Attacking Step Reduced SHA-2 Family in a Unified Framework

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

19^{th} June 2008

Abstract. In this work, we make a detailed analysis of local collisions and their applicability to obtain collisions for step reduced SHA-2 hash family. Our analysis explains previously reported collisions for up to 22-step SHA-2 hash functions with probability one.
We provide new and improved attacks against 23 and 24-step SHA-256 using a local collision given by Sanadhya and Sarkar (SS) at ACISP '08. The computational efforts for the 23-step and 24-step attacks are respectively $2^{12.5}$ and $2^{28.5}$ calls to the corresponding step reduced SHA-256. Using a look-up table having 2^{32} entries the computational effort for finding 24-step collisions can be reduced to $2^{14.5}$ calls. We exhibit colliding message pairs for both the 23 and 24-step attacks. The previous work on 23 and 24-step SHA-256 attacks is due to Indesteege et al. and utilizes the local collision presented by Nikolić and Biryukov (NB) at FSE '08. The reported computational efforts are 2^{18} and $2^{28.5}$ respectively. The previous 23 and 24-step attacks first constructed a pseudo-collision and later converted it into a collision for the reduced round SHA-256. We show that this two step procedure is unnecessary. Although these attacks improve upon the existing reduced round SHA-256 attacks, they do not threaten the security of the full SHA-2 family.

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

1 Introduction

Cryptanalysis of SHA-2 family has recently gained momentum due to the important work of Nikolić and Biryukov [5]. Prior work on finding collisions for step reduced SHA-256 was done in [3,4] and [8]. These earlier works used local collisions valid for the XOR linearized version of SHA-256 from [1] and [7]. On the other hand, the work [5] used a local collision which is valid for the actual SHA-256.

The authors in [5] developed techniques to handle nonlinear functions and the message expansion of SHA-2 to obtain collisions for up to 21-step SHA-256. The 21-step attack of [5] succeeded with probability 2^{-19}. Using similar techniques, but utilizing a different local collision, [10] showed an attack against 20-step SHA-2 which succeeds with probability one and an attack against 21-step SHA-256 which succeeds with probability 2^{-15}. Further work [9,6] developed collision attacks against 21 and 22 step SHA-2 family which succeed with probability one. Very recently, Indesteege et al. [2] have developed attacks against 23 and 24 step SHA-256. They utilize the local collision from [5] in these attacks.

Our contributions. We consider all 9-step local collisions using additive differentials in a general and unified framework. Using a combinatorial analysis, the local collisions from [5] and [10] are obtained as special cases. We show how the general analysis can be used to explain recently obtained collisions for up to 22-step SHA-2 with probability one.

Collisions for 23 and 24-step SHA-256 are described. Our analysis shows that both the NB and the SS local collisions can be used for this purpose. The 23-step collision using the SS local collision requires

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

$2^{12.5}$ calls to 23-step SHA-256. This improves upon the previously reported effort of 2^{18} calls obtained in [2] using the NB local collision.

The computational effort for the 24-step collisions is $2^{28.5}$ calls to 24-step SHA-256. While this equals the previously reported computational effort [2], we present complete details of our method including a guess-then-verify algorithm to solve a nonlinear equation arising in the analysis. This equation can also be solved using a table look-up. This brings down the computational cost of obtaining 24-step SHA-256 collisions to $2^{14.5}$ while requiring a look-up table having 2^{32} entries, where each entry consists of 8 bytes.

Examples of 23 and 24-step collisions are presented. For the case of 23-step collisions, two examples are presented. The first one uses a local collision from Steps 8 to 15, while the second one uses a local collision from Steps 9 to 16. The second 23-step collision can be seen as a simplified application of the technique for obtaining 24-step collisions.

The work in [2] describes 23 and 24-step collisions as a two-part procedure; first obtain a pseudo-collision and then convert it into a collision. In contrast, our analysis is direct and shows that such a two-part description is unnecessary. A summary of results on collision attacks against reduced SHA-2 family is given in Table 1.

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
			Prob.	Calls	utilized	
[3,4]	SHA-256	18		*	GH [1]	Linear
[8]	SHA-256	18		**	SS$_5$ [7]	”
[5]	SHA-256	20	$\frac{1}{3}$		NB [5]	Non-linear
		21	2^{-19}		”	”
[10]	SHA-256/SHA-512	18,20	1	1	SS [10]	”
	SHA-256	21	2^{-15}		”	”
[9]	SHA-256/SHA-512	21	1	1	”	”
[6]	SHA-256/SHA-512	22	1	1	”	”
[2]	SHA-256	23		2^{18}	NB [5]	”
		24		$2^{28.5}$	”	”
This work	SHA-256	23		$2^{12.5}$	SS [10]/NB [5]	”
		24		$2^{28.5}$	”	”
		24		$2^{14.5}$ †	”	”

* It is mentioned in [3,4] 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.

2 Preliminaries

In this paper we use the following notation:

- Message words: $W_i \in \{0,1\}^n$, $W'_i \in \{0,1\}^n$; n is 32 for SHA-256 and 64 for SHA-512.
- Colliding message pair: $\{W_0,\ W_1,\ W_2,\ \ldots W_{15}\}$ 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 Hash Family

Eight registers are used in the evaluation of SHA-2. In Step i, the 8 registers are updated from (a_{i-1}, b_{i-1}, c_{i-1}, d_{i-1}, e_{i-1}, f_{i-1}, g_{i-1}, h_{i-1}) to (a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i). For more details, see Section B.

