Finding SHA-2 Characteristics: Searching Through a Minefield of Contradictions

Florian Mendel, Tomislav Nad, and Martin Schläffer

IAIK, Graz University of Technology, Austria.
tomislav.nad@iaik.tugraz.at

Abstract. In this paper, we analyze the collision resistance of SHA-2 and provide the first results since the beginning of the NIST SHA-3 competition. We extend the previously best known semi-free-start collisions on SHA-256 from 24 to 32 (out of 64) steps and show a collision attack for 27 steps. All our attacks are practical and verified by colliding message pairs. We present the first automated tool for finding complex differential characteristics in SHA-2 and show that the techniques on SHA-1 cannot directly be applied to SHA-2. Due to the more complex structure of SHA-2 several new problems arise. Most importantly, a large amount of contradicting conditions occur which render most differential characteristics impossible. We show how to overcome these difficulties by including the search for conforming message pairs in the search for differential characteristics.

Keywords: hash functions, SHA-2, collision attack, differential characteristic, generalized conditions

1 Introduction

Since the breakthrough results of Wang et al. [19, 20], hash functions have been the target in many cryptanalytic attacks. These attack have especially shown that several well-known and commonly used algorithms such as MD5 and SHA-1 can no longer be considered to be secure. In fact, practical collisions have been shown for MD5 and collisions for SHA-1 can be constructed with a complexity of about 2^{63} [18]. For this reason, NIST has proposed the transition from SHA-1 to the SHA-2 family as a first solution. As a consequence, more and more companies and organizations are migrating to SHA-2. Hence, a detailed analysis of this hash function family is needed to get a good view on its security.

Although the design principles of SHA-2 are very similar to SHA-1, it is still unknown whether or how the attacks on MD5 and SHA-1 can be extended to SHA-2. Since 2008, no collision attacks have been published on SHA-2. One reason might be that the SHA-3 competition [9] initiated by NIST has attracted more attention by the cryptographic community. However, a more likely reason is the increased difficulty of extending previous collision attacks to more steps of SHA-2. In this work, we show that apart from a good attack strategy, advanced automated tools are essential to construct differential characteristics and to find confirming message pairs.

D.H. Lee and X. Wang (Eds.): ASIACRYPT 2011, LNCS 7073, pp. 288–307, 2011.
The original publication is available at http://www.springerlink.com/content/r5164u1408rg3870/

Related Work. In the past, several attempts have been made to apply the techniques known from the analysis of SHA-1 to SHA-2. The first known cryptanalysis of the SHA-2 family was published by Gilbert and Handschuh [3]. They have shown 9-step local collisions which hold with a probability of 2^{-66}. Hawkes et al. [6] have improved these results to get local collisions with a probability of 2^{-39} by considering modular differences.

In [8], Mendel et al. have analyzed how collision attacks can be applied to step reduced SHA-256. They have shown that the properties of the message expansion of SHA-256 prevent an efficient extension of the techniques of Chabaud and Joux [1] and Wang et al. [19]. Nevertheless, they presented a collision for 18 steps of SHA-256. In [12], Sanadhya and Sarkar have revisited the problem of obtaining a local collision for the SHA-2 family, and in [13] they have shown how to use one of these local collisions to construct another 18-step collision for SHA-256.

Finally, Nikolić and Biryukov [11] found a 9-step differential using modular differences which can be used to construct a practical collision for 21 steps and a semi-free-start collision for 23 steps of SHA-256. This was later extended to 22, 23 and 24 steps by Sanadhya and Sarkar in a series of papers [14–16]. The best known collision attack on SHA-256 so far was for 24 steps and has been found by Sanadhya and Sarkar [15], and Indesteege et al. [7].

All these results use rather simple differential characteristics which are constructed mostly manually or using basic cryptanalytic tools. However, the most efficient collision attacks on SHA-1 use more complex characteristics, especially in the first few steps of the attack. Constructing such complex characteristics is in general a difficult task. First, Wang et al. [19] have constructed such a characteristic for SHA-1 manually. Later, De Cannière and Rechberger [2] proposed a method to efficiently find such complex characteristics for SHA-1 in an automated way. Furthermore, also the best practical collision attack on SHA-1 (with the highest number of steps) is based on this approach [4].

Our Contribution. Currently, all collision attacks on SHA-2 are of practical complexity and based on the same basic idea: extending a local collision over 9 steps to more steps. As already mentioned in [7], this kind of attack is unlikely to be extended beyond 24 steps. In this work, we investigate new ideas to progress in the cryptanalysis of SHA-2. First, we extend the idea of finding local collisions to more than 9 steps by exploiting the nonlinearity of both the state update and message expansion.

To find such local collisions an automated tool to search for complex differential characteristics is needed. We start with the approach of De Cannière and Rechberger [2] on SHA-1. Unfortunately, their techniques cannot directly be applied to SHA-2. We have observed several problems in finding valid differential characteristics for SHA-2. In this work, we have identified these problems and show how to solve them efficiently. Most importantly, a very high number of contradicting conditions occurs which render most differential characteristics impossible.

To summarize, we present the first automatic tool to construct complex differential characteristics for reduced SHA-2. Applying our tool to SHA-256 results in practical examples of semi-free-start collisions for 32 and collisions for 27 out of 64 steps of SHA-256. The best semi-free-start collision and collision attack so far was on 24 steps of SHA-256.

Outline. The paper is structured as follows. In Section 2 we give a short description of SHA-256. In Section 3, we provide an overview of the general attack strategy and briefly mention which problems arise in the search for differential characteristics in SHA-2. In Section 4, we show how to efficiently propagate differences and conditions in SHA-2. Furthermore, we discuss why most differential characteristics are invalid and describe how to detect inconsistencies. In Section 5 we present our automated tool to construct complex differential characteristics and to find conforming message pairs in SHA-2. Finally, we conclude on our results in Section 6.

2 Description of SHA-256

SHA-256 is one of four hash functions defined in the Federal Information Processing Standard (FIPS-180-3) [10]. All four hash functions were designed by the National Security Agency (NSA) and issued by NIST in 2002. SHA-256 is an iterated cryptographic hash function with a hash output size of 256 bits, a message block size of 512 bits and using a word size of 32 bits. In the compression function of SHA-2, a state of eight chaining variables $A,\ldots,H$ is updated using 16 message words $M_0,\ldots,M_{15}$.

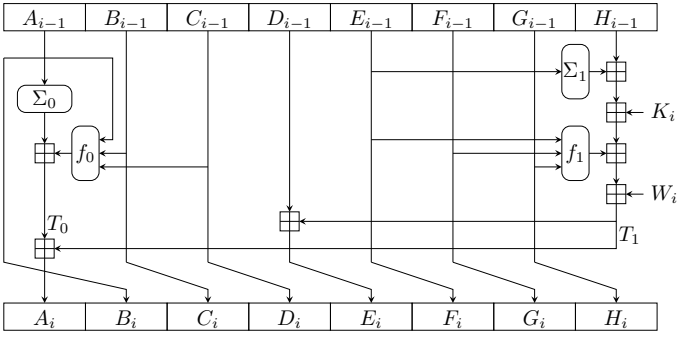

Fig. 1. The SHA-2 step update function.

