}$
VII(a)	-1	0	-1	-1	-1	0	$1 + a_{i-3}$
VII(b)	0	-1	-1	0	0	1	$2 + a_{i-3}$
VIII(a)	-1	0	-1	-1	0	1	$1 + a_{i-3}$
VIII(b)	0	-1	-1	0	-1	-1	$2 + a_{i-3}$

3.2 Simplifying $\delta W_{\mathrm{i}+4}$ to $\delta W_{\mathrm{i}+7}$

The expression for δW_{i+4} involves $\delta \Sigma_1^{i+3}(z)$ and $\delta f_{IF}^{i+3}(z, y, x)$. Joint simplification of the above two quantities is possible by ensuring that both e_{i+3} and $e_{i+3} + z$ are either 0 or -1.

1. If $z = 0$, then e_{i+3} can be either 0 or -1.
2. If $z = -w$, then we choose $e_{i+3} = 0$ if $w = 1$; and $e_{i+3} = -1$ if $w = -1$.

Similarly, simplification of δW_{i+5} is possible by ensuring that both w and $e_{i+4} + w$ are either 0 or -1.

For δW_{i+6} and δW_{i+7} we respectively ensure that e_{i+5} and e_{i+6} are either 0 or -1. The effect of these simplifications are summarized in Tables 7 and 8. In particular, the simplifying conditions and the resulting values of the respective δWs are shown. The condition on the values of e-register can be achieved by setting the corresponding message word W (see Proposition 1). So, any of the conditions in Tables 7 and 8 can be achieved.

Table 7. Summary of simplifying conditions for δW_{i+4} and δW_{i+5}. These simplifications require Rules 1 and 2 and so, in particular $w = 1$ in all these cases.

δW	Condition(s)	Value of δW
δW_{i+4}	$z = 0$, $e_{i+3} = 0$	$-w - x$
	$z = 0$, $e_{i+3} = -1$	$-w - y$
	$w = 1$, $z = -w$, $e_{i+3} = 0$	$e_{i+1} - e_{i+2} + y$
δW_{i+5}	$w = 1$, $e_{i+4} = -1$	$-w - x - y + e_{i+3} - e_{i+2}$

Table 8. Summary of simplifying conditions for δW_{i+6} and δW_{i+7}. These simplifications do not require these Rules 1 and 2 and consequently, w can be any n-bit word.

δW	Condition(s)	Value of δW
δW_{i+6}	$e_{i+5} = 0$	$-y - z$
	$e_{i+5} = -1$	$-y - w$
δW_{i+7}	$e_{i+6} = 0$	$-w - z$
	$e_{i+6} = -1$	$-z$

4 Obtaining Up To 22-Round Collisions

The basic idea is the following. Choose a suitable value for i and place the local collision from Steps i to $i + 8$. By placing we mean the following. Ensure that $\delta W_0, \ldots, \delta W_{i-1}$ are all zeros and introduce the required differences in $\delta W_i, \ldots, \delta W_{i+8}$. This creates a collision from Steps i to $i + 8$. Ensure that there are no further disturbances by setting δW_{i+9} to δW_{15} to be zero. This works well if we are interested in up to 16-round collisions.

For obtaining collisions on $r > 16$ rounds, we need to consider the message expansion. The initial words $W_0, \ldots, W_{15}$ are free and from W_{16} onwards, the words are computed using the message expansion recursion given by (2). For clarity, some word differences are shown in Table 9. The differences in the message words introduced in Steps i to $i + 8$ can possibly affect $\delta W_{16}, \delta W_{17}, \ldots, \delta W_{r-1}$. We ensure that the effects of these induced differences can be cancelled out and we have $\delta W_{16} = \delta W_{17} = \cdots = \delta W_{r-1} = 0$. This results in an r-round collision.

Table 9. Message expansion from W_{16} to W_{23}.

$\delta W_{16} = \delta\sigma_1(\delta W_{14}) + \delta W_9 + \delta\sigma_0(\delta W_1) + \delta W_0$
$\delta W_{17} = \delta\sigma_1(\delta W_{15}) + \delta W_{10} + \delta\sigma_0(\delta W_2) + \delta W_1$
$\delta W_{18} = \delta\sigma_1(\delta W_{16}) + \delta W_{11} + \delta\sigma_0(\delta W_3) + \delta W_2$
$\delta W_{19} = \delta\sigma_1(\delta W_{17}) + \delta W_{12} + \delta\sigma_0(\delta W_4) + \delta W_3$
$\delta W_{20} = \delta\sigma_1(\delta W_{18}) + \delta W_{13} + \delta\sigma_0(\delta W_5) + \delta W_4$
$\delta W_{21} = \delta\sigma_1(\delta W_{19}) + \delta W_{14} + \delta\sigma_0(\delta W_6) + \delta W_5$
$\delta W_{22} = \delta\sigma_1(\delta W_{20}) + \delta W_{15} + \delta\sigma_0(\delta W_7) + \delta W_6$
$\delta W_{23} = \delta\sigma_1(\delta W_{21}) + \delta W_{16} + \delta\sigma_0(\delta W_8) + \delta W_7$

18-Round Collisions. Deterministic 18-round collisions are easy to obtain by setting $i = 3$ (i.e., the local collision spans from $i = 3$ to $i{+}8 = 11$). So, we necessarily have $\delta W_j = 0$ for $j = 0, 1, 2, 12, 13, 14, 15$.

Additionally, we need to ensure that $\delta W_{16} = \delta W_{17} = 0$. From Table 9, we see that in the expression for δW_{16} the only possible non-zero term is $\delta W_9 = \delta W_{i+6}$. Similarly, in the expression for δW_{16}, the only possible non-zero term is $\delta W_{10} = \delta W_{i+7}$. By ensuring that $\delta W_{i+6} = \delta W_{i+7} = 0$, we will obtain $\delta W_{16} = \delta W_{17} = 0$. But, ensuring $\delta W_{i+6} = \delta W_{i+7} = 0$ can be easily done by setting a suitable condition from Table 8. For example, if $y = z = 0$, then the setting $e_{i+5} = 0$ and $e_{i+6} = -1$ ensures $\delta W_{i+6} = \delta W_{i+7} = 0$ for any choice of w. Using Proposition 1, the required values of e_{i+5} and e_{i+6} can be achieved by setting W_{i+6} and W_{i+7} to appropriate values. As a net result, we obtain deterministic 18-round collisions for any value of w.

20-Round Collisions. Set $i = 5$, i.e., the local collision spans from $i = 5$ to $i+8 = 13$, so that $\delta W_j = 0$ for $j = 0, \ldots, 4, 14, 15$. We need to ensure that $\delta W_{16} = \cdots = \delta W_{19} = 0$. From Table 9, we see that this can be achieved by setting $\delta W_9 = \delta W_{10} = \delta W_{11} = \delta W_{12} = 0$.

Since $i = 5$, this means that we have to set $\delta W_{i+4} = \delta W_{i+5} = \delta W_{i+6} = \delta W_{i+7} = 0$. The conditions for individually setting any of these to 0 are given in Tables 7 and 8. In the present case, we need to consider how to simultaneously set all of these to 0. In this situation, some conditions become infeasible. More precisely, certain conditions for obtaining $\delta W_{i+4} = 0$ are incompatible with certain conditions for obtaining $\delta W_{i+5} = 0$. The possible conditions for ensuring these two δWs to be zero are given in Table 10. In particular, we see that $z = 0$ in all cases. The conditions for setting $\delta W_{i+6} = 0$ and $\delta W_{i+7} = 0$ do not cause any conflict with other conditions. The set of conditions required for setting $\delta W_{i+4} = \delta W_{i+5} = \delta W_{i+6} = \delta W_{i+7} = 0$ are summarized in Table 11. Again achieving the appropriate values of the a and e-registers can be done using Proposition 1.

Table 10. Conditions for setting $\delta W_{i+4} = \delta W_{i+5} = 0$.

Case	w	x	y	z	e_{i+2}	e_{i+3}	e_{i+4}	Extra Condition
A	1	-1	0	0	0	0	-1	Case I
B	1	-1	-1	0	1	0	-1	Case III (b)
C	1	-2	-1	0	1	0	-1	Case VII (b)
D	1	-1	-1	0	0	-1	-1	Case III (a)
E	1	-2	0	0	1	-1	-1	Case V (b)
F	1	-2	-1	0	1	-1	-1	Case VII (b)

Table 11. Conditions for setting $\delta W_{i+4} = \delta W_{i+5} = \delta W_{i+6} = \delta W_{i+7} = 0$.

A row of Table 10 AND ($e_{i+5} = 0$ and $y = -z$) or ($e_{i+5} = -1$ and $y = -w$) AND $e_{i+6} = -1$.

Note. Tables 10 and 11 show that it is possible to deterministically set all the four δWs to zero in Case (A) which is the NB local collision. Consequently, it is possible to obtain *deterministic* 20-round collision using this local collision. This was not done in [15] but was later mentioned in [22].

21-Round Collisions. Set $i = 6$, i.e., the local collision spans from $i = 6$ to $i + 8 = 14$. We need to ensure that $\delta W_{16} = \cdots = \delta W_{20} = 0$. As in the case of 20-round collision, we set $\delta W_{i+4} = \delta W_{i+5} =$

$\delta W_{i+6} = \delta W_{i+7} = 0$ by a suitable set of conditions given by Table 11. So, we have $\delta W_j = 0$ for $j = 0, \ldots, 5, 10, 11, 12, 13, 15$. From Table 9, we see that if we can now achieve $\delta W_{16} = 0$, then we will have achieved the condition $\delta W_{16} = \cdots = \delta W_{20} = 0$.

From the structure of the differential path shown in Table 4, $\delta W_{14} = \delta W_{i+8} = -w$ and so

$$\delta W_{16} = \delta\sigma_1(\delta W_{14}) + \delta W_9. \qquad (6)$$

Consider $\delta W_9 = \delta W_{i+3}$ which by the differential path in Table 4 is equal to $z - \delta\Sigma_1^{i+2}(y) - \delta f_{IF}^{i+2}(y, x, w)$. To simplify this, we choose rows from Table 10 such that both e_{i+2} and $e_{i+2} + y$ are either 0 or -1. These are rows A and D.

In the case of row D, we have $\delta W_9 = -e_7 + e_6 + 2$; whereas for row A, we get $\delta W_9 = -1$. It is possible to deterministically satisfy the case for row D. However, row A (which is the NB local collision) cannot be used in the attack. This is due to the fact that there does not exist any word X such that $\sigma_0(X) - \sigma_0(X - 1) = -1$ either for SHA-256 or for SHA-512.

Since $i = 6$, the row of Table 6 corresponding to row D of Table 10 ensure that a_4, a_5, a_6, a_7, a_8 and e_8 are all fixed to particular values. Using CDE, we can now use a_3 to set e_7 to any specific value and then use a_2 to set e_6 to any specific value.

Now, the following strategy is used to ensure that $\delta W_{16} = 0$. Choose an arbitrary value for W_{14} and compute δ to be

$$\delta = \delta\sigma_1(\delta W_{14}) = \sigma_1(W_{14} + \delta W_{14}) - \sigma_1(W_{14}) = \sigma_1(W_{14} - w) - \sigma_1(W14).$$

Choose W_2 and W_3 to set a_2 and a_3 such that $e_7 - e_6 - 2 = -\delta$. From (6), it now follows that $\delta W_{16} = 0$. This gives a deterministic 21-round collision.