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

Cross Dependence Equation (CDE).

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

Later, we make extensive use of this relation. Note that a special case of this equation was first utilized in Section 6.1 of [10]. The equation in the form above was used in [9] and [6]. 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 [2].

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 [5]. Given a word w, we define

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

The general differential path and corresponding message differences are shown in Table 2.

Table 2. General 9-step 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 2 is that it puts no restrictions on the actual message words $W_i, \ldots, W_{i+8}$ and $W'_i, \ldots, W'_{i+8}$. Starting at any value for the registers a to h, 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.

The expression for x has the term $\delta\Sigma^i_0(w)$. This map is invariant on 0 and -1. Consequently, by setting either ($w = 1$ and $a_i = -1$) or ($w = -1$ and $a_i = 0$), we can achieve $\delta\Sigma^i_0(w) = w$. The condition ($w = -1$ and $a_i = 0$) is the dual of the condition ($w = 1$ and $a_i = -1$) in the following sense. We have defined $\delta X = X' - X$ and so $\delta W_i = w$ means $W'_i = W + w$; if we had defined δX to be $X - X'$, then W'_i would have been $W - w$. Consequently, without loss of generality one can ignore the case $-w$. Majority function can be simplified by ensuring that two of its inputs are equal. This rule can be used to simplify the expressions of x, y and z to values such as 0 or -1.

The Nikolić-Biryukov (NB) local collision [5]. A special case of Table 2 is obtained by putting $(w, x, y, z) = (1, -1, 0, 0)$. This was the first reported local collision for SHA-2 using modular differentials [5]. We will call this the NB local collision.

The Sanadhya-Sarkar (SS) Differential Path [10]. The case from Table 2 obtained by putting $z = 0$ was studied in [10]. The special case of $(w, x, y, z) = (1, -1, -1, 0)$ turned out to be important in [10] and it is also important in the current work. We will call this the SS local collision.

After simplifying the values of x, y and z, it is possible to simplify the expressions for some of the δW_is. We are interested in obtaining conditions to ensure that some of the δW_is are zero. For example, the condition $z = 0$ and $e_{i+6} = -1$ ensures $\delta W_{i+7} = 0$; while the condition ($e_{i+5} = 0$ and $y = -z$) or ($e_{i+5} = -1$ and $y = -w$) ensures $\delta W_{i+6} = 0$. The conditions for achieving $\delta W_{i+4} = \delta W_{i+5} = 0$ are a little more complicated. The complete details for the simplification of $\delta\Sigma_0$; Majority and the δW_is are given in Section C.

3.1 Obtaining up to 22-Round Collisions

For obtaining collisions with more than 16 rounds, we need to consider the message expansion. The initial free words are $W_0, \ldots, W_{15}$ and from W_{16} onwards, the words are computed using the message expansion recursion given by (14). For clarity some initial words are shown in Table 3.

Table 3. Message expansion from W_{16} to W_{26}.

$W_{16} = \sigma_1(W_{14}) + W_9 + \sigma_0(W_1) + W_0$
$W_{17} = \sigma_1(W_{15}) + W_{10} + \sigma_0(W_2) + W_1$
$W_{18} = \sigma_1(W_{16}) + W_{11} + \sigma_0(W_3) + W_2$
$W_{19} = \sigma_1(W_{17}) + W_{12} + \sigma_0(W_4) + W_3$
$W_{20} = \sigma_1(W_{18}) + W_{13} + \sigma_0(W_5) + W_4$
$W_{21} = \sigma_1(W_{19}) + W_{14} + \sigma_0(W_6) + W_5$
$W_{22} = \sigma_1(W_{20}) + W_{15} + \sigma_0(W_7) + W_6$
$W_{23} = \sigma_1(W_{21}) + W_{16} + \sigma_0(W_8) + W_7$
$W_{24} = \sigma_1(W_{22}) + W_{17} + \sigma_0(W_9) + W_8$
$W_{25} = \sigma_1(W_{23}) + W_{18} + \sigma_0(W_{10}) + W_9$
$W_{26} = \sigma_1(W_{24}) + W_{19} + \sigma_0(W_{11}) + W_{10}$

The basic technique is to place a single local collision from steps i to $i+8$ for a suitably chosen i and then ensure that message expansion does not interfere with this collision. All δW_j with $0 \leq j \leq 15$ and $j \notin \{i, i+1, \ldots, i+8\}$ are set to 0. Doing this does not place any constraint on either $W_0, \ldots, W_{15}$ or on $W'_0, \ldots, W'_{15}$.

Additionally, some δW_k with $k \in \{i, i+1, \ldots, i+8\}$ are also set to 0. This requires setting some of the a_is and e_js to specific values. This is achieved by setting the message word for the corresponding round to a specific value. Note, however, that using a single message word we cannot set both a and e registers to desired values. Also, by the CDE (Equation 1), fixing a_{i-4} to a_i sets e_i to a fixed value.

18-Round Collisions [10]. Deterministic 18-round collisions are easy to obtain by setting $i = 3$ (i.e., the local collision spans from $i = 3$ to $i + 8 = 11$) and ensuring δW_{i+6} and δW_{i+7} are both zeros. 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.

20-Round Collisions [5,10]. Deterministic 20-round collisions can be obtained by setting $i = 5$ (i.e., the local collision spans from $i = 5$ to $i + 8 = 13$) and ensuring $\delta W_{i+4} = \delta W_{i+5} = \delta W_{i+6} = \delta W_{i+7} = 0$. Details of how this can be done is described in Section D.1.