The compression function of SHA-256 consists of 64 identical step update functions which are illustrated in Fig.1 and given as follows:

$$\begin{aligned} T_0 &= \Sigma_0(A_{i-1}) + f_0(A_{i-1}, B_{i-1}, C_{i-1}) \\ T_1 &= \Sigma_1(E_{i-1}) + f_1(E_{i-1}, F_{i-1}, G_{i-1}) + H_{i-1} + K_i + W_i \\ A_i &= T_0 + T_1\,, \;\; B_i = A_{i-1}\,, \;\; C_i = B_{i-1}\,, \;\; D_i = C_{i-1} \\ E_i &= D_{i-1} + T_1 \;\; F_i = E_{i-1}\,, \;\; G_i = F_{i-1}\,, \;\; H_i = G_{i-1} \end{aligned} \tag{1}$$

The Boolean functions f_0 (MAJ) and f_1 (IF) are given by

$$\begin{aligned} f_0(x, y, z) &= (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)\,, \\ f_1(x, y, z) &= (x \wedge y) \oplus (\neg x \wedge z)\,. \end{aligned}$$

The two $GF(2)$-linear functions Σ_0 and Σ_1 are defined as follows:

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

In the i-th step of the update function, a fixed constant K_i and the i-th word W_i of the expanded message are added to the state. The message expansion takes the 16 message words M_i as input and outputs 64 expanded message words W_i as follows:

$$W_i = \begin{cases} M_i & \text{for } 0 \leq i < 16 \\ \sigma_1(W_{i-2}) + W_{i-7} + \sigma_0(W_{i-15}) + W_{i-16} & \text{for } 16 \leq i < 64 \end{cases}$$

where the functions $\sigma_0(x)$ and $\sigma_1(x)$ are defined as follows:

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

3 Basic Attack Strategy

In this section, we give a brief overview of our attack strategy. We first describe how we generalize the approach of Nikolić and Biryukov [11] to find semi-free-start collisions on a higher number of steps. Due to this extension, differential characteristics cannot be constructed manually or semi-automatic anymore. Hence, we provide a fully automated tool to construct complex differential characteristics in SHA-2. Furthermore, we discuss why it is extremely difficult to find valid differential characteristics in SHA-2. In fact, we were not able to find a valid differential characteristic without including the search for a confirming message pair in the process. Therefore, the approach of first finding a valid differential characteristic and then, independently search for a conforming message pair does not apply very well to SHA-2. Hence, our attack strategy can be summarized as follows:

1. Determine a starting point for the search which results in an attack on a large number of steps. The resulting start characteristic should span over few steps and only some message words should contain differences.
2. Use an automated search tool to find a differential characteristic for the unrestricted intermediate steps including the message expansion.
3. Continue the search to find a conforming message pair. If no message pair can be found, adjust the differential characteristic accordingly.

Note that after step 2 it is not ensured that the differential characteristic is valid. If we cannot find a conforming message pair after a certain amount of time we go back to step 2 to adjust the differential characteristic.

3.1 Determining a Starting Point

By exploiting the nonlinearity of the step update function, Nikolić and Biryukov [11] found a 9-step differential characteristic for which it is not necessary to apply corrections (differences in the message words) in each step of the differential characteristic. The fact that not all (only 5 out of 9) message words contain differences helped to overcome several steps of the message expansion resulting in a collision and semi-free-start collision attack for 21 and 23 steps, respectively. Later this approach was extended to a collision attack on 24 steps [7, 15]. However, as pointed out in [7] it is unlikely that this approach can be extended beyond 24 steps.

In our attack, we are using differential characteristics which span over $t \geq 9$ steps, which allows us to attack more steps of SHA-256. As in the attack of Nikolić and Biryukov we are interested in differential characteristics with differences in only a few message words. Then, large parts of the expanded message have no difference which in turn, results in an attack on more than 24 steps. Already by using a differential characteristic spanning over $t = 10$ steps (with differences in only 3 message words) we can construct a semi-free-start collision for 27 steps of SHA-256. This can be extended to 32 steps using a differential characteristic spanning over $t = 16$ steps with differences in 8 message words.

To construct these starting points, we first fix the value of t and consider only differential characteristics which may result in collisions on more than 24 steps. Then, we identify those message words which need to have differences such that the differential characteristic holds for the whole message expansion. Table 2 in Appendix A shows the used starting point for the attack on 32 steps. Note that we have further optimized the message difference slightly to keep it sparse, which reduces the search space for the automated tool.

3.2 Searching for Valid Differential Characteristics and Conforming Message Pairs in SHA-2

Once we have determined a good starting point we continue by constructing a valid differential characteristic for both the state update transformation and the

message expansion. We have implemented an automated search tool for SHA-2 which is similar to the one proposed in [2] to construct complex characteristics for SHA-1. However, the increased complexity of SHA-2 compared to SHA-1 complicates a direct application of their approach. In the following, we briefly outline which problems occurred and how we have resolved them.

First of all, the larger state size, the combined update of two state variables, and the higher diffusion due to the Σ_i functions increases the complexity significantly. To limit these issues, we use an alternative description of SHA-2 where only two state variables are updated separately (see Section 4.1). Furthermore, we split up one SHA-2 step (including the nonlinear message expansion) into 9 less complex sub steps. This way, the propagation of differences can be implemented much more efficiently while losing only a small amount of information (see Section 4.3).

However, the main problem in SHA-2 is that it is difficult to determine whether a differential characteristic is valid, i.e. whether a conforming message pair exists. For example, a lot more conditions on two bits of the form $A_{i,j} = A_{i-1,j}$ occur in SHA-2, compared to SHA-1 for example. Furthermore, the orthogonal applications of the Σ_i and f_i functions results in cyclic conditions which contradict with a high probability (see Section 4.4). Additionally, more complex conditions on more bits occur. One reason for these additional conditions is that two state variables (A_i, E_i) are updated using a single message word (W_i). Unfortunately, it is not possible to determine all these conditions in general. However, we have implemented different tests to efficiently check for many contradictions (for more details, see Section 4.5).

Despite all these tests, we were not able to find a valid differential characteristic. At the end, even brute-forcing a single critical message word (a message word where most bits are already set) did not lead to a solution. Therefore, we have combined the search for differential characteristics with the search for a conforming message pair (see Section 5). During the message search, we first determine critical bits and backtrack if needed. This way complex hidden conditions are resolved at an earlier stage in the search. Furthermore, we correct impossible characteristics once they are detected.

4 Difference and Condition Propagation in SHA-2

We use generalized conditions to nonlinearly propagate differences and conditions in both the state update and message expansion of SHA-2. Generalized conditions are propagated in a bit sliced manner. Note that in the case of the SHA-2, one bit of A and E is updated using 15 input bits. Hence, to simplify the bit sliced step update, we use an alternative description of SHA-2.

4.1 Alternative Description of SHA-2

In the state update transformation of SHA-2, only two state variables are updated in each step, namely A_i and E_i. Therefore, we can redefine the state

update such that only these two variables are involved. In this case, we get the following mapping between the original and new state variables:

A_i	B_i	C_i	D_i	E_i	F_i	G_i	H_i
A_i	A_{i-1}	A_{i-2}	A_{i-3}	E_i	E_{i-1}	E_{i-2}	E_{i-3}

Note that A_i is updated using an intermediate result of the step update of E_i (see Equation 1). Since this complicates the efficient bit sliced representation of the SHA-2 step update transformation we propose the following alternative description:

$$\begin{aligned} E_i &= E_{i-4} + \Sigma_1(E_{i-1}) + f_1(E_{i-1}, E_{i-2}, E_{i-3}) + A_{i-4} + K_i + W_i \\ A_i &= -A_{i-4} + \Sigma_0(A_{i-1}) + f_0(A_{i-1}, A_{i-2}, A_{i-3}) + E_i \end{aligned} \tag{2}$$

In this case we get two SHA-1 like state update transformations, one for the left (A_i) and one for the right (E_i) side of the SHA-2 state update transformation. Note that in this description, the state variables $A_{-4}, \ldots, A_{-1}$ and $E_{-4}, \ldots, E_{-1}$ represent the chaining input or initial value of the compression function. The alternative description is also illustrated in Fig.2.