It is also possible to obtain deterministic 21-round collision by placing the local collision from Steps 7 to 15. Set $i = 7$ so that the local collision spans steps $i = 7$ to $i + 8 = 15$. In this case, set $\delta W_{i+4} = \delta W_{i+5} = \delta W_{i+6} = 0$ the sufficient condition for this being any row of Table 10 AND (($e_{i+5} = 0$, $y = -z$) or ($e_{i+5} = -1$, $y = -w$)). This ensures $\delta W_{11} = \delta W_{12} = \delta W_{13} = 0$. Now

$$\begin{aligned}\delta W_{16} &= \sigma_1(\delta W_{14}) + \delta W_9,\\ \delta W_{17} &= \sigma_1(\delta W_{15}) + \delta W_{10}.\end{aligned}$$

We have $\delta W_{15} = -w$ and by setting $e_{i+6} = 0$, we also have $\delta W_{14} = -w$. Also,

$$\begin{aligned}\delta W_9 = \delta W_{i+2} &= y - \delta\Sigma_1^{i+1}(x) - \delta f_{IF}^{i+1}(x, w, 0)\\ \delta W_{10} = \delta W_{i+3} &= -\delta\Sigma_1^{i+2}(y) - \delta f_{IF}^{i+2}(y, x, w).\end{aligned}$$

To simplify $\delta W_{10} = \delta W_{i+3}$ we choose rows from Table 10 such that both e_{i+2} and $e_{i+2} + y$ are either 0 or -1. These are rows A and D (which correspond to the NB and SS local collisions respectively). Similarly, to simplify $\delta W_9 = \delta W_{i+2}$, in row A we choose $e_{i+1} = 0$; and in row D we choose $e_{i+1} = -1$.

The overall strategy is now the following. Choose arbitrary values for W_{14} and W_{15} and compute $\delta_1 = \delta\sigma_1(\delta W_{14})$ and $\delta_2 = \delta\sigma_1(\delta W_{15})$, where $\delta W_{14} = \delta W_{15} = -w$. Now set $\delta W_9 = -\delta_1$ and $W_{10} = -\delta_2$ using W_3 and W_4 to set a_3 and a_4 and hence, using CDE to set e_7 and e_8 to desired values. This can be done deterministically.

We have sketched two ways of achieving deterministic 21-round collisions. In one case, the local collision spans from Step 6 to Step 14 and in the second case, the local collision spans from Step 7 to Step 15. For the first case, only the SS local collision can be used, while in the second case, both the SS and the NB local collisions can be used. The fact that the NB local collision can be used to obtain deterministic 21-round collisions was not mentioned in [15]; it was mentioned in [20].

The sketches above can be developed into detailed algorithms. We do not describe these algorithms. This is because below we describe in details a similar algorithm for constructing deterministic 22-round collisions.

4.1 22-Round Collisions

Set $i = 7$ so that the local collision spans from $i = 7$ to $i+8 = 15$. Use sufficient conditions from Table 11 to ensure that $\delta W_{i+4} = \delta W_{i+5} = \delta W_{i+6} = \delta W_{i+7} = 0$. So $\delta W_j = 0$ for $j = 0, \ldots, 6, 11, 12, 13, 14$. If we can now ensure that $\delta W_{16} = \delta W_{17} = 0$, then from Table 9, we will have $\delta W_j = 0$ for $j = 18, 19, 20, 21$ which will give rise to a 22-round collision. Under the conditions, we have

$$\left.\begin{array}{l} \delta W_{16} = \delta W_9 \\ \delta W_{17} = \delta\sigma_1(\delta W_{15}) + \delta W_{10}. \end{array}\right\} \tag{7}$$

So, if we can achieve $\delta W_9 = \delta W_{i+2} = 0$ and $\delta\sigma_1(\delta W_{15}) + \delta W_{10} = 0$, then we are done. Note that $\delta W_{15} = -w = -1$.

First, consider the condition on δW_{17}. To simplify $\delta W_{10} = \delta W_{i+3}$ we need to choose both e_{i+2} and $e_{i+2} + y$ to be 0 or -1. These imply that we have to use either row A or row D of Table 10 (which respectively correspond to the NB and the SS local collisions).

In case of row D, we have $\delta W_{10} = -e_8 + e_7 = 2$, whereas in the case of row A, we have $\delta W_{10} = -1$. The computation for row D is as follows. (A similar computation shows the value of δW_{10} for row A.)

$$\begin{aligned} \delta W_{10} &= -\delta f^9_{IF}(-1, -1, 1) - \delta\Sigma_1(-1) \\ &= -f_{IF}(e_9 - 1, f_9 - 1, g_9 + 1) + f_{IF}(e_9, f_9, g_9) - \Sigma_1(e_9 - 1) + \Sigma_1(e_9) \\ &= -f_{IF}(e_9 - 1, e_8 - 1, e_7 + 1) + f_{IF}(e_9, e_8, e_7) - \Sigma_1(e_9 - 1) + \Sigma_1(e_9) \\ &= -f_{IF}(-1, e_8 - 1, e_7 + 1) + f_{IF}(0, e_8, e_7) - \Sigma_1(-1) + \Sigma_1(0) \\ &= -(e_8 - 1) + e_7 - (-1) + 0 \\ &= -e_8 + e_7 + 2. \end{aligned}$$

If we want to use row A, then from (7) we need to have a value for W_{15} such that $\sigma_1(W_{15}+1) - \sigma_1(W_{15}) = 1$. There is no such value for W_{15} for both SHA-256 and SHA-512. Hence, row A, which correspond to the NB local collision, cannot be used.

So, we use row D which correspond to Case III(a) of Table 6. In this case, we see that $e_8 = e_{i+1} = 2 + a_{i-3} = 2 + a_4$. We set $a_4 = -2$, so that $e_8 = 0$ and $\delta W_{10} = e_7 + 2$. Setting a_4 to -2 is done using W_4 as in Proposition 1.

Choose an arbitrary value for W_{15} and set $\delta = -\delta\sigma_1(\delta W_{15})$ where $\delta W_{15} = -1$. Then use W_3 to set a_3 such that due to CDE, e_7 gets set to a particular value required to ensure that $e_7 + 2 = \delta W_{10} = \delta$, i.e., $e_7 = \delta - 2$. This computation is as follows.

$$\begin{aligned} \delta = e_7 + 2 &= a_3 + a_7 - \Sigma_0(a_6) - f_{MAJ}(a_6, a_5, a_4) + 2 \\ &= a_3 + (-1) - \Sigma_0(-1) - f_{MAJ}(-1, -1, -2) + 2 \\ &= a_3 - 1 - (-1) - (-1) + 2 \\ &= a_3 + 3. \end{aligned}$$

So, using W_3 we need to set $a_3 = \delta - 3$.

Now consider the condition on δW_{16} given in (7), i.e., the condition $\delta W_9 = 0$. Up to this point, the values of a_3 to a_6 have been fixed as follows: $a_3 = \delta - 3$, $a_4 = -2$, $a_5 = -1$, $a_6 = -1$. Noting that $i = 7$ and row D correspond to $(w, x, y, z) = (1, -1, -1, 0)$, from Table 4, we have

$$\begin{aligned}\delta W_9 = \delta W_{i+2} &= y - \delta \Sigma_1^{i+1}(x) - \delta f_{IF}^{i+1}(x, w, 0)\\ &= -y - \Sigma_1(e_{i+1} + x) + \Sigma_1(e_{i+1})\\ &\quad -f_{IF}(e_{i+1} + x, e_i + w, e_{i-1}) + f_{IF}(e_{i+1}, e_i, e_{i-1})\\ &= -1 - (-1) + 0 - f_{IF}(-1, e_7 + 1, e_6) + f_{IF}(0, e_7, e_6)\\ &= -e_7 - 1 + e_6\\ &= 2 - \delta - 1 + e_6.\end{aligned}$$

To obtain $\delta W_9 = 0$, we need to have $e_6 = \delta - 1$. Using CDE, we have

$$\begin{aligned}e_6 &= a_6 + a_2 - \Sigma_0(a_5) - f_{MAJ}(a5, a_4, a_3)\\ &= -1 + a_2 - (-1) - f_{MAJ}(-1, -2, \delta - 3)\end{aligned}$$

So, setting $a_2 = \delta - 1 + f_{MAJ}(-1, -2, \delta - 3)$ ensures, $e_6 = \delta - 1$ as required. This completes the description. All of the above steps can be written more explicitly in an algorithmic form. We provide this below.

Algorithm to Obtain 22-Round Collisions. We define two functions which return the required message word W_i to set the register value a_i or e_i to desired values, say desired_a and desired_e, at Step i. (See Proposition 1.) Equation 1 provides the definitions of these two functions.

1. W_to_set_register_A(Step i, desired_a, Current State $\{a_{i-1}, b_{i-1}, \ldots h_{i-1}\}$) :
   = (desired_a $-\Sigma_0(a_{i-1}) - f_{MAJ}(a_{i-1}, b_{i-1}, c_{i-1}) - \Sigma_1(e_{i-1}) - f_{IF}(e_{i-1}, f_{i-1}, g_{i-1}) - h_{i-1} - K_i$)
2. W_to_set_register_E(Step i, desired_e, Current State $\{a_{i-1}, b_{i-1}, \ldots h_{i-1}\}$) :
   = (desired_e $-d_{i-1} - \Sigma_1(e_{i-1})$ $-f_{IF}(e_{i-1}, f_{i-1}, g_{i-1}) - h_{i-1} - K_i$)

Using these functions, the complete algorithm to obtain message pairs leading to deterministic 22-round collisions for SHA-2 family is described in Table 12.

A Remark on the NB Local Collision. We have mentioned that if we place the local collision from Steps 7 to 15, then row A of Table 10 cannot be used to obtain a deterministic 22-round collision. Row A corresponds to the NB local collision.

We considered the issue of whether it is possible to place the NB local collision from Steps 8 to 16 to obtain a 22-round collision (which may not be deterministic). In this case, the local collision will end at Step 16 and hence $\delta W_{16} = -1$. Recall from Table 9, that a difference in δW_{16} will affect δW_{18}. We would like to have $\delta W_{18} = 0$ so as to ensure that there are no differences after the local collision ends. Again from Table 9 and the fact that the local collision spans Steps 8 to 16, to achieve $\delta W_{18} = 0$, we need to have $\delta\sigma_1(\delta W_{16}) + \delta W_{11} = 0$.

More generally, we considered the situation, where the NB local collision spans Steps i to $(i+8)$, with $i \geq 8$ and we require $\delta W_{i+10} = 0$. From Table 9, the last condition is achieved if $\delta\sigma_1(\delta W_{i+8}) + \delta W_{i+3} = 0$. Note that $\delta W_{i+8} = -1$.