21-Round Collisions [9]. Nikolić-Biryukov show how to find local collisions for SHA-256 with probability 2^{-19} using $(w, x, y, z) = (1, -1, 0, 0)$. This was improved to probability 2^{-15} using $(w, x, y, z) = (1, -1, -1, 0)$ in [10]. But, none of these works could provide 21-round SHA-512 collisions.

In a later work [9], deterministic 21-round collisions were shown for both SHA-256 and SHA-512 using $(w, x, y, z) = (1, -1, -1, 0)$. In [9], it was also shown that the NB local collision, i.e., $(w, x, y, z) = (1, -1, 0, 0)$ is unlikely to produce 21-step SHA-512 collisions. Some details of the deterministic 21-round collisions are given in Section D.2.

22-Round Collisions [6]. The technique of [9] was extended to obtain deterministic 22-round collisions in [6]. Some details of how this can be done is given in Section D.3. Again, it is unlikely that the NB local collision can be used to produce 22-step SHA-512 collisions [6].

4 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 farther) and hence the local collision will end at Step 16 (or farther). This will require us to analyze the message expansion more carefully.

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

In the rest of the paper, we will only consider SHA-256 and not SHA-512. The techniques depend to some extent on the nature of the expansion function σ_1 used in SHA-256. More importantly, we require to solve some nonlinear equations involving 32-bit quantities. The differential properties of σ_1 are summarized in Section B.1. For SHA-512, this will involve 64-bit quantities and hence solving such equations will be much more difficult.

4.1 A Class of Local Collisions

A local collision of the type shown in Table 2 is completely determined by the values of w, x, y and z and the values of δW_i to δW_{i+8}. We need to consider some special values for the δWs. Let

$$(\delta W_i, \ldots, \delta W_{i+8}) = (w, -w, \delta_1, \delta_2, 0, 0, 0, u, -w). \tag{3}$$

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 2, Equation 3 gives rise to the following 9 equations.

$$\begin{array}{lllll}
\text{(A)} & \delta W_i & = & & = w; \\
\text{(B)} & \delta W_{i+1} & = & x - \delta\Sigma_1^{i}(w) - \delta f_{IF}^{i}(w, 0, 0) & = -w; \\
\text{(C)} & \delta W_{i+2} & = & y - \delta\Sigma_1^{i+1}(x) - \delta f_{IF}^{i+1}(x, w, 0) & = \delta_1; \\
\text{(D)} & \delta W_{i+3} & = & z - \delta\Sigma_1^{i+2}(y) - \delta f_{IF}^{i+2}(y, x, w) & = \delta_2; \\
\text{(E)} & \delta W_{i+4} & = & -w - \delta\Sigma_1^{i+3}(z) - \delta f_{IF}^{i+3}(z, y, x) & = 0; \\
\text{(F)} & \delta W_{i+5} & = & -x - \delta\Sigma_1^{i+4}(w) - \delta f_{IF}^{i+4}(w, z, y) & = 0; \\
\text{(G)} & \delta W_{i+6} & = & -y - \delta f_{IF}^{i+4}(0, w, z) & = 0; \\
\text{(H)} & \delta W_{i+7} & = & -z - \delta f_{IF}^{i+4}(0, 0, w) & = u; \\
\text{(I)} & \delta W_{i+8} & = & & = -w.
\end{array}$$

The values of x, y and z from (2) are the following.

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

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 satisfy the values of δWs. 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 the following, we will only consider $z = 0$. (Equations arising from the case $z = -1$ are more complicated and require further analysis.) 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 4.

Table 4. Values of a and e register for the δWs given by (3) to hold. We have $w = 1$, $x = -1$ and $z = 0$. If $\beta = \alpha$, then $y = 0$, while if $\beta = \overline{\alpha}$, then $y = -1$. If $y = 0$, then $\lambda = \alpha - \Sigma_0(\alpha)$, while if $y = -1$, then $\lambda = \alpha + \overline{\alpha} + 1 - \Sigma_0(\overline{\alpha})$. The value of u is either 0 or w. By the 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 4 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 (B), we have

$$\delta W_{i+1} = x - \delta\Sigma_1^{i}(w) - \delta f_{IF}^{i}(w, 0, 0)$$

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

Similarly, Equations (F), (G) and (H) can be verified. Equations (C), (D) and (E) on the other hand give rise to the following conditions on the values of α, γ and μ.

$$\left.\begin{array}{l}
\delta_1 = y - \Sigma_1(\mu + x) + \Sigma_1(\mu) - f_{IF}(\mu + x, 0, \gamma + 1) + f_{IF}(\mu, -1, \gamma + 1)\\
\delta_2 = -\Sigma_1(\lambda + y) + \Sigma_1(\lambda) - f_{IF}(\lambda + y, \mu + x, 0) + f_{IF}(\lambda, \mu, -1)\\
w = -f_{IF}(\lambda + y, \lambda + y, \mu + x) + f_{IF}(\lambda + y, \lambda, \mu).
\end{array}\right\} \qquad (4)$$

The special case of these equations with $y = 0$ have been reported in [2] and a method for solving them has been discussed. For the case $y = -1$, the following strategy (which is somewhat similar to the case $y = 0$ given in [2]) can be used to solve the equations.