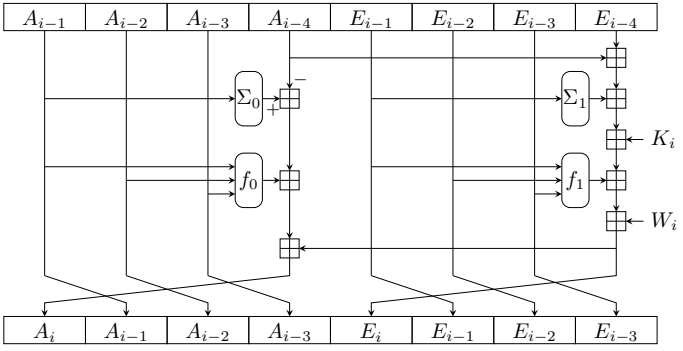

Fig. 2. Alternative description of the SHA-2 state update transformation.

4.2 Generalized Conditions

Inspired by signed-bit differences [19], De Cannière and Rechberger introduced generalized conditions for differences, where all 16 possible conditions on a pair of bits are taken into account [2]. Table 1 lists all these possible conditions and introduces notations for the various cases.

Definition 1 (Generalized Conditions for Differences [2]). *Let* $X \in \{0,1\}^n$ *and* $X^* \in \{0,1\}^n$*, then the notation*

$$\nabla X = [c_{n-1}, \ldots, c_0],$$

Table 1. Notation for possible generalized conditions on a pair of bits [2].

(X_i, X_i^*)	(0,0)	(1,0)	(0,1)	(1,1)	(X_i, X_i^*)	(0,0)	(1,0)	(0,1)	(1,1)
?	✓	✓	✓	✓	3	✓	✓	-	-
-	✓	-	-	✓	5	✓	-	✓	-
x	-	✓	✓	-	7	✓	✓	✓	-
0	✓	-	-	-	A	-	✓	-	✓
u	-	✓	-	-	B	✓	✓	-	✓
n	-	-	✓	-	C	-	-	✓	✓
1	-	-	-	✓	D	✓	-	✓	✓
#	-	-	-	-	E	-	✓	✓	✓

where c_i denotes one of the conditions of Table 1 for the i-th bit, defines a subset of pairs $(X, X^*) \in \{0,1\}^n \times \{0,1\}^n$ that conforms to the specified conditions.

For example, all pairs of 8-bit words X and X^* that satisfy

$$\{(X, X^*) \in \{0,1\}^8 \times \{0,1\}^8 \mid X_7 \cdot X_7^* = 0, X_i = X_i^* \text{ for } 1 \leq i \leq 5, X_0 \neq X_0^*\},$$

can be conveniently written in the form

$$\nabla X = \texttt{[7?-----x]}.$$

4.3 Efficiently Implementing the Propagation of Generalized Conditions

We propagate generalized conditions similar as in the attack on SHA-1. However, the complexity of propagating generalized conditions increases exponentially with the number of input bits and additions. While there are only 6 input bits in the case of SHA-1 (excluding the carry), we have 9 input bits in the update of E_i and 8 input bits in the update of each of A_i and W_i in SHA-2.

To reduce the computational complexity of the propagation in SHA-2, we have further split the update of W_i, E_i and A_i into 3 sub steps. In more detail, we independently compute each output bit of the σ_i, Σ_i and f_i functions and then, compute the modular additions. This way, the number of input bits reduces to 3 for σ_i, Σ_i and f_i and we get at most 5 input bits for the modular additions. This split of functions reduces the computation complexity by a factor of about 100.

Furthermore, for the sub steps without modular addition we have precomputed the propagation of all generalized input conditions. For the modular additions we use a hash map to store already computed bit sliced results. In this case, the bit slice update of each sub step reduces to simple table or hash map lookups. Our experiments have shown a speedup of another factor 100 by caching already computed results. The drawback of this method is that we lose the relation between the sub steps compared to a combined propagation. Furthermore, due to memory restrictions we are not able to precompute or keep all possibilities for the modular additions.

4.4 Two-Bit Conditions

Apart from generalized conditions, additional conditions on more than a single bit are present in a differential characteristic. Especially, conditions on two bits are needed such that a differential path is valid. These two-bit conditions have already been used by Wang et al. in their attacks on the members of the MD4 family [17]. Such two-bit conditions occur mostly in the propagation of differences through the Boolean function. For example, if an input difference in A_{i-1} at bit position j should result in a zero output difference of $f_0(A_{i-1}, A_{i-2}, A_{i-3})$, the remaining two input bits should be equal. In this case, we get the two-bit condition $A_{i-2,j} = A_{i-3,j}$. Similar conditions occur not only in the f_i, σ_i and Σ_i functions but also in the modular additions.

Two-bit conditions are not covered by generalized conditions and thus, not shown in the characteristics given in [2]. However, two-bit conditions may lead to additional inconsistencies. For example, in two subsequent f_0 functions the following two contradicting conditions may occur:

$$(A_{i-2,j} = A_{i-3,j}) \wedge (A_{i-2,j} \neq A_{i-3,j}).$$

Since such contradicting conditions occur only rarely in SHA-1, simple additional checks are sufficient to verify whether a given differential characteristic is valid at the end of the search.

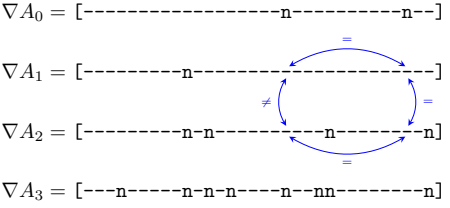

Fig. 3. Example of four cyclic and contradicting two-bit conditions. Such cases commonly occur in SHA-2 and are not covered by generalized conditions. For the two Σ_0 functions (XOR) we have twice $\Sigma_0(\mathtt{n}, \mathtt{-}, \mathtt{-}) = \mathtt{n}$ which results in the two equalities $A_{1,2} = A_{1,13}$ and $A_{2,2} = A_{2,13}$. For the f_0 function (MAJ) at bit position 2 we get $f_0(\mathtt{-}, \mathtt{-}, \mathtt{n}) = \mathtt{n}$ if and only if $A_{2,2} = A_{1,2}$, while for bit position 13 we get $f_0(\mathtt{-}, \mathtt{-}, \mathtt{n}) = \mathtt{-}$ if and only if $A_{2,13} \neq A_{1,13}$. Note that in this example, all involved bits of E_i do not contain any difference.

This is not the case in SHA-2. Note that the nonlinear Boolean functions f_0 and f_1 update the same bit position of different words, while the linear Σ_i functions update different bit positions within the same word. Hence, more complex cyclic two-bit relations occur. A still simple example is given in Fig.3. In this case, 4 bits of two Σ_i and two Boolean functions are related in a cyclic form which results in a contradiction. We have observed that for a given differential characteristic even more complex relations with cycle lengths larger than

10 commonly occur. Of course already a single contradicting cycle results in an impossible differential characteristic.

4.5 Inconsistency Checks

To avoid inconsistent differential characteristics, we have evaluated a number of checks to detect contradictions as early and efficiently as possible. Note that a test which is able to detect many contradictions is usually also less efficient. However, also a simple test may detect a contradiction at a later point in the search. Due to the high number of complex conditions in SHA-2 and the difficulty to detect them we need to make a trade-off here.

Two-Bit Condition Check. Two-bit conditions are linear conditions in $GF(2)$ since such conditions can only be either equal ($A_{i,j} = A_{i-1,j}$) or non-equal ($A_{i,j} \neq A_{i-1,j}$). Contradictions in two-bit condition cycles can be efficiently detected by determining all two-bit conditions, setting up a linear system of equations and checking if the system can be solved using Gaussian elimination. Although a large number of contradictions are detected this way, most characteristics are still invalid after this check.