For SHA-512, using the NB local collision makes achieving the condition $\delta\sigma_1(\delta W_{i+8}) + \delta W_{i+3} = 0$ difficult. This is because of the fact that there is a "gap" in the values of $|\delta W_{i+3}|$ and $|\delta\sigma_1(\delta W_{i+8})|$. In Appendix B, we show that the probability of $|\delta W_{i+3}| \geq 2^j$ is less than $1/2^{j-1}$; and for any 64-bit value for W_{i+8}, $|\sigma_1(W_{i+8}) - \sigma_1(W_{i+8} - 1)| \geq 2^{42} + 2^{39} + 2^{38} + 2^{36} - 2^3$. As a consequence, to achieve

Table 12. Deterministic algorithm to obtain message pairs leading to collisions for 22-round SHA-2.

```
external  W_to_set_register_A(Step $i$, desired_a, Current State $\{a_{i-1}, b_{i-1}, \dots h_{i-1}\}$) :
    Returns the required message $W_i$ to be used in step $i$ so that $a_i$ is set to the given value.
external  W_to_set_register_E(Step $i$, desired_e, Current State $\{a_{i-1}, b_{i-1}, \dots h_{i-1}\}$) :
    Returns the required message $W_i$ to be used in step $i$ so that $e_i$ is set to the given value.
First Message words:
1. Select $W_0$, $W_1$, $W_{14}$ and $W_{15}$ randomly.
2. Set DELTA $= \sigma_1(W_{15}) - \sigma_1(W_{15} - 1)$.
3. Run Steps 0 and 1 of hash evaluation to define $\{a_1, b_1, \dots h_1\}$.
4. Choose $W_2$ = W_to_set_register_A(2, DELTA $- 1 + f_{MAJ}(-1, -2,$ DELTA $- 3)$, $\{a_1, b_1, \dots h_1\}$).
5. Run Step 2 of hash evaluation to define $\{a_2, b_2, \dots h_2\}$.
6. Choose $W_3$ = W_to_set_register_A(3, DELTA $- 3$, $\{a_2, b_2, \dots h_2\}$).
7. Run Step 3 of hash evaluation to define $\{a_3, b_3, \dots h_3\}$.
8. Choose $W_4$ = W_to_set_register_A(4, $-2$, $\{a_3, b_3, \dots h_3\}$).
9. Run Step 4 of hash evaluation to define $\{a_4, b_4, \dots h_4\}$.
10. Choose $W_5$ = W_to_set_register_A(5, $-1$, $\{a_4, b_4, \dots h_4\}$).
11. Run Step 5 of hash evaluation to define $\{a_5, b_5, \dots h_5\}$.
12. Choose $W_6$ = W_to_set_register_A(6, $-1$, $\{a_5, b_5, \dots h_5\}$).
13. Run Step 6 of hash evaluation to define $\{a_6, b_6, \dots h_6\}$.
14. Choose $W_7$ = W_to_set_register_A(7, $-1$, $\{a_6, b_6, \dots h_6\}$).
15. Run Step 7 of hash evaluation to define $\{a_7, b_7, \dots h_7\}$.
16. Choose $W_8$ = W_to_set_register_A(8, 0, $\{a_7, b_7, \dots h_7\}$).
17. Run Step 8 of hash evaluation to define $\{a_8, b_8, \dots h_8\}$.
18. Choose $W_9$ = W_to_set_register_A(9, 0, $\{a_8, b_8, \dots h_8\}$).
19. Run Step 9 of hash evaluation to define $\{a_9, b_9, \dots h_9\}$.
20. Choose $W_{10}$ = W_to_set_register_E(10, $-1$, $\{a_9, b_9, \dots h_9\}$).
21. Run Step 10 of hash evaluation to define $\{a_{10}, b_{10}, \dots h_{10}\}$.
22. Choose $W_{11}$ = W_to_set_register_E(11, $-1$, $\{a_{10}, b_{10}, \dots h_{10}\}$).
23. Run Step 11 of hash evaluation to define $\{a_{11}, b_{11}, \dots h_{11}\}$.
24. Choose $W_{12}$ = W_to_set_register_E(12, $-1$, $\{a_{11}, b_{11}, \dots h_{11}\}$).
25. Run Step 12 of hash evaluation to define $\{a_{12}, b_{12}, \dots h_{12}\}$.
26. Choose $W_{13}$ = W_to_set_register_E(13, $-1$, $\{a_{12}, b_{12}, \dots h_{12}\}$).
Second message words:
27. Define $\delta W_i = 0$ for $i \in \{0, 1, 2, 3, 4, 5, 6, 9, 11, 12, 13, 14\}$.
28. Define $\delta W_7 = 1$ and $\delta W_{15} = -1$.
29. Define $\delta W_8 = -1 - f_{IF}(e_7 + 1, f_7, g_7) + f_{IF}(e_7, f_7, g_7) - \Sigma_1(e_7 + 1) + \Sigma_1(e_7)$. (Refer Table 4.)
30. Define $\delta W_{10} = -f_{IF}(e_9 - 1, f_9 - 1, g_9 + 1) + f_{IF}(e_9, f_9, g_9) - \Sigma_1(e_9 - 1) + \Sigma_1(e_9)$. (Refer Table 4.)
31. Compute $W'_i = W_i + \delta W_i$ for $0 \le i \le 15$.
```

$\delta\sigma_1(\delta W_{i+8}) + \delta W_{i+3} = 0$, we need to have $|\delta W_{i+3}| > 2^{42}$, an event which occurs with probability less than 2^{-41}.

The above probability computation is over uniform random choices of W_{i+8} and W_{i+3}. In fact, this was one of the factors that had led us to focus only on the SS local collision. It was shown in [9] that the NB local collision can be used to obtain 23 and 24-round SHA-512 collision. However, the time complexities of the NB local collision attack is more than that of the SS local collision attack. This fact is possibly attributable to the "gap" in the values of $|\delta W_{i+3}|$ and $|\delta\sigma_1(\delta W_{i+8})|$ mentioned above.

5 A General Idea for Obtaining 23 and 24-Round Collisions

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

For obtaining collisions up to 22 rounds, we also needed to consider message expansion. But, 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 further), 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 strategy followed so far. 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-round collision, then δW_{16} is necessarily $-w$ and we will require $\delta W_{17} = \cdots = \delta W_{22} = 0$.

5.1 A Class of Local Collisions

A local collision of the type shown in Table 4 is completely determined by the values of w, x, y and z which in turn determine the values of δW_i to δW_{i+8}. We need to consider some special values for the δWs. Let

$$(\delta W_i, \ldots, \delta W_{i+8}) = (w, -w, \delta_1, \delta_2, 0, 0, 0, u, -w) \text{ with } w = 1. \tag{8}$$

The value of u is either 0 or w and the values of δ_1 and δ_2 will be explained later. Using the form of the δWs from Table 4, Equation 8 gives rise to the following 9 equations. We will refer to them as (9.1) to (9.9).

$$\left.\begin{array}{llll}
(1) & \delta W_i & = & = w; \\
(2) & \delta W_{i+1} & = x - \delta\Sigma_1^i(w) - \delta f_{IF}^i(w,0,0) & = -w; \\
(3) & \delta W_{i+2} & = y - \delta\Sigma_1^{i+1}(x) - \delta f_{IF}^{i+1}(x,w,0) & = \delta_1; \\
(4) & \delta W_{i+3} & = z - \delta\Sigma_1^{i+2}(y) - \delta f_{IF}^{i+2}(y,x,w) & = \delta_2; \\
(5) & \delta W_{i+4} & = -w - \delta\Sigma_1^{i+3}(z) - \delta f_{IF}^{i+3}(z,y,x) & = 0; \\
(6) & \delta W_{i+5} & = -x - \delta\Sigma_1^{i+4}(w) - \delta f_{IF}^{i+4}(w,z,y) & = 0; \\
(7) & \delta W_{i+6} & = -y - \delta f_{IF}^{i+4}(0,w,z) & = 0; \\
(8) & \delta W_{i+7} & = -z - \delta f_{IF}^{i+4}(0,0,w) & = u; \\
(9) & \delta W_{i+8} & = & = -w.
\end{array}\right\} \tag{9}$$

The values of x, y and z from (4) are the following.

$$x = -\delta\Sigma_0^i(w) - \delta f_{MAJ}^i(w,0,0); \quad y = -\delta f_{MAJ}^{i+1}(0,w,0); \quad z = -\delta f_{MAJ}^{i+2}(0,0,w).$$

We now set conditions on the values for a and the e registers to obtain desired values for x, y and z and also to simplify the values of δWs. Using the kind of analysis done to obtain Rules 1 and 2, the following are easy to verify.

1. If $a_i = -1$ and $a_{i-1} = a_{i-2} = \alpha$, then $x = -1$.
2. If $a_{i+1} = a_{i-1}$, then $y = 0$; if $a_{i+1} = \overline{a_{i-1}}$, then $y = -1$.
3. If $a_{i+2} = a_{i+1}$, then $z = 0$; if $a_{i+2} = \overline{a_{i+1}}$, then $z = -1$.

Note. In our analysis of up to 22-round SHA-2, we saw that $z = 0$ arose as a necessary condition. Motivated by this, we will continue to work with $z = 0$. So, we will have $a_{i+2} = a_{i+1}$. Let this common value be β. Further, if $\beta = \alpha$, then $y = 0$ and if $\beta = \overline{\alpha}$, then $y = -1$. These and other values of a and e registers are shown in Table 13. We note the following.

1. If $y = 0$, then $\lambda = \alpha - \Sigma_0(\alpha)$.
2. If $y = -1$, then $\lambda = \alpha + \overline{\alpha} + 1 - \Sigma_0(\overline{\alpha}) = -\Sigma_0(\overline{\alpha})$.

At later point in the analysis, we will be obtaining λ and will require to obtain a corresponding value for α. In the case $y = -1$, $\alpha = \overline{\Sigma_0^{-1}(-\lambda)}$ and it is easy to obtain α from λ. The case $y = 0$ is not so simple. For SHA-256, one works with 32-bit words and then obtaining α from λ can be done by exhaustive search; however, for SHA-512, one has to work with 64-bit words and then things become more difficult. This is one of the reasons why it is more convenient to work with $y = -1$. (Note that $(w, x, y, z) = (1, -1, 0, 0)$ corresponds to the NB local collision, whereas $(w, x, y, z) = (1, -1, -1, 0)$ corresponds to the SS local collision.)

Table 13. Values of a and e register for the δWs given by (8) to hold. The value of u is either 0 or w. We have $w = 1$, $x = -1$ and $z = 0$. If $y = 0$, then $\beta = \alpha$, while if $y = -1$, then $\beta = \overline{\alpha}$. By CDE, we have $\lambda = \beta + \alpha - \Sigma_0(\beta) - f_{MAJ}(\beta, -1, \alpha)$. 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	$\lambda+y$	-1	y	$-1-u$

The values shown in Table 13 have been chosen so that the conditions on δW_{i+1} and δW_{i+5} to δW_{i+7} hold with probability one. Consider, for example, δW_{i+1}. From (9.2), we have

$$\begin{aligned}
\delta W_{i+1} &= x - \delta\Sigma_1^i(w) - \delta f_{IF}^i(w, 0, 0) \\
&= x - (\Sigma_1(e_i + w) - \Sigma_1(e_i)) - (f_{IF}(e_i + w, e_{i-1}, e_{i-2}) - f_{IF}(e_i, e_{i-1}, e_{i-2})) \\
&= -1 - (0 - (-1)) - (e_{i-2} - e_{i-1}) \\
&= -2 - \gamma + \gamma + 1 \\
&= -1.
\end{aligned}$$

Similarly, Equations (9.6), (9.7) and (9.8) can be verified. Equations (9.3), (9.4) and (9.5) on the other hand give rise to the following conditions on the values of α, γ and μ.

$$\left.\begin{array}{l}
\delta_1 = y - \Sigma_1(\mu + x) + \Sigma_1(\mu) - f_{IF}(\mu + x, 0, \gamma + 1) + f_{IF}(\mu, -1, \gamma + 1) \\
\delta_2 = -\Sigma_1(\lambda + y) + \Sigma_1(\lambda) - f_{IF}(\lambda + y, \mu + x, 0) + f_{IF}(\lambda, \mu, -1) \\
w = -f_{IF}(\lambda + y, \lambda + y, \mu + x) + f_{IF}(\lambda + y, \lambda, \mu).
\end{array}\right\} \qquad (10)$$

The special case of these equations with $y = 0$ have been reported in [9] and a method for solving them has been discussed. The method to solve these equations is different for SHA-256 and for SHA-512. Next, we discuss methods to solve (10) for the case $y = -1$.

5.2 Solving (10) for $y = -1$

The following provides an outline of the method to solve (10) for μ, γ and λ when $y = -1$ and δ_1 and δ_2 are given. From λ, we obtain α.