- The third equation has a solution if and only if both λ and μ are odd.
- Given that the third equation holds, the second equation simplifies to $\delta_2 = -\Sigma_1(\lambda - 1) + \Sigma_1(\lambda) + (\lambda - 1)$. For odd values of δ occurring in the distribution of $\sigma_1(W) - \sigma_1(W - 1)$, it is possible to solve this equation for odd λ. The maximum value of freq_δ for odd δ is 2^{16}. One example is $\delta =$ `ff006001` as shown in Table 10. (As mentioned earlier, if $\mathsf{freq}_\delta > 2^{16}$, then δ is even and for such values of δ, we could not find any odd λ satisfying the second equation.)
- Given such a λ, it is possible to invert the equation $\lambda = \alpha + \overline{\alpha} + 1 - \Sigma_0(\overline{\alpha})$ to obtain a suitable value of α.
- Now, we choose a γ and obtain an odd μ such that the first equation holds.

Solving all the three equations for α, γ and μ can be done in a few seconds on a current PC. Examples are provided in Table 5.

Table 5. Values leading to collisions for different number of rounds. Here $w = 1$, $x = y = -1$ and $z = 0$. 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`

5 23-Round Collisions

We show that by suitably placing a local collision of the type described in Section 4.1 and using proper values for α, γ and μ, it is possible to obtain several 23-round collisions for SHA-256.

5.1 Case $i = 8$

The local collision is started at $i = 8$ and ends at $i = 16$. We have $w = 1, x = -1, z = 0$ and we choose $y = -1$ by setting $\beta = \overline{\alpha}$. 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 $\delta = \delta_2$.

Since the local collision ends at Step 16, it necessarily follows that $\delta W_{16} = -1$. Consequently, we need to consider δW_{18} to ensure that it is zero. Since the collisions starts at $i = 8$, all δW_j for $0 \leq j \leq 7$ are zero. Consequently, we can write $\delta W_{18} = \delta\sigma_1(W_{16}) + \delta W_{11}$, where $\delta\sigma_1(W_{16}) = \sigma_1(W_{16} - 1) - \sigma_1(W_{16})$. So, for δW_{18} to be zero, we need $\delta W_{11} = -\delta\sigma_1(W_{16})$, so that δW_{11} should be one of the values which occur in the distribution of $\sigma_1(W) - \sigma_1(W-1)$ for some W. Table 5 shows an example of solution of (4) in this case.

Obtaining proper values for the constants only ensures that the local collision holds from Steps i to $i + 8$ as expected. It does not, however, guarantee that the reduced round collision holds. In the present case, we need to have δW_{18} to be zero. This will happen only if W_{16} takes a value such that $\sigma_1(W_{16} - 1) - \sigma_1(W_{16})$ is equal to $-\delta$. This can be ensured probabilistically in the following manner. The δ that we have used (shown in Table 5) is such that $\mathsf{freq}_\delta = 2^{16}$ and so by trying approximately 2^{16} possible random choices of W_0 and W_1 we expect a proper value for W_{16} and hence, a 23-round collision. This shows that it is possible to obtain a 23-round collision with probability 2^{-16}. In [2], the corresponding probability is 2^{-19}. So, our technique is an improvement.

Since $i = 8$, from Table 4, we see that a_6 to a_{10} get defined and e_6 to e_{14} get defined. Using CDE, the values of e_9 down to e_6 is set by fixing values of a_5 down to a_2. In other words, the values of a_2 to a_{10} are fixed. Now, consider

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

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

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

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

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

Again in the right hand side, all quantities other than a_1 have fixed values. So, we can write $W_9 = C - a_1$, where C is a fixed value. (This relation has already been observed in [2].)

Now,

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

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

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

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

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

There are 2^{16} values of W_{16} for which $\sigma(W_{16}-1)-\sigma(W_{16})$ equals δ. So, we have to solve this equation for W_0 and W_1 such that W_{16} is one of these 2^{16} possible values. The simplest way to do this is to try out random choices of W_0 and W_1 until W_{16} takes one of the desired values. On an average, success is obtained after 2^{16} trials. Each trial corresponds to about a single step of SHA-256 computation. So, the total cost of finding suitable W_0 and W_1 is about $2^{12.5}$ tries of 23-step SHA-256 computations.

For each such solution (W_0, W_1) and an arbitrary choice of W_{15} we obtain a 23-round collision for SHA-256. Note that after W_0 and W_1 has been obtained everything else is deterministic, i.e., no random tries are required. The task of obtaining a suitable W_0 and W_1 can be viewed as a pre-computation of the type required to find the values of α, γ and μ. Then, the actual task of finding collisions becomes deterministic. An example of a collision obtained using this method is given in Table 6.

5.2 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 deterministically obtain 23-step collisions for both $y = 0$ and $y = -1$. We do not provide these details, since essentially the same 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 7.

5.3 Relation to the 23-Round Collision from [2]

The NB local collision (i.e., $(w, x, y, z) = (1, -1, 0, 0)$) has been used in [2]. The local collision was placed from Step 9 to Step 17. It was remarked in [2] that they were not able to use the local collision with $(w, x, y, z) = (1, -1, -1, 0)$ (i.e., the SS local collision) to obtain 23-round collisions.

In comparison, we have shown that the SS local collision gives rise to two kinds of 23-round collision. The first one is obtained by placing the local collision from Steps 8 to 16, and the second one is obtained by placing the local collision from Steps 9 to 17. Examples of both kinds are given in the Appendix.