Complete Condition Check. A quite expensive test is to check for every bit restricted to `'-'` or `'x'` whether both possible cases (`'0'` and `'1'`, or `'n'` and `'u'`) are indeed still valid. If both choices for a single bit are invalid we know that the whole characteristic is impossible. Of course these tests can be extended to other generalized conditions as well. However, it turned out to be more efficient to apply this check only rarely and only to specific conditions during the search. Furthermore, we have improved the speed of this complete test by applying it only to bits which are restricted by two-bit conditions.

Complete Condition Check on a Set of Bits. Since even the complete condition check is not able to detect many contradictions, we have analyzed different variants of setting all possibilities for all or selected combinations of 2, 3 or 4 bits. Such tests indeed detect more impossible characteristics but are very inefficient to compute and thus, cannot be used during the search for differential characteristics in SHA-2.

5 Searching for Differential Characteristics

In general, our search techniques can be divided into three parts: decision, deduction and backtracking. Note that the same separation is done in many other fields, like SAT solvers [5]. The first aspect of our search strategy is the decision, where we decide which bit is chosen and which condition is imposed at its position. In the deduction part we compute the propagation of the imposed condition and check for contradictions. If a contradiction occurs we need to backtrack and

undo decisions, which is the third part of the search strategy. A basic search strategy to find differential characteristics has been described in [2] and works as follows.

Let U be the set of all '?' and 'x', then repeat the following until U is empty.
Decision
1. Pick randomly a bit in U.
2. Impose a '-' for a '?' or randomly a sign ('u' or 'n') for 'x'.

Deduction
3. Compute the propagation.
4. If a contradiction is detected start backtracking, else go to step 1.

Backtracking
5. Jump back to an earlier state of the search and go to step 1.

We have applied this strategy to SHA-2 but could not find a valid differential characteristics. In any case at least one of the checks described in Section 4.5 failed. The reason for this is that conditions which are not covered by generalized or two-bit conditions appear much more often in SHA-2 than in SHA-1. Since more advanced checks are too expensive, we have developed a more sophisticated search strategy to find valid differential characteristics for SHA-2 as described in the next section.

5.1 Search Strategy

In our approach we already determine some message bits during the search for a differential characteristic. Generally speaking, we are combining the search for a conforming message pair with the search for a differential characteristic. In doing so we consider those bits much earlier, which are involved in many relations with other bits. This way, we can detect invalid characteristics at an early stage of the search. However, this should not be done too early to not restrict the message freedom too much. In addition, we are remembering critical bits during the search to improve the backtracking and speed-up the search process. In the following we describe the used search strategy in more detail.

In general we have two phases in our search strategy where different bits are chosen (guessed) and we switch between these two dynamically. In the following, we describe both phases in detail. Phase 1 can be described as follows.

Let U be the set of all '?' and 'x'. Repeat the following until U is empty:
Decision
1. Pick randomly a bit in U.
2. Impose a '-' for a '?' or randomly a sign ('u' or 'n') for 'x'.

Deduction
3. Compute the propagation as described in Section 4.3.
4. If a contradiction is detected start backtracking, else apply the additional checks of Section 4.5.
5. Continue with step 1 if all checks passed, if not start backtracking.

Backtracking

6. If the decision bit is `'x'` try the second choice for the sign or if the decision bit is `'?'` impose a `'x'`.
7. If still a contradiction occurs mark bit as critical.
8. Jump back until the critical bit ca be resolved.
9. Continue with step 1.

Note that, the additional checks in step 4 are optional and a trade-off between number of checks and speed has to be done. The additional steps in the backtracking process improve the search speed significantly and prevent that critical bits result in a contradiction again.

Once phase 1 is finished (U is empty) we continue with phase 2 which can be summarized as follows.

Let U' be the set of all `'-'` with many two-bit conditions.
Repeat the following until U' is empty:

Decision

1. Pick randomly a bit in U'.
2. Impose randomly a `'0'` or `'1'`.

Deduction

3. Compute the propagation as described in Section 4.3.
4. If a contradiction is detected start backtracking, else apply additional checks from Section 4.5.
5. Continue with step 1 if all checks passed, if not start backtracking.

Backtracking

6. Try the second choice of the decision bit.
7. If still a contradiction occurs mark bit as critical.
8. Jump back until the critical bit can be resolved.
9. If necessary jump back to phase 1, otherwise continue with step 1.

Choosing a decision bit with many two-bit conditions ensures that bits which influence a lot of other bits are chosen first. Therefore, many other bits propagate by defining the value of a single bit. Furthermore, in step 7 and 8 of the backtracking we can also mark more than one bit as critical. We want to note that due to step 9, we actually switch quite often between both phases in our search.

Additionally, we restart the search from scratch after a certain amount of contradictions or iterations to terminate branches which appear to be stuck because of exploring a search space far from a solution.

5.2 Results

Using the start characteristic given in Table 2 and the search strategy described above, we can find a valid characteristic and confirming inputs which result in semi-free-collisions for 32 out of 64 steps of SHA-256. An example of a semi-free-start for 32 steps is shown in Table 4. The according differential characteristic and the set of conditions is given in Table 3 and Table 5. The find this example for 32 steps our tool was running a few days on a cluster with 32 nodes.

So far we have only considered semi-free-start collision attacks in which an attacker is allowed to choose the chaining value. However, in a collision attack on the hash function the chaining value is fixed, which makes an attack much more difficult. In order to construct a collision for step-reduced SHA-256, we are interested in differential characteristics with no differences in the first few message words. Then, the additional freedom in the first message words can be used to transform a semi-free-start collision into a real collision. Similar characteristics have also been used in the collision attacks on 24 steps of SHA-256 in [7].

By using a differential characteristic spanning over $t = 11$ steps with differences in only 5 expanded message words and with no differences in the first 7 message words (see Table 6) we are able to construct a collision for 27 steps of SHA-256. The colliding message pair is shown in Table 8 and the differential characteristic and the set of conditions is given in Table 7 and Table 9.

6 Conclusions and Future Work

In this paper, we have presented a collision for 27 and a semi-free-start collision for 32 steps of SHA-256 with practical complexity. This significantly improves upon the best previously published (semi-free-start) collision attacks on SHA-256 for up to 24 steps. We have extended and generalized existing approaches and developed a fully automatic tool to construct complex differential characteristics for SHA-2.

Our tool extends the techniques proposed by De Cannière and Rechberger to construct complex characteristics for SHA-1 using generalized conditions. The more complex structure of SHA-256 complicates a direct application of their approach. We have identified several problems and have shown how to overcome them. Most importantly, a high amount of found differential characteristics are invalid due to many contradicting conditions in SHA-2. We have resolved this problem by by identifying critical bits during the whole search process, and by combining the search for differential characteristics with the computation of conforming message pairs.

To summarize, the search for valid differential characteristics and conforming message pairs in SHA-2 is increasingly difficult and unpredictable, compared to more simple ARX-based designs like MD5 and SHA-1. Nevertheless, we were able to construct a powerful tool to find practical examples for (semi-free-start) collisions in SHA-256 which can also be applied to other ARX based hash functions.

Acknowledgments

We would like to thank Vincent Rijmen, Christian Rechberger, Christophe De Cannière and the anonymous referees for useful comments and discussions. The work in this paper has been supported in part by the Secure Information Technology Center-Austria (A-SIT), by the European Commission under contract

ICT-2007-216646 (ECRYPT II), by the Austrian Science Fund (FWF, project P21936) and the German Federal Office for Information Security (BSI).