- The third equation holds with probability 1 if both λ and μ are odd.
- Given that λ and μ are odd, the second equation simplifies to $\delta_2 = -\Sigma_1(\lambda - 1) + \Sigma_1(\lambda) + \overline{(\lambda - 1)}$. For a given odd value of δ_2 occurring in the distribution of $\sigma_1(W) - \sigma_1(W - 1)$, it is possible to solve this equation for odd λ.
- Given such a λ, it is easy to solve the equation $\lambda = -\Sigma_0(\overline{\alpha})$ to obtain a suitable value of α, since Σ_0 is an invertible mapping for both SHA-256 and SHA-512.
- For the first equation, the term $-f_{IF}(\mu - 1, 0, \gamma + 1) + f_{IF}(\mu, -1, \gamma + 1)$ is equal to μ, if γ is odd. This term is equal to $\mu - 1$ if γ is even. Further, we note that $-\Sigma_1(\mu - 1) + \Sigma_1(\mu)$ is always even for both SHA-256 and SHA-512. Thus taking an arbitrary odd value of γ, the first equation is in the single variable μ and can be solved easily for a given δ_1.

Now we provide proofs of the observations above.

Lemma 1 *If $y = -1$, then the third equation of (10) is satisfied for any odd λ and odd μ.*

Proof. We have to show that

$$1 = -f_{IF}(\lambda - 1, \lambda - 1, \mu - 1) + f_{IF}(\lambda - 1, \lambda, \mu).$$

The quantities λ and $\lambda - 1$ differ only in their least significant bit since λ is odd. Similarly, μ and $\mu - 1$ differ only in their least significant bit since μ is odd. Let x_i denote the i^{th} bit of x, then λ_0=1, $(\lambda - 1)_0 = 0$, $\mu_0 = 1$ and $(\mu - 1)_0 = 0$. Let $(\lambda - 1)_i = \lambda_i = 1$ and $(\lambda - 1)_j = \lambda_j = 0$ for some non-zero indices i and j. Also, let $\mu_i = b_1$ and $\mu_j = b_2$ for these bit positions i and j. Now we are ready to write the bit patterns of the quantities occuring in the third equation.

bit	$63 \ldots\ i \ldots\ j \ldots\ 0$
$\lambda - 1$	$. \ \ldots\ 1 \ldots\ 0 \ldots\ 0$
λ	$. \ \ldots\ 1 \ldots\ 0 \ldots\ 1$
μ	$. \ \ldots\ b_1 \ldots\ b_2 \ldots\ 1$
$f_{IF}(\lambda - 1, \lambda, \mu)$	$. \ \ldots\ 1 \ldots\ b_2 \ldots\ 1$

Similarly,

bit	$63 \ldots\ i \ldots\ j \ldots\ 0$
$\lambda - 1$	$. \ \ldots\ 1 \ldots\ 0 \ldots\ 0$
$\lambda - 1$	$. \ \ldots\ 1 \ldots\ 0 \ldots\ 0$
$\mu - 1$	$. \ \ldots\ b_1 \ldots\ b_2 \ldots\ 0$
$f_{IF}(\lambda - 1, \lambda - 1, \mu - 1)$	$. \ \ldots\ 1 \ldots\ b_2 \ldots\ 0$

From the two bit patterns above, we get that

$$f_{IF}(\lambda - 1, \lambda, \mu) - f_{IF}(\lambda - 1, \lambda - 1, \mu - 1) = 1.$$

□

Lemma 2 *Let $y = -1$. For odd λ and odd μ, the second equation of (10) simplifies to* $\delta_2 = -\Sigma_1(\lambda - 1) + \Sigma_1(\lambda) + \overline{(\lambda - 1)}$.

Proof. Consider the following expression

$$-f_{IF}(\lambda - 1, \mu - 1, 0) + f_{IF}(\lambda, \mu, -1).$$

Similar to the proof of the previous lemma, we consider the bit patterns of the quantities occuring in the above equation. Let $\lambda_i = 1$ and $\lambda_j = 0$ for some non-zero i, j. Also, let $\mu_i = b_1$ and $\mu_j = b_2$. Then the following bit patterns can be seen for the various quantities.

bit	63 ...	i ...	j ...	0
λ	. ...	1 ...	0 ...	1
μ	. ...	b_1 ...	b_2 ...	1
-1	1 ...	1 ...	1 ...	1
$f_{IF}(\lambda, \mu, -1)$	. ...	b_1 ...	1 ...	1

Similarly,

bit	63 ...	i ...	j ...	0
$\lambda - 1$	. ...	1 ...	0 ...	0
$\mu - 1$	. ...	b_1 ...	b_2 ...	0
0	0 ...	0 ...	0 ...	0
$f_{IF}(\lambda - 1, \mu - 1, 0)$	. ...	b_1 ...	0 ...	0

From the two bit patterns above, we get that $f_{IF}(\lambda, \mu, -1)$ and $f_{IF}(\lambda - 1, \mu - 1, 0)$ will have the same bit value whenever the corresponding bit of λ is 1 and different bit value whenever the corresponding bit of λ is 0, except the least significant bit which will always be different. Comparing this difference with the bit pattern $\overline{\lambda - 1}$, we obtain

$$f_{IF}(\lambda, \mu, -1) - f_{IF}(\lambda - 1, \mu - 1, 0) = \overline{\lambda - 1}.$$

This completes the proof.

□

Lemma 3 *Let $y = -1$. For odd μ and odd γ, the first equation of (10) simplifies to* $\delta_1 = -1 - \Sigma_1(\mu - 1) + \Sigma_1(\mu) + \mu$.

Proof. By considering the bit patterns of μ, $\mu - 1$ and $\gamma + 1$ the following can be proved in a manner similar to the previous two lemmas.

$$f_{IF}(\mu, -1, \gamma + 1) - f_{IF}(\mu - 1, 0, \gamma + 1) = \begin{cases} \mu & \text{if } \gamma \text{ is odd.} \\ \mu - 1 & \text{if } \gamma \text{ is even.} \end{cases}$$

Substituting the above value in the equation for δ_1 gives the required proof.

□

SHA-256. For SHA-256 we did not solve the second equation explicitly since random search is itself good enough, producing a solution in few seconds. Solving all the three equations for α, γ and μ can be done in a few seconds on a current PC. Examples of values of (δ_1, δ_2) and the solutions to (10) for λ, γ and μ are provided in Table 14. The value of α is obtained from λ as explained earlier. The justification for choosing these particular values for the δs as well as the explanation for the first column will be provided later.

Table 14. Values leading to collisions for different number of steps of SHA-256. The value of i denotes the start point of the local collision, i.e., the local collision is placed from Step i to $i+8$.

(# rnds, i)	δ_1	δ_2	u	α	λ	γ	μ
(23, 8)	0	ff006001	0	32b308b2	051f9f7f	684e62b7	041fff81
(23, 9) (24, 10)	00006000	ff006001	1	32b308b2	051f9f7f	98e3923b	fbe05f81

SHA-512. It is possible to solve (10) for SHA-512 as well, although we require a different approach than SHA-256. The main difference is in solving the first and the second equations. Since now 64-bit quantities are involved, it is no longer possible to solve the first and second equations by exhaustive search. We describe a method to solve the second equation with the aid of an example.

Examples of values of (δ_1, δ_2) and the solutions to (10) for λ, γ and μ are provided in Table 15 and the value of α is obtained from λ as explained earlier. As in the case of SHA-256, the justification for choosing these particular values for the δs as well as the explanation for the first column will become clear later.

Table 15. Values leading to collisions for different number of steps of SHA-512. The value of i denotes the start point of the local collision, i.e., the local collision is placed from Step i to $i+8$.

(# rnds, i)	δ_1	δ_2	u	α	λ	γ	μ
(23, 8)	0	600000000237	0	7201b90f9f8df85e	3e000007ffdc9	1	43fffff800001
(23, 9) (24, 10)	200000000008	600000000237	1	7201b90f9f8df85e	3e000007ffdc9	1	45fffff800009

Solving the Second Equation of (10) For SHA-512. As shown in Lemma 2, for odd λ the second equation simplifies to

$$\delta_2 = -\Sigma_1(\lambda - 1) + \Sigma_1(\lambda) + \overline{(\lambda - 1)}.$$

We need to get an odd λ satisfying the above equation for a given value of δ_2. Since $-\Sigma_1(\lambda-1)+\Sigma_1(\lambda)$ is always even and $\overline{(\lambda - 1)}$ is odd due to our choice of odd λ, we require δ_2 to be odd. This equation can be solved by hand. We explain the method to solve this equation for $\delta_2 = \mathtt{600000000237}$.

First note that $\Sigma_1(x)$ is the XOR addition of 3 n-bit quantities which are rotated/shifted forms of x. If λ is odd, then λ and $\lambda - 1$ differ only in the least significant bit. Therefore, the bit patterns of $\Sigma_1(\lambda)$ and $\Sigma_1(\lambda - 1)$ will be same except at 3 bit positions. These 3 bit positions are indexed by 23, 46 and 50. By the structure of Σ_1 function and using the fact that λ is odd (i.e. $\lambda_0 = 1$), we have the following

$$\begin{aligned}
b_1 &= (\Sigma_1(\lambda))_{23} = \lambda_0 \oplus \lambda_{37} \oplus \lambda_{41} = 1 \oplus \lambda_{37} \oplus \lambda_{41}, \\
b_2 &= (\Sigma_1(\lambda))_{46} = \lambda_0 \oplus \lambda_{23} \oplus \lambda_{60} = 1 \oplus \lambda_{23} \oplus \lambda_{60}, \\
b_3 &= (\Sigma_1(\lambda))_{50} = \lambda_0 \oplus \lambda_4 \ \oplus \lambda_{27} = 1 \oplus \lambda_4 \ \oplus \lambda_{27}.
\end{aligned}$$

Also, because $(\lambda - 1)_0 = 0$, we have $(\Sigma_1(\lambda - 1))_{23} = \overline{b_1}$, $(\Sigma_1(\lambda - 1))_{46} = \overline{b_2}$ and $(\Sigma_1(\lambda - 1))_{60} = \overline{b_3}$.

Now consider the bit pattern of various quantities as follows.

bit	63 ... 50 ... 46 ... 23 ... 0
$A = \Sigma_1(\lambda - 1)$	.... $\overline{b_3}$ ... $\overline{b_2}$ ... $\overline{b_1}$ .... .
$B = \Sigma_1(\lambda)$	.... b_3 ... b_2 ... b_1 .... .
$A - B$	.... 1 0... 0
δ_2	....
$A - B + \delta_2$	....

We require the quantity $(A - B + \delta_2)$ to be equal to $\overline{(\lambda - 1)}$. It is clear from the bit pattern above that the lowest 23 bits (indexed from 0 to 22) of $(A - B + \delta_2)$ will be same as those of δ_2. Equating these bits to corresponding bits of $\overline{(\lambda - 1)}$, we immediately get the lowest 23 bits of λ.

Now consider the bits between 23 and 46 of $(A - B)$. It is clear that all these bits will be equal. Further, all these bits will be equal to 1 if $b_1 = 1$ due to the borrow while subtracting B from A at bit position 23. Similarly, all these bits of $(A - B)$ will be equal to 0 if $b_1 = 0$. Our choice of δ_2 has all these bits equal to zero, hence the term $(A - B + \delta_2)$ will too have all these bits equal. But since this term is equal to $\overline{(\lambda - 1)}$, all these bits of $(\lambda - 1)$ will also be equal. Finally, note that λ and $(\lambda - 1)$ differ only in the lowest bit position, hence all the bits between 23 and 46 of λ will also be equal. In particular, we will have $\lambda_{37} = \lambda_{41}$, hence we have that $b_1 = 1 \oplus \lambda_{37} \oplus \lambda_{41} = 1$.