The description of the attack in [2] is quite complicated. First they consider a 23-round pseudo-collision which is next converted into 23-round collision. This two-step procedure is unnecessary. Our detailed combinatorial analysis allows us to directly describe the attacks. In fact, our analytical framework can also be used to explain how one may obtain a 23-round collision from the NB local collision by placing a local collision from Steps 9 to 17. We omit the details since a lot of it would be repetitive.

6 24-Round Collisions

Note. In this and the next section, we will be working with the SS local collision, i.e., with $(w, x, y, z) = (1, -1, -1, 0)$. (The same analysis also goes through with small changes for the NB local collision, i.e., with $(w, x, y, z) = (1, -1, 0, 0)$ which has been used in [2].)

The local collision described in Section 4.1 is placed from Step $i = 10$ to Step $i + 8 = 18$ with $w = 1$, $x = y = -1$, $z = 0$ and $u = 1$. The values of δ_1, δ_2 as well as suitable values of α, γ and μ need to be chosen.

Since, the collision ends at Step 18 and $u = 1$, we will have $\delta W_{17} = 1$ and $\delta W_{18} - 1$. As a result, to ensure $\delta W_{19} = \delta W_{20} = 0$, we need to have $\delta_1 = \delta W_{12} = -(\sigma_1(W_{17} + 1) - \sigma_1(W_{17}))$ and $\delta_2 = \delta W_{13} = -(\sigma_1(W_{18} - 1) - \sigma_1(W_{18}))$. Based on the differential behaviour of σ_1 described in Section B.1, we should try to choose δ_1 and δ_2 such that $\mathsf{freq}_{-\delta_1}$ and freq_{δ_2} are as high as possible. (Here $-\delta_1$ denotes $-\delta_1 \bmod 2^{32}$.) But, at the same time, the chosen δ_1 and δ_2 must be such that (4) are satisfied.

As mentioned earlier, if we choose δ_2 such that $\mathsf{freq}_{\delta_2} > 2^{16}$, then it is not possible to solve (4). So 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 $\mathsf{freq}_{-\delta_1} = 2^{29} + 2^{26}$. For these values of δ_1 and δ_2, it is possible to solve (4) to obtain suitable α, γ and μ, which in turn determine $\beta = \overline{\alpha}$ and λ. An example of these values is shown in Table 5. The same values also hold for obtaining 23-step collision by placing a local collision from Step 9 to 17.

Now we consider Table 4. This table tells us what the values of the different a and e-registers need to be. Since messages up to W_{15} are free, we can set values for a and e registers up to Step 15. But, we see that $e_{16} = -1 - u = -2$. This can be achieved by setting W_{16} to

$$W_{16} = e_{16} - \Sigma_1(e_{15}) - f_{IF}(e_{15}, e_{14}, e_{13}) - a_{12} - e_{12} - K_{16}. \tag{5}$$

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

Since $i = 10$, from Table 4, we see that a_8 to a_{12} have to be set to fixed values and e_8 to e_{16} have to be set to fixed values. Using CDE, the values of e_{11} down to e_8 are determined by a_7 to a_4. So, the values of a_0 to a_3 are free and correspondingly the choices of words W_0 to W_3 are free.

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

$$\left.\begin{aligned} W_{14} &= e_{14} - \Sigma_1(e_{13}) - f_{IF}(e_{13}, e_{12}, e_{11}) - a_{10} - e_{10} - K_{14} \\ W_{15} &= e_{15} - \Sigma_1(e_{14}) - f_{IF}(e_{14}, e_{13}, e_{12}) - a_{11} - e_{11} - K_{15}. \end{aligned}\right\} \tag{6}$$

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

$$\left.\begin{aligned} W_9 &= -W_1 + C_4 + f_{MAJ}(a_4, a_3, a_2) - \Phi_0 \\ W_{10} &= -W_2 + C_5 + f_{MAJ}(a_5, a_4, a_3) - \Phi_1 \\ W_{11} &= -W_3 + C_6 + f_{MAJ}(a_6, a_5, a_4) - \Phi_2 \end{aligned}\right\} \tag{7}$$

where

$$\left.\begin{aligned} C_i &= e_{i+5} - \Sigma_1(e_{i+4}) - f_{IF}(e_{i+4}, e_{i+3}, e_{i+2}) - 2a_{i+1} - K_{i+5} + \Sigma_0(a_i) \\ \Phi_i &= \Sigma_0(a_i) + f_{MAJ}(a_i, b_i, c_i) + \Sigma_1(e_i) + f_{IF}(e_i, f_i, g_i) + h_i + K_{i+1}. \end{aligned}\right\} \tag{8}$$

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

$$\left.\begin{aligned} W_{16} &= \sigma_1(W_{14}) + C_4 - W_1 + f_{MAJ}(a_4, a_3, a_2) - \Phi_0 + \sigma_0(W_1) + W_0 \\ W_{17} &= \sigma_1(W_{15}) + C_5 - W_2 + f_{MAJ}(a_5, a_4, a_3) - \Phi_1 + \sigma_0(W_2) + W_1 \\ W_{18} &= \sigma_1(W_{16}) + C_6 - W_3 + f_{MAJ}(a_6, a_5, a_4) - \Phi_2 + \sigma_0(W_3) + W_2. \end{aligned}\right\} \tag{9}$$

We need to ensure that W_{16} has the desired value given by (5) and that W_{17} and W_{18} take values which lead to desired values for $\delta\sigma_1(W_{17})$ and $\delta\sigma_1(W_{18})$ as explained above.