References

1. Chabaud, F., Joux, A.: Differential Collisions in SHA-0. In: Krawczyk, H. (ed.) CRYPTO. LNCS, vol. 1462, pp. 56–71. Springer (1998)
2. De Cannière, C., Rechberger, C.: Finding SHA-1 Characteristics: General Results and Applications. In: Lai, X., Chen, K. (eds.) ASIACRYPT. LNCS, vol. 4284, pp. 1–20. Springer (2006)
3. Gilbert, H., Handschuh, H.: Security Analysis of SHA-256 and Sisters. In: Matsui, M., Zuccherato, R.J. (eds.) Selected Areas in Cryptography. LNCS, vol. 3006, pp. 175–193. Springer (2003)
4. Grechnikov, E.: Collisions for 72-step and 73-step SHA-1: Improvements in the Method of Characteristics. Cryptology ePrint Archive, Report 2010/413 (2010)
5. Gu, J., Purdom, P.W., Franco, J., Wah, B.W.: Algorithms for the Satisfiability (SAT) Problem: A Survey. In: DIMACS Series in Discrete Mathematics and Theoretical Computer Science. pp. 19–152. American Mathematical Society (1996)
6. Hawkes, P., Paddon, M., Rose, G.G.: On Corrective Patterns for the SHA-2 Family. Cryptology ePrint Archive, Report 2004/207 (2004)
7. Indesteege, S., Mendel, F., Preneel, B., Rechberger, C.: Collisions and Other Nonrandom Properties for Step-Reduced SHA-256. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) Selected Areas in Cryptography. LNCS, vol. 5381, pp. 276–293. Springer (2008)
8. Mendel, F., Pramstaller, N., Rechberger, C., Rijmen, V.: Analysis of Step-Reduced SHA-256. In: Robshaw, M.J.B. (ed.) FSE. LNCS, vol. 4047, pp. 126–143. Springer (2006)
9. National Institute of Standards and Technology: Cryptographic Hash Algorithm Competition (November 2007), available online: `http://csrc.nist.gov/groups/ST/hash/sha-3/index.html`
10. National Institute of Standards and Technology: FIPS PUB 180-3: Secure Hash Standard. Federal Information Processing Standards Publication 180-3, U.S. Department of Commerce (October 2008), available online: `http://www.itl.nist.gov/fipspubs`
11. Nikolić, I., Biryukov, A.: Collisions for Step-Reduced SHA-256. In: Nyberg, K. (ed.) FSE. LNCS, vol. 5086, pp. 1–15. Springer (2008)
12. Sanadhya, S.K., Sarkar, P.: New Local Collisions for the SHA-2 Hash Family. In: Nam, K.H., Rhee, G. (eds.) ICISC. LNCS, vol. 4817, pp. 193–205. Springer (2007)
13. Sanadhya, S.K., Sarkar, P.: Attacking Reduced Round SHA-256. In: Bellovin, S.M., Gennaro, R., Keromytis, A.D., Yung, M. (eds.) ACNS. LNCS, vol. 5037, pp. 130–143 (2008)
14. Sanadhya, S.K., Sarkar, P.: Deterministic Constructions of 21-Step Collisions for the SHA-2 Hash Family. In: Wu, T.C., Lei, C.L., Rijmen, V., Lee, D.T. (eds.) ISC. LNCS, vol. 5222, pp. 244–259. Springer (2008)
15. Sanadhya, S.K., Sarkar, P.: New Collision Attacks against Up to 24-Step SHA-2. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT. LNCS, vol. 5365, pp. 91–103. Springer (2008)
16. Sanadhya, S.K., Sarkar, P.: Non-linear Reduced Round Attacks against SHA-2 Hash Family. In: Mu, Y., Susilo, W., Seberry, J. (eds.) ACISP. LNCS, vol. 5107, pp. 254–266. Springer (2008)

17. Wang, X., Lai, X., Feng, D., Chen, H., Yu, X.: Cryptanalysis of the Hash Functions MD4 and RIPEMD. In: Cramer, R. (ed.) EUROCRYPT. LNCS, vol. 3494, pp. 1–18. Springer (2005)
18. Wang, X., Yao, A., Yao, F.: New Collision Search for SHA-1. Presented at rump session of CRYPTO (2005)
19. Wang, X., Yin, Y.L., Yu, H.: Finding Collisions in the Full SHA-1. In: Shoup, V. (ed.) CRYPTO. LNCS, vol. 3621, pp. 17–36. Springer (2005)
20. Wang, X., Yu, H.: How to Break MD5 and Other Hash Functions. In: Cramer, R. (ed.) EUROCRYPT. LNCS, vol. 3494, pp. 19–35. Springer (2005)

A Differential Characteristics and Conditions

Table 2. Starting point for a semi-free-start collision for 32 steps. Using the alternative description of SHA-2 (Section 4.1) and the notion of generalized conditions (Section 4.2).

i	∇A_i	∇E_i	∇W_i
-4	----------	----------	
-3	----------	----------	
-2	----------	----------	
-1	----------	----------	
0	----------	----------	----------
1	----------	----------	----------
2	----------x--	----------x--	----------x--
3	??????????????????????????????	??????????????????????????????	??????????????????????????????
4	??????????????????????????????	??????????????????????????????	??????????????????????????????
5	??????????????????????????????	??????????????????????????????	??????????????????????????????
6	??????????????????????????????	??????????????????????????????	??????????????????????????????
7	????????????????----------	??????????????????????????????	??????????????????????????????
8	????????????????----------	??????????????????????????????	????????????????x----------
9	????????????????x----------	??????????????????????????????	----------
10	----------	??????????????????????????????	----------
11	----------	????????????????----------	----------
12	----------	????????????????----------	----------
13	----------	????????????????x----------	----------
14	----------	----------	----------
15	----------	----------	----------
16	----------	----------	----------
17	----------	----------	---x-------x----------
18	----------	----------	----------
19	----------	----------	----------
	...	...	...
30	----------	----------	----------
31	----------	----------	----------

Table 3. Characteristic for a semi-free-start collision for 32 steps of SHA-256.

i	∇A_i	∇E_i	∇W_i
-4	----------	----------	
-3	----------	----------	
-2	----------	----------	
-1	----------	----------	
0	----------	----------	----------
1	----------1-----	--0-0---1--1----1-0-0-------011-	----------
2	----------0--u-	--1-1-1000-1--11101101---1--1u0-	----------u--
3	10n10nnn1n0n-11n1u01u11000uu0n0n	-1n1n10un0un101-n1n1n0110un0u0n0	uu-un-----un---n-u-uu-n---u--un-
4	-----n----------0---------0-1---	-0n0n1nuuun0-1u1unnnuu011n000nn1	1n---1u--uu1u-uu------nn--0n----
5	----------n---------1-----	0u1nn1n-1010-00001u0101-11101110	01-1-un0-1-1n-nn1u1n0-0un-0-n--n
6	----------n--u--------u---n--	00u01un0000000n111u00100101uu11u	n----nnuu-n-nu---n--n----------
7	----------	-n10u000u1un0101nn10n00001n000u1	1n0001un10u0nnn-01n01u10000unnnn
8	----------	-10-1n0-0--1-01-0-1-0----n011-10	----u------unnn----------0-----
9	----u---------u----------	-0--u00-1-01-1--1---1----n1---0-	----------
10	----------	---nunn------n--n--------u-u-u--	----------
11	----------	---0-10----100--0--------1-0-0--	----------
12	----------	---0011----011--1--------0-1-1--	----------
13	----------	---un------unnnn----------------	----------
14	----------	---00------00000----------------	----------
15	----------	---11------11111----------------	----------
16	----------	----------	----------
17	----------	----------	----n---------n----------
18	----------	----------	----------
19	----------	----------	----------
20	----------	----------	----------
21	----------	----------	----------
22	----------	----------	----------
23	----------	----------	----------
24	----------	----------	----------
25	----------	----------	----------
26	----------	----------	----------
27	----------	----------	----------
28	----------	----------	----------
29	----------	----------	----------
30	----------	----------	----------
31	----------	----------	----------