Continuing reasoning on bit positions in this way, for any given δ_2, either we can solve for λ or determine that a solution does not exist. For $\delta_2 =$ `600000000237` we obtained the solution $\lambda =$ `3e000007ffdc9`. Note that the method explained above does not require any particular structure of the bits of δ_2. As another example, we also solved for $\delta_2 =$ `19fffffffdd9` and obtained the solution as $\lambda =$ 2200000800227.

Note.

1. The first equation can be solved in a similar manner for μ for a given δ_1.
2. It is possible to design an algorithm to do the task described above. But, such an algorithm will be complicated. Since we are interested in solving for a single value of δ_2, we chose not to describe and implement an algorithm. The method of solving by hand is good enough.

6 Finding 23 and 24-Round Collisions

We show that by suitably placing a local collision of the type described in Section 5.1 and using proper values for α, γ and μ, it is possible to obtain several 23 and 24-round collisions for SHA-2. For the description below, we will be considering the SS local collision, i.e., $(w, x, y, z) = (1, -1, -1, 0)$.

6.1 23-Round Collisions

There are two options of placing the SS local collision. From Step $i = 8$ to Step $i + 8 = 16$ and from Step $i = 9$ to Step $i + 8 = 17$. This gives rise to two kinds of 23-round collisions for SHA-2.

Case $i = 8$. The local collision is started at $i = 8$ and ends at $i = 16$.

We have $(w, x, y, z) = (1, -1, -1, 0)$ and $\beta = \overline{\alpha}$. Also, we set $u = 0$ 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 $\delta = \delta_2$. The value of δ_2 has to be chosen so that (10) has a solution. The time complexity of the algorithm depends on freq_δ (see Section 2.3 for the meaning of freq_δ) as explained below, so, one would like to choose δ such that freq_δ is as high as possible. At the same time, we have to ensure that (10) can be solved for the particular value of δ. Our choices of δ given in the rows with $(23, 8)$ of Tables 14 and 15 have the highest value of freq_δ for which it is possible to solve (10).

Since the local collision ends at Step 16, from Table 8 it necessarily follows that $\delta W_{16} = -1$. To obtain a 23-round collision, we want to ensure that $\delta W_{17} = \cdots = \delta W_{22} = 0$. From Table 9, (8) and the fact that $\delta W_j = 0$ for $0 \leq j \leq 7$, it follows that the condition $\delta W_{17} = \cdots = \delta W_{22} = 0$ is achieved if we can ensure $\delta W_{18} = 0$. Again, from Table 9, we have

$$\delta W_{18} = \delta\sigma_1(W_{16}) + \delta W_{11}. \tag{11}$$

So, for δW_{18} to be zero, we need $\delta = \delta W_{11} = -\delta\sigma_1(W_{16})$, 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. (This is the reason why we analysed the differential behaviour of σ_1 in Section 2.3.) The word W_{16} is defined using message recursion and so, we cannot control this word directly. Instead, we analyse which message words can be used to control W_{16}.

First, let us consider which register values need to be set to specific values. Since $i = 8$, from Table 13, we see that a_6 to a_{10} and e_6 to e_{14} get defined. Using CDE, the value of e_{10} is actually determined by the values of a_6 to a_{10}. Using CDE, the values of e_9 down to e_6 determine the values of a_5 down to a_2. So, the values of a_2 to a_{10} and the values of e_{11} to e_{14} are fixed.

From message recursion, the expression for W_{16} is the following.

$$W_{16} = \sigma_1(W_{14}) + W_9 + \sigma_0(W_1) + W_0.$$

From the update function of the e-register, we have

$$W_{14} = e_{14} - (\varSigma_1(e_{13}) + f_{IF}(e_{13}, e_{12}, e_{11}) + a_{10} + e_{10} + K_{14}).$$

In this equation, all values other than W_{14} have already been fixed. So, W_{14} and hence $\sigma_1(W_{14})$ have fixed values. Let us now consider W_9. From the update function of the a-register, we can write

$$W_9 = a_9 - \varSigma_0(a_8) - f_{MAJ}(a_8, a_7, a_6) - \varSigma_1(e_8) - f_{IF}(e_8, e_7, e_6) - e_5 - K_9.$$

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

$$e_5 = a_5 + a_1 - \varSigma_0(a_4) - f_{MAJ}(a_4, a_3, a_2).$$

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. Now,

$$a_1 = \varSigma_0(a_0) + f_{MAJ}(a_0, b_0, c_0) + \varSigma_1(e_0) + f_{IF}(e_0, f_0, g_0) + h_0 + K_1 + W_1$$

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 the initialization vector and hence are constants. Thus, we can write $a_1 = \varPhi(W_0) + W_1$, where

$$\varPhi(W_0) = \varSigma_0(a_0) + f_{MAJ}(a_0, b_0, c_0) + \varSigma_1(e_0) + f_{IF}(e_0, f_0, g_0) + h_0 + K_1.$$

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

$$\begin{aligned} W_{16} &= \sigma_1(W_{14}) + W_9 + \sigma_0(W_1) + W_0 \\ &= \sigma_1(W_{14}) + C - \varPhi(W_0) - W_1 + \sigma_0(W_1) + W_0 \\ &= D - \varPhi(W_0) - W_1 + \sigma_0(W_1) + W_0. \end{aligned} \tag{12}$$

We need to obtain W_0 and W_1 such that the value of W_{16} given by (12) satisfies the condition $\sigma_1(W_{16} - 1) - \sigma_1(W_{16}) = -\delta$ and then using (11) we obtain $\delta W_{18} = 0$ giving us the required condition of $\delta W_{17} = \cdots = \delta W_{22} = 0$.

Once W_0, W_1 have been obtained, a collision can be constructed in a manner similar to that for the 22-round case and as shown in Table 12. The idea is to first run SHA-2 for two steps using W_0 and W_1. This determines the registers $(a_1, \ldots, h_1)$. Now, using Proposition 1, run SHA-2 step-by-step using W_i to set a_i to the desired value for $2 \leq i \leq 10$. Then run SHA-2 step-by-step using W_i to set e_i to the desired value for $11 \leq i \leq 14$. Finally, choose any value for W_{15}. The values of W_i' are determined by the values of W_i and δW_i for $0 \leq i \leq 15$. This gives a colliding message pair $(W_0, \ldots, W_{15})$ and $(W_0', \ldots, W_{15}')$.

Estimate of Computation Effort. The main computational effort is in solving (12) for W_0 and W_1 such that $\sigma_1(W_{16} - 1) - \sigma_1(W_{16}) = -\delta$. We did not attempt an analytic solution. Instead, we tried random choices of W_0 and W_1 until we found a suitable W_{16}. There are freq_δ values of W_{16} for which $\sigma(W_{16}) - \sigma(W_{16} - 1)$ equals δ. 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-round SHA-2 computations.

SHA-256. The value of δ given in Table 14 is such that $\mathsf{freq}_\delta = 2^{16}$. (See Table 2 in Section 2.3.) So, the complexity of finding 23-round SHA-256 collision is about $2^{11.5}$ tries of 23-round SHA-256 computations. A message pair colliding for 23-round SHA-256 is given in Table 18 of Section A.

SHA-512. In this case, we have estimates on freq_δ. (Again, see Section 2.3 for discussion on this issue.) For the particular value of δ given in Table 15, our estimate is $\mathsf{freq}_\delta \approx 2^{43}$. (See Table 3.) So, the effort required is about $\frac{\mathsf{freq}_\delta}{2^{4.5}} = \frac{2^{21}}{2^{4.5}} = 2^{16.5}$ trials of 23-round SHA-512. A message pair colliding for 23-round SHA-512 is given in Table 21 of Section A.

Case $i = 9$. It is possible to place the local collision from Step 9 to Step 17 and then perform an analysis to show that it is possible to obtain 23-round collisions for both $y = 0$ and $y = -1$. We do not provide these details, since a similar technique with an additional constraint is required for 24-round collision for which we provide complete details. An example of a collision obtained using this method is given in Table 19 of Section A.

6.2 24-Round Collisions

The SS local collision is placed from Step $i = 10$ to Step $i + 8 = 18$, i.e. $(w, x, y, z) = (1, -1, -1, 0)$. The message differences are as given by (8) where we choose $u = 1$. The values of δ_1, δ_2 need to be suitably chosen and then the values of λ, γ and μ can be found by solving (10) as explained in Section 5.2. From λ, we find α as explained earlier.

Since the collision ends at Step 18 and $u = 1$, from (8) we have $\delta W_{17} = 1$ and $\delta W_{18} = -1$. To obtain a 24-round collision, we need to ensure $\delta W_{19} = \cdots = \delta W_{23} = 0$.

From Table 9, (8) and the fact that $\delta W_j = 0$ for $0 \leq j \leq 9$, we get that the conditions $\delta W_{19} = \delta W_{20} = 0$ translate into the conditions

$$\left.\begin{array}{l} \delta_1 = \delta W_{12} = -(\sigma_1(W_{17} + 1) - \sigma_1(W_{17})) \\ \delta_2 = \delta W_{13} = -(\sigma_1(W_{18} - 1) - \sigma_1(W_{18})). \end{array}\right\} \quad (13)$$

As in the case of 23-round collisions, based on the differential behaviour of σ_1 (described in Section 2.3), 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.

Consider Table 13. This table tells us what the values of the different a and e-registers need to be. The values of a_8 to a_{12} and the values of e_8 to e_{16} get defined. Using CDE, the values of e_{11} down to e_8 determine the values of a_7 down to a_4. Thus, the values of a_4 to a_{12} and e_{13} to e_{16} are fixed. So, the values of a_0 to a_3 are free. In particular, we see that $e_{16} = -1 - u = -2$. This can be achieved by setting W_{16} to

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

Since all 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. This is in addition to the requirement that the value of W_{17} and W_{18} satisfy (13).

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

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

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

$$\left.\begin{aligned} W_9 &= -W_1 + C_4 + f_{MAJ}(a_4, a_3, a_2) - \Phi_0 \\ W_{10} &= -W_2 + C_5 + f_{MAJ}(a_5, a_4, a_3) - \Phi_1 \\ W_{11} &= -W_3 + C_6 + f_{MAJ}(a_6, a_5, a_4) - \Phi_2 \end{aligned}\right\} \tag{16}$$

where

$$\left.\begin{aligned} C_i &= e_{i+5} - \Sigma_1(e_{i+4}) - f_{IF}(e_{i+4}, e_{i+3}, e_{i+2}) - 2a_{i+1} - K_{i+5} + \Sigma_0(a_i) \\ \Phi_i &= \Sigma_0(a_i) + f_{MAJ}(a_i, b_i, c_i) + \Sigma_1(e_i) + f_{IF}(e_i, f_i, g_i) + h_i + K_{i+1}. \end{aligned}\right\} \tag{17}$$

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

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

We need to ensure that W_{16} has the desired value given by (14) and that W_{17} and W_{18} take values which satisfy (13).

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

$$D = W_{16} - (\sigma_1(W_{14}) + C_4 + f_{MAJ}(a_4, a_3, a_2) - \Phi_0 + W_0). \tag{19}$$

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.

$$D = -W_1 + \sigma_0(W_1). \tag{20}$$

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 Section 6.3.