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

$$D = W_{16} - (\sigma_1(W_{14}) + C_4 + f_{MAJ}(a_4, a_3, a_2) - \Phi_0 + W_0). \tag{10}$$

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.

$$D = -W_1 + \sigma_0(W_1). \tag{11}$$

A guess-then-verify algorithm can be used to solve this equation. By guessing a total of 18 bits (15 least significant bits of W_1 and three other possible carry bits), it is possible to reconstruct the entire W_1 and then verify whether the reconstructed value is correct. Thus, by trying a total of 2^{18} combinations, it is possible to determine whether (11) has a solution and if so to find all possible solutions. The algorithm is given in Section E. (We note that in [2], 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-verify algorithm that we provide in Section E is not present in [2].)

In our experiments we found that for almost every other value of D, Equation (11) has solutions, the number of solutions being one or two. So, for a random choice of D, we consider (11) to hold with probability ≈ 1.

Solving (11) using table look-up. An alternative approach would be to use a pre-computed table. For each of the 2^{32} possible W_1s, 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 (11) can be found by a simple look-up into the table using D. The table would have 2^{32} 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.

$$\left.\begin{array}{l} W_2 = a_2 - (\Sigma_0(a_1) + f_{MAJ}(a_1, b_1, c_1) + h_1 + \Sigma_1(e_1) + f_{IF}(e_1, f_1, g_1) + K_2) \\ W_3 = a_3 - (\Sigma_0(a_2) + f_{MAJ}(a_2, b_2, c_2) + h_2 + \Sigma_1(e_2) + f_{IF}(e_2, f_2, g_2) + K_3) \end{array}\right\} \tag{12}$$

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-256 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 (8) and then D using (10).
5. Solve (11) for W_1 using the guess-then-verify algorithm in Section E.
6. Run SHA-256 with W_1 to define $a_1, \ldots, h_1$.
7. Determine Φ_1 using (8) and then W_2 using (12).
8. Run SHA-256 with W_2 to define $a_2, \ldots, h_2$.
9. Determine Φ_2 using (8) and then W_3 using (12).
10. Compute W_{17} and W_{18} using (9).
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. An example of a collision obtained using this method is given in Table 8.

Estimate of computation effort. Step 5 involves a computation of 2^{18} operations, where each operation is much faster than a single step of SHA-256; by our assessment the time for each operation is around 2^{-4} times the cost of a single step of SHA-256. Thus, the time for Step 5 is about 2^{14} single SHA-256 steps.

By the choice of δ_1, the equality $\sigma_1(W_{17}+1) - \sigma_1(W_{17}) = -\delta_1$ holds roughly with probability 2^{-3} while by the choice of δ_2 the equality $\sigma_1(W_{18}-1) - \sigma_1(W_{18}) = \delta_2$ holds roughly with probability 2^{-16} and we obtain success in Step 11 with roughly 2^{-19} probability. So, the entire procedure needs to be carried out around 2^{19} times to obtain a collision.

The time for executing the entire procedure once is about $(2^{14}+3)$ single SHA-256 steps which is about $2^{9.5}$ 24-step SHA-256 computations. Hence, success is obtained after about $2^{28.5}$ 24-step SHA-256 computations. In our experiments, we found that the computation effort required to find $W_0, \ldots, W_3$ actually turns out to less than the estimated effort of $2^{28.5}$ 24-step SHA-256 computations. The value of $2^{28.5}$ matches the figure given in [2]. But, [2] does not provide the detailed analysis of their cost.

If (11) 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-256 step so that the overall cost reduces to about $2^{14.5}$ 24-step SHA-256 computations. The trade-off is that we need to use a look-up table having 2^{32} entries.

6.1 More Number of Steps

The method of attack described so far cannot be meaningfully extended beyond 24 steps as already mentioned in [2]. This is due to the fact that every extra step will introduce a new condition on the previous message words. The 24-step collision already utilized the freedom in the first message word W_0. To have a 25-step 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 4.1, 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.

Note: The work [2] acknowledges the VIC computer cluster of K.U. Leuven for obtaining most of their experimental results. Not having access to such excellent computational resources, we used only a standard PC. This necessitated a detailed and careful analysis of the nonlinear equations and the computational effort to solve them.

Acknowledgements

We would like to thank Sebastiaan Indesteege for pointing out that the extended attacks beyond 24 steps utilizing a single local collision will not work. Such "attacks" were described in the previous version. Section 6.1 is based on Sebastiaan's comments.

References

1. 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.

2. 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/2008/131`.
3. 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.
4. 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`.
5. 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 Pre-proceedings version of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2008.
6. Somitra Kumar Sanadhya and Palash Sarkar. Collision attacks against 22-step SHA-512. Communicated.
7. Somitra Kumar Sanadhya and Palash Sarkar. New Local Collisions for the SHA-2 Hash Family. In Kil-Hyun Nam and Gwangsoo Rhee, editors, *Information Security and Cryptology - ICISC 2007, 10th International Conference, Seoul, Korea, November 29-30, 2007, Proceedings*, volume 4817 of *Lecture Notes in Computer Science*, pages 193–205. Springer, 2007.
8. Somitra Kumar Sanadhya and Palash Sarkar. Attacking Reduced Round SHA-256. In Steven Bellovin and Rosario Gennaro, editors, *Applied Cryptography and Network Security - ACNS 2008, 6th International Conference, New York, NY, June 03-06, 2008, Proceedings*, volume 5037 of *Lecture Notes in Computer Science*, pages 130–143. Springer, 2008.
9. Somitra Kumar Sanadhya and Palash Sarkar. Deterministic Constructions of 21-Step Collisions for the SHA-2 Hash Family. In Tzong-Chen Wu and Chin-Laung Lei, editors, *Information Security - ISC 2008, 11th International Conference, Taipei, Taiwan, September 15-18, 2008, Proceedings*, volume To appear of *Lecture Notes in Computer Science.* Springer, 2008.
10. 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 To appear of *Lecture Notes in Computer Science.* Springer, 2008.