Table 4. Semi-free-start collision for 32 steps of SHA-256.

h_0	764d264f 268a3366 285fecb1 4c389b22 75cd568d f5c8f99b 6e7a3cc3 1b4ea134
h_0^*	764d264f 268a3366 285fecb1 4c389b22 75cd568d f5c8f99b 6e7a3cc3 1b4ea134
Δh_0	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
m	52a600a8 2c3b8434 ea92dfcf d4eaf9ad b77fe08d 7c50e542 69c783a6 86a14e10 baf88b0b 12665efb ce7c3a31 3030f09d 9bd52eb8 7549997e fa976e0d 86ebacbc
m^*	52a600a8 2c3b8434 ea92dfcb 0cdba38b f514e39d 7a5bb4cb ee6bcba6 c58f6a0f b2f78b0b 12665efb ce7c3a31 3030f09d 9bd52eb8 7549997e fa976e0d 86ebacbc
Δm	00000000 00000000 00000004 d8315a26 426b0310 060b5189 87ac4800 432e241f 080f0000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
h_1	d0b41ffa e1f519a2 e3cad2ed a19d5795 906ac05f c995f6c8 cf309f95 9fb9ca57
h_1^*	d0b41ffa e1f519a2 e3cad2ed a19d5795 906ac05f c995f6c8 cf309f95 9fb9ca57
Δh_1	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Table 5. Set of conditions for the semi-free-start collision for 32 steps.