Solving (20) 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 (20) 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.

$$\left.\begin{array}{l} W_2 = a_2 - (\Sigma_0(a_1) + f_{MAJ}(a_1, b_1, c_1) + h_1 + \Sigma_1(e_1) + f_{IF}(e_1, f_1, g_1) + K_2) \\ W_3 = a_3 - (\Sigma_0(a_2) + f_{MAJ}(a_2, b_2, c_2) + h_2 + \Sigma_1(e_2) + f_{IF}(e_2, f_2, g_2) + K_3) \end{array}\right\} \qquad (21)$$

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 (17) and then D using (19).
5. Solve (20) 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 (17) and then W_2 using (21).
8. Run SHA-2 with W_2 to define $a_2, \ldots, h_2$.
9. Determine Φ_2 using (17) and then W_3 using (21).
10. Compute W_{17} and W_{18} using (18).
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-round collision. The actual construction of the collision is similar to the procedure for obtaining 22-round collisions described in Table 12; using the obtained values of $W_0, \ldots, W_3$ run SHA-2 for 4 steps to define the values of $(a_3, \ldots, h_3)$. Use Proposition 1 to set $W_4, \ldots, W_{12}$ to values so that $a_4, \ldots, a_{12}$ get the required values. Set W_{13}, W_{14}, W_{15} to ensure that e_{13}, e_{14}, e_{15} get the required values. Finally, set $W'_i = W_i + \delta W_i$ for $i = 0, \ldots, 15$. Then the message pairs $(W_0, \ldots, W_{15})$ and $(W'_0, \ldots, W'_{15})$ provide a 24-round 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-round SHA-2 computations. Since the entire process needs to be repeated $\frac{2^{2n}}{\mathsf{freq}_{\delta_1} \times \mathsf{freq}_{\delta_2}}$ times for obtaining success, the number of 24-round SHA-2 computations till success is obtained is about

$$\left(\frac{2^{2n}}{\mathsf{freq}_{\delta_1} \times \mathsf{freq}_{\delta_2}}\right) \times \left(2^{-4.5} \times \left(\frac{g}{2^4} + 3\right) \times \frac{1}{p}\right).$$

If (20) 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

$$\left(\frac{2^{2n}}{\mathsf{freq}_{\delta_1} \times \mathsf{freq}_{\delta_2}}\right) \times \left(2^{-4.5} \times 3 \times \frac{1}{p}\right)$$

24-round SHA-2 computations. The trade-off is that we need to use a look-up table having 2^n entries.

SHA-256. We choose $\delta_2 = \texttt{ff006001}$ with $\mathsf{freq}_{\delta_2} = 2^{16}$. Also, we choose $\delta_1 = \texttt{00006000}$ so that $-\delta_1 = \texttt{ffffa000}$ and $\mathsf{freq}_{-\delta_1} = 2^{29} + 2^{26}$. (See Table 2 in Section 2.3.) (For choices of δ_2 with higher value of freq_{δ_2} there are no solutions to the second equation of (10).)

For these values of δ_1 and δ_2, it is possible to solve (10) to obtain suitable λ, γ and μ, which in turn determine α. An example of these values is shown in Table 14 in the row $(24, 9)$. (The same values also hold for obtaining 23-round collision by placing a local collision from Step 9 to 17.)

The values of g, freq_{δ_1} and freq_{δ_2} are 2^{18}, 2^{29} and 2^{16} respectively. So, the time complexity is about $2^{28.5}$ 24-round SHA-256 computations. In our experiments, we found that the computation effort required to find $W_0, \ldots, W_3$ actually turns out to be less than the estimated effort of $2^{28.5}$ 24-round SHA-256 computations. The value of $2^{28.5}$ matches the figure given in [9], but [9] does not provide the detailed analysis of their cost. A message pair colliding for 24-round SHA-256 is given in Table 20 of Section A.

As already explained, if (20) is solved using a table look-up, then the cost reduces to about $2^{15.5}$ 24-round SHA-256 computations.

SHA-512. We choose $\delta_2 = \texttt{600000000237}$ with $\mathsf{freq}_{\delta_2} \approx 2^{43}$. Also, we choose $\delta_1 = \texttt{200000000008}$ so that $\mathsf{freq}_{-\delta_1} \approx 2^{61.5}$. See Table 3 in Section 2.3 For these values of δ_1 and δ_2, it is possible to solve (10) to obtain suitable λ, γ and μ, which in turn determine α. An example of these values is shown in the row marked $(24, 10)$ of Table 15.

The guess-then-determine attack for SHA-512 case requires $g = 2^{15}$ operations. Hence, the effort required for 24-round SHA-512 attack is about

$$\left(\frac{2^{2\times 64}}{2^{61.5} \times 2^{43}}\right) \times \left(2^{-4.5} \times \left(\frac{2^{15}}{2^4} + 3\right) \times \frac{1}{2^{-2.5}}\right) = 2^{32.5}$$

trials of 24-round SHA-512. In [9], the corresponding effort is 2^{53} trials of 24-round SHA-512. This significant improvement in the attack complexity allows us to provide the first example of a colliding message pair for 24-round SHA-512. A message pair colliding for 24-round SHA-512 is given in Table 22 of Section A.

Note that using a table having 2^{64} entries to solve (20) will reduce the computational effort to about $2^{22.5}$ trials of 24-round SHA-512.

6.3 Guess-Then-Determine Algorithm for Solving (20)

For the ease of notation, in this section we will use W instead of W_1.

For SHA-256. Consider Table 1 where the structure of W and $\sigma_0(W)$ is shown for SHA-256. We have $-W + \sigma_0(W) = D$, where $D = (d_{31}, \ldots, d_0)$ is a 32-bit constant. For $31 \geq k \geq l \geq 0$, we will use the notation $X[k, l]$ to denote bits $x_k, \ldots, x_l$ of the 32-bit quantity X.

We explain how the guess-then-determine algorithm proceeds. Suppose that we guess $W[14,0]$. Let $X = D + W$ and $Y = (W[14,0] \gg 3) \oplus (W[14,0] \gg 7)$. Then $W[25,18] = (X \oplus Y)\&(\texttt{ff})$. Having determined $W[25,18]$ we next determine $W[29,26]$ using positions 22 to 19 of Table 1. This time, however, there may have been a possible carry into the 19th bit and we need to account for that. Let c_0 be a bit. Define $X = (D \gg 19) + (W[25,18] \gg 1) + c_0$ and $Y = (W[14,0] \gg 5) \oplus (W[25,18] \gg 4)$. Then $W[29,26] = (X \oplus Y)\&(\texttt{f})$. This illustrates the general idea and can be extended to determine the other bits. Once the entire W has been determined we need to determine whether $-W + \sigma_0(W) = D$. The entire algorithm is shown in Figure 2. This algorithm involves guessing $W[14,0]$ and bits c_0, c_1, c_2, which is a total of 18 bits. If the equation $D = -W + \sigma_0(W)$ does not have any solution, then none will be returned by this algorithm; on the other hand, if there is a solution or there are more than one solutions, then all solutions will be returned. A total of 2^{18} operations are required. The time for each operation is significantly less than the time for a single SHA-256 step and by our assessment it is about 2^{-4} times the time for a single SHA-256 step.

Fig. 1. Structure of W and $\sigma_0(W)$ for SHA-256.

W	w_{31}	w_{30}	w_{29}	w_{28}	w_{27}	w_{26}	w_{25}	w_{24}	w_{23}	w_{22}	w_{21}	w_{20}	w_{19}	w_{18}	w_{17}	w_{16}
$W \gg 3$	0	0	0	w_{31}	w_{30}	w_{29}	w_{28}	w_{27}	w_{26}	w_{25}	w_{24}	w_{23}	w_{22}	w_{21}	w_{20}	w_{19}
$W \ggg 7$	w_6	w_5	w_4	w_3	w_2	w_1	w_0	w_{31}	w_{30}	w_{29}	w_{28}	w_{27}	w_{26}	w_{25}	w_{24}	w_{23}
$W \ggg 18$	w_{17}	w_{16}	w_{15}	w_{14}	w_{13}	w_{12}	w_{11}	w_{10}	w_9	w_8	w_7	w_6	w_5	w_4	w_3	w_2

W	w_{15}	w_{14}	w_{13}	w_{12}	w_{11}	w_{10}	w_9	w_8	w_7	w_6	w_5	w_4	w_3	w_2	w_1	w_0
$W \gg 3$	w_{18}	w_{17}	w_{16}	w_{15}	w_{14}	w_{13}	w_{12}	w_{11}	w_{10}	w_9	w_8	w_7	w_6	w_5	w_4	w_3
$W \ggg 7$	w_{22}	w_{21}	w_{20}	w_{19}	w_{18}	w_{17}	w_{16}	w_{15}	w_{14}	w_{13}	w_{12}	w_{11}	w_{10}	w_9	w_8	w_7
$W \ggg 18$	w_1	w_0	w_{31}	w_{30}	w_{29}	w_{28}	w_{27}	w_{26}	w_{25}	w_{24}	w_{23}	w_{22}	w_{21}	w_{20}	w_{19}	w_{18}

Fig. 2. A guess-then-determine algorithm for solving $D = -W + \sigma_0(W)$ for SHA-256.

```
1. Guess $W[14,0]$.
2. Let $X = D + W$ and $Y = (W[14,0] \gg 3) \oplus (W[14,0]) \gg 7$
   and set $W[25,18] = (X \oplus Y)\&(\texttt{ff})$.
3. Guess $c_0$.
4.     Let $X = (D \gg 19) + (W[25,18] \gg 1) + c_0$ and $Y = (W[14,0] \gg 5) \oplus (W[25,18] \gg 4)$
       and set $W[29,26] = (X \oplus Y)\&(\texttt{f})$.
5.     Guess $c_1$.
6.         Let $X = (D \gg 23) + (W[25,18] \gg 6) + c_1$ and $Y = (W[14,0] \gg 9) \oplus (W[29,26] \gg 4)$
           and set $W[31,20] = (X \oplus Y)\&(\texttt{3})$.
7.         Guess $c_2$.
8.             Let $X = (D \gg 8) + (W[14,0] \gg 8) + c_2$ and $Y = (W[14,0] \gg 11) \oplus (W[29,26])$
               and set $W[31,20] = (X \oplus Y)\&(\texttt{7})$.
9.             If $-W + \sigma_0(W) = D$, then output $W$ as one solution.
```

Note. In [9], it has been remarked that "*by guessing the least 15 bits of W_1 the entire W_1 can be reconstructed and with probability 2^{-14} it is going to be correct*". No details are provided. In particular, the guess-then-determine algorithm that we have described is not present in [9].

In our experiments with SHA-256, we found that for almost every other value of D, (20) has solutions, the number of solutions being one or two. So, for a random choice of D, we consider (20) to hold with probability $p \approx 1$.

For SHA-512. Consider Table 3 where the structure of W and $\sigma_0(W)$ is shown for SHA-512. We have $-W + \sigma_0(W) = D$, where $D = (d_{63}, \ldots, d_0)$ is a 64-bit constant. For $63 \geq k \geq l \geq 0$, we will use the notation $X[k, l]$ to denote bits $x_k, \ldots, x_l$ of the 64-bit quantity X.

We explain how the guess-then-determine attack proceeds. Suppose that we guess $W[7, 0]$. So we know the 7 bits $W[7, 1]$ and $W[6, 0]$. Now, consider the lowest 7 bits of $D+W$. We need $D+W$ to be equal to $\sigma_0(W)$. The term $\sigma_0(W)$ consists of 3 quantities XOR'ed, one of which, $W[7, 1]$, is already known. The other two quantities are $W[13, 7]$ and $W[14, 8]$. So we can compute $X = W[13, 7] \oplus W[14, 8] = (D + W) \oplus W[7, 1]$. Now, consider the least significant bit of X. This is the XOR of $W[7]$ and $W[8]$. We already know $W[7]$, so it is possible to compute $W[8]$. Once $W[8]$ is known, we can compute $W[9]$ by considering the second least significant bit of X. Continuing this way, we can get $W[14, 7]$.

Now consider the quantity $(D + W) \oplus (W \ggg 1)$ for bit positions 7 to 13. If the possible carry bit into the addition $D + W$ at bit position 7 can be guessed, then $W[21, 15]$ can be determined. Extending this reasoning further, we need to guess 7 carry bits and the initial 8 bits of W to completely determine W. If the obtained value of W satisifies $-W + \sigma_0(W) = D$, then we have the correct solution. The entire algorithm is shown in Figure 4.