A Colliding Message Pairs

Examples of colliding message pairs for 23-step and 24-step SHA-256 using the standard IV are shown in Tables 6, 7 and 8.

B Details of the SHA-2 Hash Family

Eight registers are used in the evaluation of SHA-2. The initial value in the registers is specified by an $8 \times n$ bit IV, n=32 for SHA-256 and $n = 64$ for SHA-512. In Step i, the 8 registers are updated from (a_{i-1}, b_{i-1}, c_{i-1}, d_{i-1}, e_{i-1}, f_{i-1}, g_{i-1}, h_{i-1}) to (a_i, b_i, c_i, d_i, e_i, f_i, g_i, h_i) according to the following Equations:

$$\left.\begin{aligned}
a_i &= \Sigma_0(a_{i-1}) + f_{MAJ}(a_{i-1}, b_{i-1}, c_{i-1}) + \Sigma_1(e_{i-1}) \\
&\quad + f_{IF}(e_{i-1}, f_{i-1}, g_{i-1}) + h_{i-1} + K_i + W_i \\
b_i &= a_{i-1} \\
c_i &= b_{i-1} \\
d_i &= c_{i-1} \\
e_i &= d_{i-1} + \Sigma_1(e_{i-1}) + f_{IF}(e_{i-1}, f_{i-1}, g_{i-1}) \\
&\quad + h_{i-1} + K_i + W_i \\
f_i &= e_{i-1} \\
g_i &= f_{i-1} \\
h_i &= g_{i-1}
\end{aligned}\right\} \tag{13}$$

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

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

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

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

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

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

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:

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

For a t-bit words α, β, γ and integer i, we use the short-hands given in Table 9.

Table 9. Some useful short-hands.

$\begin{aligned} \delta\Sigma_1^i(\alpha) &= \Sigma_1(e_i + \alpha) - \Sigma_1(e_i) \\ &= \Sigma_1(e_i') - \Sigma_1(e_i). \\ \delta\Sigma_0^i(\alpha) &= \Sigma_0(a_i + \alpha) - \Sigma_0(a_i) \\ &= \Sigma_0(a_i') - \Sigma_0(a_i). \end{aligned}$	$\begin{aligned} \delta f_{IF}^i(\alpha,\beta,\gamma) &= f_{IF}(e_i + \alpha, f_i + \beta, g_i + \gamma) - f_{IF}(e_i, f_i, g_i) \\ &= f_{IF}(e_i', f_i', g_i') - f_{IF}(e_i, f_i, g_i). \\ \delta f_{MAJ}^i(\alpha,\beta,\gamma) &= f_{MAJ}(a_i + \alpha, b_i + \beta, c_i + \gamma) - f_{IF}(a_i, b_i, c_i) \\ &= f_{MAJ}(a_i', b_i', c_i') - f_{IF}(a_i, b_i, c_i). \end{aligned}$

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

$$W_i = \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16} \tag{14}$$

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:

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

B.1 Differential Properties of σ_1

The linear function σ_1 of SHA-256 used in the message expansion has very poor differential properties with respect to modular addition. Consider the distribution of $\delta = \sigma_1(W) - \sigma_1(W-1)$ as W ranges over all 2^{32} values. It has already been observed in [2] 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 10. Interestingly, we have observed that if freq_δ is greater than 2^{16}, then δ is always even.

Table 10. Some examples of high frequency values of $\delta = \sigma_1(W) - \sigma_1(W-1)$.

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}

C 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, as was done by Nikolić-Biryukov [5]. This is done using several rules which are actually sufficient conditions. The rules and their consequences are described below.

C.1 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 operator 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 either ($w = 1$ and $a_i = -1$) or ($w = -1$ and $a_i = 0$).

C.2 Simplifying Majority

The output of $f_{MAJ}(a, b, c)$ can be predicted with probability one if two of the inputs are equal. Based on this, we make 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 11. The related sufficient conditions are given in Table 12, where we consider only the case $w = 1, a_i = -1$, since the other case $w = -1, a_i = 0$ is the dual of the first one.

Table 11. 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)
$(w, -2w, 0, 0)$	$(w, -2w, 0, -w)$	$(w, -2w, -w, 0)$	$(w, -2w, -w, -w)$

C.3 Simplifying δW_{i+4} to δW_{i+7}

The expression for δW_{i+4} involves $\delta\Sigma_1^{i+3}(z)$ and $\delta f_{IF}^{i+3}(z, y, x)$. By imposing certain conditions, it is possible to simplify both these expressions.

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. If $z = 0$, then e_{i+3} can be either 0 or -1. 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 is summarized in Table 13. In particular, the simplifying conditions and the resulting values of the respective δWs are shown.

Table 12. 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	$2 + a_{i-3}$

Table 13. Summary of simplifying conditions for δW_{i+4} to δW_{i+7}. The simplifications for δW_{i+4} and δW_{i+5} require Rules 1 and 2, whereas the simplifications for δW_{i+6} and δW_{i+7} do not require these rules.