i	set of conditions	
0	$E_{0,2} = 0$	1
1	$A_{1,5} = 1$, $A_{1,2} \neq A_{0,2}$, $A_{1,0} = E_{1,0}$, $E_{1,1} = 1$, $E_{1,2} = 1$, $E_{1,3} = 0$, $E_{1,11} = 0$, $E_{1,13} = 0$, $E_{1,15} = 1$, $E_{1,20} = 1$, $E_{1,23} = 1$, $E_{1,27} = 0$, $E_{1,29} = 0$	13
2	$A_{2,2} = 1$, $A_{2,5} = 0$, $A_{2,0} = A_{1,0}$, $A_{2,4} \neq A_{1,4}$, $A_{2,11} = A_{1,11}$, $A_{2,14} \neq A_{1,14}$, $A_{2,16} = A_{1,16}$, $A_{2,20} = A_{1,20}$, $A_{2,22} = A_{1,22}$, $A_{2,24} \neq A_{1,24}$, $A_{2,25} \neq A_{1,25}$, $A_{2,26} \neq A_{1,26}$, $A_{2,29} \neq A_{1,29}$, $A_{2,23} = A_{2,11}$, $A_{2,22} = A_{2,13}$, $A_{2,25} \neq A_{2,14}$, $E_{2,1} = 0$, $E_{2,2} = 1$, $E_{2,3} = 1$, $E_{2,6} = 1$, $E_{2,10} = 1$, $E_{2,11} = 0$, $E_{2,12} = 1$, $E_{2,13} = 1$, $E_{2,14} = 0$, $E_{2,15} = 1$, $E_{2,16} = 1$, $E_{2,17} = 1$, $E_{2,20} = 1$, $E_{2,22} = 0$, $E_{2,23} = 0$, $E_{2,24} = 0$, $E_{2,25} = 1$, $E_{2,27} = 1$, $E_{2,29} = 1$, $E_{2,5} \neq E_{1,5}$, $E_{2,21} = E_{2,7}$	40
	$W_{2,2} = 1$, $W_{2,30} \neq W_{2,13}$, $W_{2,23} \neq W_{2,19}$	
3	$A_{3,0} = 0$, $A_{3,1} = 0$, $A_{3,2} = 0$, $A_{3,3} = 0$, $A_{3,4} = 1$, $A_{3,5} = 1$, $A_{3,6} = 0$, $A_{3,7} = 0$, $A_{3,8} = 0$, $A_{3,9} = 1$, $A_{3,10} = 1$, $A_{3,11} = 1$, $A_{3,12} = 1$, $A_{3,13} = 0$, $A_{3,14} = 1$, $A_{3,15} = 1$, $A_{3,16} = 0$, $A_{3,17} = 1$, $A_{3,18} = 1$, $A_{3,20} = 0$, $A_{3,21} = 0$, $A_{3,22} = 0$, $A_{3,23} = 1$, $A_{3,24} = 0$, $A_{3,25} = 0$, $A_{3,26} = 0$, $A_{3,27} = 0$, $A_{3,28} = 1$, $A_{3,29} = 0$, $A_{3,30} = 0$, $A_{3,31} = 1$, $E_{3,0} = 0$, $E_{3,1} = 0$, $E_{3,2} = 0$, $E_{3,3} = 1$, $E_{3,4} = 0$, $E_{3,5} = 0$, $E_{3,6} = 1$, $E_{3,7} = 0$, $E_{3,8} = 1$, $E_{3,9} = 1$, $E_{3,10} = 0$, $E_{3,11} = 0$, $E_{3,12} = 1$, $E_{3,13} = 0$, $E_{3,14} = 1$, $E_{3,15} = 0$, $E_{3,17} = 1$, $E_{3,18} = 0$, $E_{3,19} = 1$, $E_{3,20} = 0$, $E_{3,21} = 1$, $E_{3,22} = 0$, $E_{3,23} = 0$, $E_{3,24} = 1$, $E_{3,25} = 0$, $E_{3,26} = 1$, $E_{3,27} = 0$, $E_{3,28} = 1$, $E_{3,29} = 0$, $E_{3,30} = 1$	92
	$W_{3,1} = 0$, $W_{3,2} = 1$, $W_{3,5} = 1$, $W_{3,9} = 0$, $W_{3,11} = 1$, $W_{3,12} = 1$, $W_{3,14} = 1$, $W_{3,16} = 0$, $W_{3,20} = 0$, $W_{3,21} = 1$, $W_{3,27} = 0$, $W_{3,28} = 1$, $W_{3,30} = 1$, $W_{3,31} = 1$, $W_{3,17} = W_{3,0}$, $W_{3,24} \neq W_{3,3}$, $W_{3,25} = W_{3,4}$, $W_{3,10} = W_{3,6}$, $W_{3,23} \neq W_{3,6}$, $W_{3,22} = W_{3,7}$, $W_{3,24} \neq W_{3,7}$, $W_{3,23} = W_{3,8}$, $W_{3,25} = W_{3,10}$, $W_{3,17} = W_{3,13}$, $W_{3,24} \neq W_{3,13}$, $W_{3,19} = W_{3,15}$, $W_{3,26} = W_{3,15}$, $W_{3,22} \neq W_{3,18}$, $W_{3,29} = W_{3,18}$, $W_{3,23} = W_{3,19}$, $W_{3,26} = W_{3,22}$	
4	$A_{4,3} = 1$, $A_{4,5} = 0$, $A_{4,15} = 0$, $A_{4,26} = 0$, $A_{4,0} = A_{2,0}$, $A_{4,4} \neq A_{2,4}$, $A_{4,11} = A_{2,11}$, $A_{4,14} \neq A_{2,14}$, $A_{4,16} \neq A_{2,16}$, $A_{4,20} = A_{2,20}$, $A_{4,22} = A_{2,22}$, $A_{4,24} \neq A_{2,24}$, $A_{4,29} \neq A_{2,29}$, $A_{4,17} = A_{4,6}$, $E_{4,0} = 1$, $E_{4,1} = 0$, $E_{4,2} = 0$, $E_{4,3} = 0$, $E_{4,4} = 0$, $E_{4,5} = 0$, $E_{4,6} = 0$, $E_{4,7} = 1$, $E_{4,8} = 1$, $E_{4,9} = 0$, $E_{4,10} = 1$, $E_{4,11} = 1$, $E_{4,12} = 0$, $E_{4,13} = 0$, $E_{4,14} = 0$, $E_{4,15} = 1$, $E_{4,16} = 1$, $E_{4,17} = 1$, $E_{4,18} = 1$, $E_{4,20} = 0$, $E_{4,21} = 0$, $E_{4,22} = 1$, $E_{4,23} = 1$, $E_{4,24} = 1$, $E_{4,25} = 0$, $E_{4,26} = 1$, $E_{4,27} = 0$, $E_{4,28} = 0$, $E_{4,29} = 0$, $E_{4,30} = 0$, $E_{4,19} \neq A_{3,19}$	67
	$W_{4,4} = 0$, $W_{4,5} = 0$, $W_{4,8} = 0$, $W_{4,9} = 0$, $W_{4,16} = 1$, $W_{4,17} = 1$, $W_{4,19} = 1$, $W_{4,20} = 1$, $W_{4,21} = 1$, $W_{4,22} = 1$, $W_{4,25} = 1$, $W_{4,26} = 1$, $W_{4,30} = 0$, $W_{4,31} = 1$, $W_{4,18} \neq W_{4,1}$, $W_{4,13} = W_{4,2}$, $W_{4,23} \neq W_{4,2}$, $W_{4,10} = W_{4,6}$, $W_{4,11} \neq W_{4,7}$, $W_{4,23} = W_{4,12}$, $W_{4,27} = W_{4,12}$, $W_{4,28} = W_{4,13}$	
5	$A_{5,5} = 1$, $A_{5,15} = 0$, $A_{5,0} \neq A_{4,0}$, $A_{5,2} \neq A_{4,2}$, $A_{5,4} \neq A_{4,4}$, $A_{5,6} \neq A_{4,6}$, $A_{5,11} = A_{4,11}$, $A_{5,14} = A_{4,14}$, $A_{5,16} \neq A_{4,16}$, $A_{5,18} = A_{4,18}$, $A_{5,20} = A_{4,20}$, $A_{5,22} = A_{4,22}$, $A_{5,24} = A_{4,24}$, $A_{5,25} = A_{4,25}$, $A_{5,29} \neq A_{4,29}$, $A_{5,26} = A_{5,3}$, $A_{5,24} = A_{5,4}$, $A_{5,27} \neq A_{5,6}$, $E_{5,0} = 0$, $E_{5,1} = 1$, $E_{5,2} = 1$, $E_{5,3} = 1$, $E_{5,4} = 0$, $E_{5,5} = 1$, $E_{5,6} = 1$, $E_{5,7} = 1$, $E_{5,9} = 1$, $E_{5,10} = 0$, $E_{5,11} = 1$, $E_{5,12} = 0$, $E_{5,13} = 1$, $E_{5,14} = 1$, $E_{5,15} = 0$, $E_{5,16} = 0$, $E_{5,17} = 0$, $E_{5,18} = 0$, $E_{5,20} = 0$, $E_{5,21} = 1$, $E_{5,22} = 0$, $E_{5,23} = 1$, $E_{5,25} = 0$, $E_{5,26} = 1$, $E_{5,27} = 0$, $E_{5,28} = 0$, $E_{5,29} = 1$, $E_{5,30} = 1$, $E_{5,31} = 0$, $E_{1,0} = A_{1,0}$	74
	$W_{5,0} = 0$, $W_{5,3} = 0$, $W_{5,5} = 0$, $W_{5,7} = 0$, $W_{5,8} = 1$, $W_{5,9} = 0$, $W_{5,11} = 0$, $W_{5,12} = 0$, $W_{5,13} = 1$, $W_{5,14} = 1$, $W_{5,15} = 1$, $W_{5,16} = 0$, $W_{5,17} = 0$, $W_{5,19} = 0$, $W_{5,20} = 1$, $W_{5,22} = 1$, $W_{5,24} = 0$, $W_{5,25} = 0$, $W_{5,26} = 1$, $W_{5,28} = 1$, $W_{5,30} = 1$, $W_{5,31} = 0$, $W_{5,29} = W_{5,1}$, $W_{5,23} = W_{5,2}$, $W_{5,21} = W_{5,4}$, $W_{5,29} \neq W_{5,18}$	