In the algorithm, we use a function GTD, which takes low order $7i$ bits of W as input and produces low order $7i + 7$ bits of W. This function is described at the end of the figure.

This algorithm involves guessing $W[7, 0]$ and bits $c_1, c_2, \ldots c_7$, which is a total of 15 bits. If the equation $D = -W + \sigma_0(W)$ does not have any solution, then none will be returned by this algorithm; on the other hand, if there is a solution or there are more than one solutions, then all solutions will be returned. A total of 2^{15} operations are required. The time for each operation is significantly less than the time for a single SHA-512 step and by our assessment it is about 2^{-4} times the time for a single SHA-512 step.

7 Concluding Remarks

The method of attack described so far cannot be meaningfully extended beyond 24 steps as already mentioned in [9]. This is due to the fact that every extra step will introduce a new condition on the previous message words. The 24-round collision already utilized the freedom in the first message word W_0. To have a 25-round collision by starting the local collision at Step $i = 11$, will introduce impossibility in ensuring that the message word difference $\delta W_{16} = 0$. This is explained below.

As shown in Section 5.1, the local collision is $\{w, -w, \delta_1, \delta_2, 0, 0, 0, u, w\}$. If we start this local collision at Step $i = 11$, then $\delta W_{15} = \delta W_{16} = \delta W_{17} = 0$. Now from the message recursion of SHA-2, we have:

$$W_{16} = \sigma_1(W_{14}) + W_9 + \sigma_0(W_1) + W_0.$$

All the terms in the above equation, except W_{14}, are zero. Therefore this equation cannot be satisfied by this local collision. Similar reasons apply for longer round collisions.

Perhaps more fundamentally the problem is that, we are using only a single local collision. Since the local collision is nonlinear in nature, it is difficult to combine two or more such collisions. Further progress in analysis of step-reduced SHA-256 collisions will require some method to combined more than one (linear or non-linear) local collision.

References

1. Eli Biham and Rafi Chen. Near-Collisions of SHA-0. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 290–305. Springer, 2004.

Fig. 3. Structure of W and $\sigma_0(W)$ for SHA-512.

W	w_{63}	w_{62}	w_{61}	w_{60}	w_{59}	w_{58}	w_{57}	w_{56}	w_{55}	w_{54}	w_{53}	w_{52}	w_{51}	w_{50}	w_{49}	w_{48}
$W \gg 7$	0	0	0	0	0	0	0	w_{63}	w_{62}	w_{61}	w_{60}	w_{59}	w_{58}	w_{57}	w_{56}	w_{55}
$W \ggg 1$	w_0	w_{63}	w_{62}	w_{61}	w_{60}	w_{59}	w_{58}	w_{57}	w_{56}	w_{55}	w_{54}	w_{53}	w_{52}	w_{51}	w_{50}	w_{49}
$W \ggg 8$	w_7	w_6	w_5	w_4	w_3	w_2	w_1	w_0	w_{63}	w_{62}	w_{61}	w_{60}	w_{59}	w_{58}	w_{57}	w_{56}
W	w_{47}	w_{46}	w_{45}	w_{44}	w_{43}	w_{42}	w_{41}	w_{40}	w_{39}	w_{38}	w_{37}	w_{36}	w_{35}	w_{34}	w_{33}	w_{32}
$W \gg 7$	w_{54}	w_{53}	w_{52}	w_{51}	w_{50}	w_{49}	w_{48}	w_{47}	w_{46}	w_{45}	w_{44}	w_{43}	w_{42}	w_{41}	w_{40}	w_{39}
$W \ggg 1$	w_{48}	w_{47}	w_{46}	w_{45}	w_{44}	w_{43}	w_{42}	w_{41}	w_{40}	w_{39}	w_{38}	w_{37}	w_{36}	w_{35}	w_{34}	w_{33}
$W \ggg 8$	w_{55}	w_{54}	w_{53}	w_{52}	w_{51}	w_{50}	w_{49}	w_{48}	w_{47}	w_{46}	w_{45}	w_{44}	w_{43}	w_{42}	w_{41}	w_{40}
W	w_{31}	w_{30}	w_{29}	w_{28}	w_{27}	w_{26}	w_{25}	w_{24}	w_{23}	w_{22}	w_{21}	w_{20}	w_{19}	w_{18}	w_{17}	w_{16}
$W \gg 7$	w_{38}	w_{37}	w_{36}	w_{35}	w_{34}	w_{33}	w_{32}	w_{31}	w_{30}	w_{29}	w_{28}	w_{27}	w_{26}	w_{25}	w_{24}	w_{23}
$W \ggg 1$	w_{32}	w_{31}	w_{30}	w_{29}	w_{28}	w_{27}	w_{26}	w_{25}	w_{24}	w_{23}	w_{22}	w_{21}	w_{20}	w_{19}	w_{18}	w_{17}
$W \ggg 8$	w_{39}	w_{38}	w_{37}	w_{36}	w_{35}	w_{34}	w_{33}	w_{32}	w_{31}	w_{30}	w_{29}	w_{28}	w_{27}	w_{26}	w_{25}	w_{24}
W	w_{15}	w_{14}	w_{13}	w_{12}	w_{11}	w_{10}	w_9	w_8	w_7	w_6	w_5	w_4	w_3	w_2	w_1	w_0
$W \gg 7$	w_{22}	w_{21}	w_{20}	w_{19}	w_{18}	w_{17}	w_{16}	w_{15}	w_{14}	w_{13}	w_{12}	w_{11}	w_{10}	w_9	w_8	w_7
$W \ggg 1$	w_{16}	w_{15}	w_{14}	w_{13}	w_{12}	w_{11}	w_{10}	w_9	w_8	w_7	w_6	w_5	w_4	w_3	w_2	w_1
$W \ggg 8$	w_{23}	w_{22}	w_{21}	w_{20}	w_{19}	w_{18}	w_{17}	w_{16}	w_{15}	w_{14}	w_{13}	w_{12}	w_{11}	w_{10}	w_9	w_8

Fig. 4. A guess-then-determine algorithm for solving $D = -W + \sigma_0(W)$ for SHA-512.

```
Guess $W[7,0]$ and carry bits $c_1, c_2, c_3, c_4, c_5, c_6, c_7$.
Let $c_0 = 0$.
for($i = 0$; $i \le 7$; $i$++)
    $W[7i+7,0] = \text{GTD}(W[7i,7i-6],c_i)$;
If $-W + \sigma_0(W) = D$, then output $W$ as one solution.
```

```
function GTD($W[7i,7i-6],c_i$){
$X = (((D \gg (7i-7))\&(\texttt{7f})) + c_i + (W \gg (7i-7))\&(\texttt{7f})) \oplus ((W \gg (7i-6))\&(\texttt{7f}))$;
    $T_1 = (X\&\texttt{1}) \oplus ((W \gg (7i))\&\texttt{1})$;
    $T_2 = ((X \gg 1)\&\texttt{1}) \oplus T1$;
    $T_3 = ((X \gg 2)\&\texttt{1}) \oplus T2$;
    $T_4 = ((X \gg 3)\&\texttt{1}) \oplus T3$;
    $T_5 = ((X \gg 4)\&\texttt{1}) \oplus T4$;
    $T_6 = ((X \gg 5)\&\texttt{1}) \oplus T5$;
    $T_7 = ((X \gg 6)\&\texttt{1}) \oplus T6$;
temp = $T1 \oplus (T2 \ll 1) \oplus (T3 \ll 2) \oplus (T4 \ll 3) \oplus (T5 \ll 4) \oplus (T6 \ll 5) \oplus (T7 \ll 6)$;
$W[7i+7,0] = W[7i,7i-6] \oplus (temp \ll (7i+1))$;
Return $W[7i+7,0]$.
```

2. Gilles Brassard, editor. *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*. Springer, 1990.
3. 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.
4. 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*. Springer, 2005.
5. Ivan Damgård. A Design Principle for Hash Functions. In Brassard [2], pages 416–427.
6. Hans Dobbertin. Cryptanalysis of MD4. In Dieter Gollmann, editor, *Fast Software Encryption*, volume 1039 of *Lecture Notes in Computer Science*, pages 53–69. Springer, 1996.
7. Hans Dobbertin. Cryptanalysis of MD4. *Journal of Cryptology*, 11(4):253–271, 1998.
8. 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.
9. Sebastiaan Indesteege, Florian Mendel, Bart Preneel, and Christian Rechberger. Collisions and other Non-Random Properties for Step-Reduced SHA-256. In *Selected Areas in Cryptography, 15th Annual International Workshop, SAC 2008, Revised Papers*, 2008. To appear.
10. Sebastiaan Indesteege, Florian Mendel, Bart Preneel, and Christian Rechberger. Collisions and other Non-Random Properties for Step-Reduced SHA-256. *Cryptology eprint Archive*, April 2008. Available at `http://eprint.iacr.org/cgi-bin/versions.pl?entry=2008/131`, there are 7 versions dated 25 Mar, 27 Mar, 01 Apr, 08 Apr (2 versions), 14 Jul, 15 Jul, 2008.
11. Vlastimil Klima. Tunnels in Hash Functions: MD5 Collisions Within a Minute. Cryptology ePrint Archive, Report 2006/105, 2006. `http://eprint.iacr.org/2006/105`.
12. 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.
13. 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`.
14. Ralph C. Merkle. One Way Hash Functions and DES. In Brassard [2], pages 428–446.
15. Ivica Nikolić and Alex Biryukov. Collisions for Step-Reduced SHA-256. In Kaisa Nyberg, editor, *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, March 26-28, 2008*, volume 5086 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2008.
16. Somitra Kumar Sanadhya and Palash Sarkar. 22-step Collisions for SHA-2. arXiv e-print archive, arXiv:0803.1220v1, March 2008. Available at `http://de.arxiv.org/abs/0803.1220`, dated 08 Mar, 2008.
17. 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.
18. Somitra Kumar Sanadhya and Palash Sarkar. A Combinatorial Analysis of Recent Attacks on Step Reduced SHA-2 Family. Cryptology ePrint Archive, Report 2008/271, 2008. `http://eprint.iacr.org/2008/271`.
19. 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 5037 of *Lecture Notes in Computer Science*. Springer, 2008.
20. Somitra Kumar Sanadhya and Palash Sarkar. Deterministic Constructions of 21-Step Collisions for the SHA-2 Hash Family. In Editors, editor, *Information Security, 11th International Conference, ISC 2008, Taipei, Taiwan, September 2008, Proceedings*, volume 5222 of *Lecture Notes in Computer Science*, pages 244–259. Springer, 2008.
21. Somitra Kumar Sanadhya and Palash Sarkar. New Collision Attacks Against Up To 24-step SHA-2 . In D.R. Chowdhury, V. Rijmen, and A. Das, editors, *Progress in Cryptology - INDOCRYPT 2008, 9th International Conference on Cryptology in India*, volume 5365 of *Lecture Notes in Computer Science*, pages 91–103. Springer, 2008.
22. Somitra Kumar Sanadhya and Palash Sarkar. Non-Linear Reduced Round Attacks Against SHA-2 Hash family. In Yi Mu and Willy Susilo, editors, *Information Security and Privacy - ACISP 2008, The 13th Australasian Conference, Wollongong, Australia, 7-9 July 2008, Proceedings*, volume 5107 of *Lecture Notes in Computer Science*. Springer, 2008.
23. Somitra Kumar Sanadhya and Palash Sarkar. A New Hash Family Obtained by Modifying the SHA-2 Family. In *ACM Symposium on Information, Computer and Communications Security - ASIACCS 2009, Sydney, Australia*. ACM, March 2009. To appear. Full version available at `http://eprint.iacr.org/2008/272`.