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}	$w = 1$, $e_{i+4} = -1$	$-w - x - y + e_{i+3} - e_{i+2}$
δW_{i+6}	$e_{i+5} = 0$	$-y - z$
	$e_{i+5} = -1$	$-y - w$
δW_{i+7}	$e_{i+6} = 0$	$-w - z$
	$e_{i+6} = -1$	$-z$

D Details of up to 22-Round Collisions

D.1 20-Round Collisions [5,10]

Deterministic 20-round collisions can be obtained by setting $i = 5$ (i.e., the local collision spans from $i = 5$ to $i+8 = 13$) and ensuring $\delta W_{i+4} = \delta W_{i+5} = \delta W_{i+6} = \delta W_{i+7} = 0$. The conditions for individually setting any of these to 0 are given in Table 13.

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 14. In particular, we see that $z = 0$ in all cases.

Table 14. 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)

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 15.

Table 15. Conditions for setting $\delta W_{i+4} = \delta W_{i+5} = \delta W_{i+6} = \delta W_{i+7} = 0$.

A row of Table 14
AND
($e_{i+5} = 0$ and $y = -z$) or ($e_{i+5} = -1$ and $y = -w$)
AND
$e_{i+6} = -1$.

Note. Tables 14 and 15 show that it is possible to deterministically set all the four δWs to zero using the Nikolić-Biryukov local collision. Consequently, it is possible to obtain *deterministic* 20-round collision using this local collision. This was not done in [5] but mentioned in [10] later.

D.2 21-Round Collisions [9]

We set $i = 6$, i.e., the local collision spans from $i = 6$ to $i + 8 = 14$. 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 15.

Let $\delta\sigma_1(\delta W_i)$ denote $\sigma_1(W_i + \delta W_i) - \sigma_1(W_i)$. We have $\delta W_{14} = \delta W_{i+8} = -w$ and so

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

We now consider $\delta W_9 = W_{i+3}$ which by the differential path 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 14 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 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 $i = 6$, rows of Tables 12 corresponding to rows A and D of Table 14 ensure that a_4, a_5, a_6, a_7, a_8 and e_8 are all fixed to particular values. Due to 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

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

Choose W_2 and W_3 to set a_2 and a_3 such that $e_7 - e_6 = -\delta$. This ensures that $\delta W_{16} = 0$ and hence, provides a deterministic 21-round collision.

It is possible to obtain deterministic 21-round collision by placing the SS 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 14 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,

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

To simplify $\delta W_{10} = \delta W_{i+3}$ we choose rows from Table 14 such that both e_{i+2} and $e_{i+2} + y$ are either 0 or -1. These are rows A and D. 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.

D.3 22-Round Collisions [6]

Set $i = 7$ so that the local collision spans from $i = 7$ to $i+8 = 15$. Use sufficient conditions from Table 15 to ensure that $\delta W_{i+4} = \delta W_{i+5} = \delta W_{i+6} = \delta W_{i+7} = 0$. This ensures that $\delta W_j = 0$ for $j = 18, 19, 20, 21$ provided $\delta W_{16} = \delta W_{17} = 0$. To ensure $\delta W_{16} = 0$, we need to set $\delta W_9 = \delta W_{i+2} = 0$ and $\delta W_{17} = 0$ if we can ensure $\delta\sigma_1(\delta W_{15}) + \delta W_{10} = 0$.

So, apart from the conditions required to set $\delta W_{i+4} = \delta W_{i+5} = \delta W_{i+6} = \delta W_{i+7} = 0$, we need sufficient conditions to set $\delta W_9 = \delta W_{i+2} = 0$ and to set $\delta\sigma_1(\delta W_{15}) + \delta W_{10} = 0$.

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 14. For reasons similar to the case of 21-step attack, only row D can be utilized for the attack. The rest of the strategy is similar to the previously described collisions.

Choose an arbitrary value for W_{15} and set $\delta_1 = \delta\sigma_1(\delta W_{15})$ where $\delta W_{15} = -w$. Then use W_4 to set a_4 such that due to CDE, e_8 gets set to a particular value required to ensure that $\delta W_{10} = -\delta_1$. Similarly, use W_3 to set a_3 such that due to CDE, e_7 gets set to a particular value required to ensure that $\delta W_9 = 0$. Both of these can be done deterministically, giving rise to deterministic 22-round collisions.

E Guess-Then-Verify Algorithm for Solving (11)

For the ease of notation, in this section we will use W instead of W_1. Consider Table 1 where the structure of W and $\sigma_0(W)$ is shown. 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-verify attack proceeds. Suppose that we guess $W[14, 0]$. Let $X = D + W$ and $Y = (W[14, 0] \ggg 3) \oplus (W[14, 0] \ggg 7)$. 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 $X = (D \ggg 19) + (W[25, 18] \ggg 1) + c_0$ and $Y = (W[14, 0] \ggg 5) \oplus (W[25, 18] \ggg 4)$. 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 verify 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 $\sigma_0(W)$.

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-verify algorithm for solving $D = -W + \sigma_0(W)$.

```
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 $W[25,18] = (X \oplus Y)\&(\mathtt{ff})$.
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 $W[29,26] = (X \oplus Y)\&(\mathtt{f})$.
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 $W[31,20] = (X \oplus Y)\&(\mathtt{3})$.
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 $W[31,20] = (X \oplus Y)\&(\mathtt{7})$.
9.             If $-W + \sigma_0(W) = D$, then output $W$ as one solution.
```