24. 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`.
25. Marc Stevens, Arjen K. Lenstra, and Benne de Weger. Chosen-Prefix Collisions for MD5 and Colliding X.509 Certificates for Different Identities. In Moni Naor, editor, *EUROCRYPT*, volume 4515 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 2007.
26. Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, and Xiuyuan Yu. Cryptanalysis of the Hash Functions MD4 and RIPEMD. In Cramer [4], pages 1–18.
27. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu. Finding Collisions in the Full SHA-1. In 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*, pages 17–36. Springer, 2005.
28. Xiaoyun Wang and Hongbo Yu. How to Break MD5 and Other Hash Functions. In Cramer [4], pages 19–35.

A Colliding Message Pairs

Examples of colliding message pairs for 22, 23 and 24-round SHA-256 and SHA-512 using the standard IV are shown in Tables 16 to 22.

Table 16. Colliding message pair for 22-round SHA-512 with standard IV. These messages have been generated using the algorithm of Table 12.

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 17. Colliding message pair for 22-round SHA-256 with standard IV. These messages have been generated using the algorithm of Table 12.

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

B A Property of the NB Local Collision for SHA-512

The NB local collision has $(w, x, y, z) = (1, -1, 0, 0)$; $\delta W_i = 1$, $\delta W_{i+8} = -1$ and $\delta W_j = 0$, for $j = i+4, i+5, i+6, i+7$. Here $8 \le i \le 10$. The message word differences $\delta W_{i+1}, \delta W_{i+2}$ and δW_{i+3} are given by the following equations:

$$\left.\begin{aligned} \delta W_{i+1} &= -1 - \delta f_{IF}^{i}(1,0,0) - \delta \Sigma_1(e_i), \\ \delta W_{i+2} &= -\delta f_{IF}^{i+1}(-1,1,0) - \delta \Sigma_1(e_{i+1}), \\ \delta W_{i+3} &= -\delta f_{IF}^{i+2}(0,-1,1). \end{aligned}\right\} \qquad (22)$$

Table 18. Colliding message pair for 23-round 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 19. Colliding message pair for 23-round 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 20. Colliding message pair for 24-round 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 21. Colliding message pair for 23-round 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 22. Colliding message pair for 24-round 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

Suppose that the NB local collision is placed between Step i and Step $i+8$, $i = 8, 9, 10$; and it is desired to obtain a collision for $i + 14$ steps. Since the local collison ends at Step $i + 8$, from the differential path of the local collision in Table 4, we require the difference in the message word δW_{i+8} to be -1.

The basic idea is to ensure that the message word differences are all zero after the local collision ends. This will ensure that the two messages will not introduce any difference in the registers. Therefore, $\delta W_{i+9} = \delta W_{i+10} = \ldots = \delta W_{i+14} = 0$. From Table 9 it follows that we require $\delta\sigma_1(\delta W_{i+8}) + \delta W_{i+3} = 0$ to ensure that $\delta W_{i+10} = 0$.

We now show for SHA-512, it is difficult to find values of δW_{i+3} and $\delta\sigma_1(W_{i+8})$ which are of the same order of magnitude. The values of δW_{i+3} are biased towards small magnitudes. In contrast, the values of $\sigma_1(W_{i+8}) - \sigma_1(W_{i+8} - 1)$ for SHA-512 are biased towards large magnitudes. This makes it difficult to achieve equality of the two terms as required to ensure $\delta W_{i+10} = 0$.

In the discussion that follows, we use X_i to denote the i^{th} bit of a 64-bit quantity X. We also use the convention that the index of the least significant bit is 0.

Proposition 1 *$Pr[P_j \neq (P + 1)_j] = 1/2^j$, where the probability is taken over random P.*

Proof. The necessary and sufficient condition for the j^{th} bit of P and $P + 1$ to differ is that all the bits from 0 to $(j - 1)$ in P are 1. This happens with probability $1/2^j$, hence proved. □

Proposition 2 *If two numbers X and Y are such that $X_i \neq Y_i$ and $X_{i-1} = Y_{i-1}$, then $|X - Y| \geq 2^{i-1} + 1$.*

Proof. Without loss of generality, suppose $X_i = 1$ and $Y_i = 0$. Let $Z = X - Y$. If $Z_i = 1$, then clearly $Z \geq 2^i$ and we are done.

So, suppose $Z_i = 0$ and consider the process of binary subtraction of Y from X to obtain Z. Since $X_i = 1$ and $Y_i = 0$, the result $Z_i = 0$ can happen only if the subtraction of $Y_{i-1}Y_{i-2}\ldots Y_0$ from $X_{i-1}X_{i-2}\ldots X_0$ produces a carry. But since $X_{i-1} = Y_{i-1}$, this implies the following two things.
1. $Z_{i-1} = 1$.
2. The subtraction of $Y_{i-2}Y_{i-3}\ldots Y_0$ from $X_{i-2}X_{i-3}\ldots\ X_0$ produces a carry.

1. $Z_{i-1} = 1$.
2. The subtraction of $Y_{i-2}Y_{i-3}\ldots Y_0$ from $X_{i-2}X_{i-3}\ldots\ X_0$ produces a carry.

The second point implies that at least one bit of $Z_{i-2}Z_{i-3}\ldots\ Z_0$ must be 1. This together with the first point $Z_{i-1} = 1$ implies that $Z \geq 2^{i-1} + 1$. Hence proved. □

Next we prove that the probability that the absolute value of δW_{i+3}, in the NB local collision is larger than 2^j is bounded above by $1/2^{j-1}$.

Lemma 4 *If the NB local collision is started at Step i, then $Pr[|\delta W_{i+3}| \geq 2^j] < 1/2^{j-1}$.*

Proof. Since the local collision is started from step i, the message difference δW_{i+3} is given by Equation 22. This equation gives:

$$\begin{aligned}\delta W_{i+3} &= -\delta f_{IF}^{i+2}(0, -1, 1),\\ &= -f_{IF}(e_{i+2}, f_{i+2} - 1, g_{i+2} + 1) + f_{IF}(e_{i+2}, f_{i+2}, g_{i+2}),\\ &= -f_{IF}(e_{i+2}, e_{i+1} - 1, e_i + 1) + f_{IF}(e_{i+2}, e_{i+1}, e_i).\end{aligned}$$

The two f_{IF} terms in the computation above have the same first argument e_{i+2}. The second and the third arguments have a modular difference of ±1. If the j^{th} bit of e_{i+2} is 1 then the two f_{IF} functions will select the corresponding bit from the middle argument, else from the third argument.

Let $A = f_{IF}(e_{i+2}, e_{i+1} - 1, e_i + 1)$ and $B = f_{IF}(e_{i+2}, e_{i+1}, e_i)$. Further, let P_n be the event that $A_n \neq B_n$. The event $\delta W_{i+3} \geq 2^j$ can happen if and only if at least one of the bits $j, j+1, \ldots 63$ of δW_{i+3} is 1, i.e., if and only if at least one of the events $P_j, P_{j+1}, \ldots P_{63}$ holds.

Now we are ready to bound the probability of the required event. In the fourth step below, we use the fact that $f_{IF}(a, b, c) = b$ if $a = 1$ and $= c$ if $a = 0$.

$$\begin{aligned}
Pr[\delta W_{i+3} \geq 2^j] &= Pr[\bigcup_{i \geq j} P_i] \\
&\leq \sum_{i \geq j} Pr[P_i] \\
&= \sum_{i \geq j} (Pr[(e_{i+2})_i = 0] \cdot Pr[P_i | ((e_{i+2})_i = 0)] + Pr[(e_{i+2})_i = 1] \cdot Pr[P_i | ((e_{i+2})_i = 1)]) \\
&= \sum_{i \geq j} \left(\frac{1}{2} \cdot Pr[(e_i + 1)_i \neq e_i] + \frac{1}{2} \cdot Pr[(e_{i+1} - 1)_i \neq e_{i+1}] \right) \\
&= \frac{1}{2} \cdot \sum_{i \geq j} \left(\frac{1}{2^i} + \frac{1}{2^i} \right) \quad \text{(Using Proposition 1)} \\
&< \frac{1}{2^{j-1}}.
\end{aligned}$$

This proves the result. □

We now look at the distribution of values of $\sigma_1(W) - \sigma_1(W - 1)$ for random choices of W.

Lemma 5 *For the function σ_1 used in SHA-512,*

$$|\sigma_1(W) - \sigma_1(W - 1)| \geq (2^{42} + 2^{39} + 2^{38} + 2^{36} - 2^3),$$

where W is any 64-bit word.

Proof. The function σ_1 is defined for SHA-512 as:

$$\sigma_1(W) = ROTR^{19}(W) \oplus ROTR^{61}(W) \oplus SHR^6(W). \tag{23}$$

Let the 64-bit word W be specified as $(w_{63}, w_{62}, \ldots, w_1, w_0)$ where w_0 is the least significant bit of W. Then $\sigma_1(W)$ can be expressed as bit-wise XOR of three quantities having bit pattern shown below.

Bit Index	63 62 ... 58	57 ... 45	44 ... 0
$ROTR^{19}$	w_{18} w_{17} ... w_{13}	w_{12} ... w_0	w_{63} ... w_{19}
$ROTR^{61}$	w_{60} w_{59} ... w_{55}	w_{54} ... w_{42}	w_{41} ... w_{61}
SHR^6	0 0 ... 0	w_{63} ... w_{51}	w_{50} ... w_6

Let $W' = W - 1$. Then similar structure for $\sigma_1(W')$ can also be visualized. We are interested in the magnitude of $\sigma_1(W) - \sigma_1(W')$.

Let j be the least index such j^{th} bit of W is 1. That is, $w_j = 1$ and $w_i = 0$ for all $i \leq j - 1$. Then we have, $w_i \neq w'_i$ for $i \leq j$ and $w_i = w'_i$ for $i > j$. Now we consider two cases for j.

Case 1: $0 \leq j \leq 40$. In this case, we have that $w_i = w'_i$ for $i =$ 63, 51, 50, 42, 41 and $w_0 \neq w'_0$. From the structure of $\sigma_1(W)$ and $\sigma_1(W')$, we note that their 45^{th} bits will be unequal but their 44^{th} bits will be equal. Using Proposition 2 we get, $|\sigma_1(W) - \sigma_1(W')| \geq 2^{44} + 1$.

Case 2: $j \geq 41$. We need to consider the individual cases $j = 41, 42, \ldots 63$ here. Consider the case $j = 41$ first. In this case, we know the exact bit pattern in W and W' up to 41 bits. Only the high order bits from 42 to 63 are unknown in these two quantities. We also know that these high order bits are the same in W and W'. Since these are only 22 bits, we can exhaustively search this space and compute the value $|\sigma_1(W) - \sigma_1(W')|$ for the case $j = 41$. As j is increased, the same idea can be used with even smaller search space. The size of the complete search space is $1+2+2^2+\ldots+2^{22} = 2^{23}-1$. A C program running on an ordinary PC takes a fraction of a second to traverse this space.
Using exhaustive search, we found the minimum vale of $|\sigma_1(W) - \sigma_1(W-1)|$ to be `000004cffffffff8` which occurred for $j = 42$. This value is equal to $(2^{42} + 2^{39} + 2^{38} + 2^{36} - 2^3)$.

We have left one particular case of W undiscussed. This is the special case when all the bits in W are zero. In this case, we can compute the difference directly since $\sigma_1(0) = 0$ and $\sigma_1(-1) = 1 + 2 + \ldots + 2^{57}$. Thus, we have the difference $= 2^{58} - 1$.

Combining all the cases, the result is proved. □