6	$A_{6,2} = 0$, $A_{6,6} = 1$, $A_{6,15} = 1$, $A_{6,18} = 0$, $A_{6,26} = A_{5,26}$, $A_{6,26} = A_{6,3}$, $A_{6,24} \neq A_{6,4}$, $A_{6,27} \neq A_{6,7}$, $A_{6,30} \neq A_{6,9}$, $A_{6,23} \neq A_{6,11}$, $A_{6,22} \neq A_{6,13}$, $A_{6,25} = A_{6,14}$, $A_{6,26} \neq A_{6,17}$, $E_{6,0} = 1$, $E_{6,1} = 1$, $E_{6,2} = 1$, $E_{6,3} = 1$, $E_{6,4} = 1$, $E_{6,5} = 1$, $E_{6,6} = 0$, $E_{6,7} = 1$, $E_{6,8} = 0$, $E_{6,9} = 0$, $E_{6,10} = 1$, $E_{6,11} = 0$, $E_{6,12} = 0$, $E_{6,13} = 1$, $E_{6,14} = 1$, $E_{6,15} = 1$, $E_{6,16} = 1$, $E_{6,17} = 0$, $E_{6,18} = 0$, $E_{6,19} = 0$, $E_{6,20} = 0$, $E_{6,21} = 0$, $E_{6,22} = 0$, $E_{6,23} = 0$, $E_{6,24} = 0$, $E_{6,25} = 0$, $E_{6,26} = 1$, $E_{6,27} = 1$, $E_{6,28} = 0$, $E_{6,29} = 1$, $E_{6,30} = 0$, $E_{6,31} = 0$,	73
	$W_{6,11} = 0$, $W_{6,14} = 0$, $W_{6,18} = 1$, $W_{6,19} = 0$, $W_{6,21} = 0$, $W_{6,23} = 1$, $W_{6,24} = 1$, $W_{6,25} = 0$, $W_{6,26} = 0$, $W_{6,31} = 0$, $W_{6,17} \neq W_{6,0}$, $W_{6,28} = W_{6,0}$, $W_{6,22} = W_{6,1}$, $W_{6,7} \neq W_{6,3}$, $W_{6,20} = W_{6,3}$, $W_{6,8} \neq W_{6,4}$, $W_{6,22} = W_{6,5}$, $W_{6,10} = W_{6,6}$, $W_{6,27} \neq W_{6,6}$, $W_{6,22} = W_{6,7}$, $W_{6,28} \neq W_{6,7}$, $W_{6,12} \neq W_{6,8}$, $W_{6,29} = W_{6,8}$, $W_{6,13} \neq W_{6,9}$, $W_{6,30} = W_{6,9}$, $W_{6,27} \neq W_{6,10}$, $W_{6,30} = W_{6,15}$, $W_{6,20} \neq W_{6,16}$	
7	$A_{7,2} = A_{5,2}$, $A_{7,6} \neq A_{5,6}$, $A_{7,18} = A_{5,18}$, $E_{7,0} = 1$, $E_{7,1} = 1$, $E_{7,2} = 0$, $E_{7,3} = 0$, $E_{7,4} = 0$, $E_{7,5} = 0$, $E_{7,6} = 1$, $E_{7,7} = 0$, $E_{7,8} = 0$, $E_{7,9} = 0$, $E_{7,10} = 0$, $E_{7,11} = 0$, $E_{7,12} = 0$, $E_{7,13} = 1$, $E_{7,14} = 0$, $E_{7,15} = 0$, $E_{7,16} = 1$, $E_{7,17} = 0$, $E_{7,18} = 1$, $E_{7,19} = 0$, $E_{7,20} = 0$, $E_{7,21} = 1$, $E_{7,22} = 1$, $E_{7,23} = 1$, $E_{7,24} = 0$, $E_{7,25} = 0$, $E_{7,26} = 0$, $E_{7,27} = 1$, $E_{7,28} = 0$, $E_{7,29} = 1$, $E_{7,30} = 0$	66
	$W_{7,0} = 0$, $W_{7,1} = 0$, $W_{7,2} = 0$, $W_{7,3} = 0$, $W_{7,4} = 1$, $W_{7,5} = 0$, $W_{7,6} = 0$, $W_{7,7} = 0$, $W_{7,8} = 0$, $W_{7,9} = 1$, $W_{7,10} = 1$, $W_{7,11} = 1$, $W_{7,12} = 0$, $W_{7,13} = 0$, $W_{7,14} = 1$, $W_{7,15} = 0$, $W_{7,17} = 0$, $W_{7,18} = 0$, $W_{7,19} = 0$, $W_{7,20} = 0$, $W_{7,21} = 1$, $W_{7,22} = 0$, $W_{7,23} = 1$, $W_{7,24} = 0$, $W_{7,25} = 1$, $W_{7,26} = 1$, $W_{7,27} = 0$, $W_{7,28} = 0$, $W_{7,29} = 0$, $W_{7,30} = 0$, $W_{7,31} = 1$, $W_{7,16} \neq E_{3,16}$	
8	$A_{8,2} = A_{7,2}$, $A_{8,6} \neq A_{7,6}$, $A_{8,15} \neq A_{7,15}$, $A_{8,16} \neq A_{7,16}$, $A_{8,18} = A_{7,18}$, $A_{8,27} \neq A_{7,27}$, $E_{8,0} = 0$, $E_{8,1} = 1$, $E_{8,3} = 1$, $E_{8,4} = 1$, $E_{8,5} = 0$, $E_{8,6} = 0$, $E_{8,11} = 0$, $E_{8,13} = 1$, $E_{8,15} = 0$, $E_{8,17} = 1$, $E_{8,18} = 0$, $E_{8,20} = 1$, $E_{8,23} = 0$, $E_{8,25} = 0$, $E_{8,26} = 0$, $E_{8,27} = 1$, $E_{8,29} = 0$, $E_{8,30} = 1$, $E_{8,12} = E_{8,7}$, $E_{8,21} \neq E_{8,8}$,	46
	$W_{8,5} = 0$, $W_{8,16} = 0$, $W_{8,17} = 0$, $W_{8,18} = 0$, $W_{8,19} = 1$, $W_{8,27} = 1$, $W_{8,0} \neq A_{4,0}$, $W_{8,1} = A_{4,1}$, $W_{8,4} \neq A_{4,4}$, $W_{8,21} = W_{8,0}$, $W_{8,22} = W_{8,1}$, $W_{8,23} \neq W_{8,2}$, $W_{8,7} \neq W_{8,3}$, $W_{8,8} \neq W_{8,4}$, $W_{8,23} \neq W_{8,6}$, $W_{8,31} \neq W_{8,10}$, $W_{8,28} \neq W_{8,13}$, $W_{8,29} \neq W_{8,14}$, $W_{8,30} \neq W_{8,15}$, $W_{8,31} = W_{8,20}$	
9	$A_{9,16} = 1$, $A_{9,27} = 1$, $A_{9,25} = A_{9,5}$, $A_{9,15} = A_{9,6}$, $A_{9,18} \neq A_{9,7}$, $A_{9,28} = A_{9,7}$, $E_{9,1} = 0$, $E_{9,5} = 1$, $E_{9,6} = 0$, $E_{9,11} = 1$, $E_{9,15} = 1$, $E_{9,18} = 1$, $E_{9,20} = 1$, $E_{9,21} = 0$, $E_{9,23} = 1$, $E_{9,25} = 0$, $E_{9,26} = 0$, $E_{9,27} = 1$, $E_{9,30} = 0$, $E_{9,14} \neq E_{9,0}$, $E_{9,13} \neq E_{9,8}$, $E_{9,22} = E_{9,9}$	22
10	$A_{10,16} = A_{8,16}$, $A_{10,27} = A_{8,27}$, $E_{10,2} = 1$, $E_{10,4} = 1$, $E_{10,6} = 1$, $E_{10,15} = 0$, $E_{10,18} = 0$, $E_{10,25} = 0$, $E_{10,26} = 0$, $E_{10,27} = 1$, $E_{10,28} = 0$, $E_{10,13} \neq E_{10,0}$, $E_{10,14} \neq E_{10,0}$, $E_{10,23} = E_{10,5}$, $E_{10,20} = E_{10,7}$, $E_{10,21} \neq E_{10,8}$, $E_{10,22} \neq E_{10,9}$, $E_{10,23} = E_{10,9}$, $E_{10,23} = E_{10,10}$, $E_{10,29} = E_{10,10}$, $E_{10,30} \neq E_{10,12}$, $E_{10,31} \neq E_{10,13}$, $E_{10,29} \neq E_{10,16}$, $E_{10,22} \neq E_{10,17}$, $E_{10,24} = E_{10,19}$	25
11	$A_{11,16} = A_{10,16}$, $A_{11,27} = A_{10,27}$, $E_{11,2} = 0$, $E_{11,4} = 0$, $E_{11,6} = 1$, $E_{11,15} = 0$, $E_{11,18} = 0$, $E_{11,19} = 0$, $E_{11,20} = 1$, $E_{11,25} = 0$, $E_{11,26} = 1$, $E_{11,28} = 0$	12
12	$E_{12,2} = 1$, $E_{12,4} = 1$, $E_{12,6} = 0$, $E_{12,15} = 1$, $E_{12,18} = 1$, $E_{12,19} = 1$, $E_{12,20} = 0$, $E_{12,25} = 1$, $E_{12,26} = 1$, $E_{12,27} = 0$, $E_{12,28} = 0$, $E_{12,16} \neq E_{11,16}$	12
13	$E_{13,16} = 0$, $E_{13,17} = 0$, $E_{13,18} = 0$, $E_{13,19} = 0$, $E_{13,20} = 1$, $E_{13,27} = 0$, $E_{13,28} = 1$, $E_{13,13} \neq E_{13,0}$, $E_{13,14} = E_{13,0}$, $E_{13,6} = E_{13,1}$, $E_{13,14} \neq E_{13,1}$, $E_{13,15} \neq E_{13,1}$, $E_{13,15} = E_{13,2}$, $E_{13,29} = E_{13,2}$, $E_{13,21} \neq E_{13,3}$, $E_{13,30} \neq E_{13,3}$, $E_{13,22} \neq E_{13,4}$, $E_{13,31} \neq E_{13,4}$, $E_{13,23} \neq E_{13,5}$, $E_{13,24} \neq E_{13,6}$, $E_{13,25} = E_{13,7}$, $E_{13,13} \neq E_{13,8}$, $E_{13,14} = E_{13,9}$, $E_{13,30} = E_{13,11}$, $E_{13,31} \neq E_{13,12}$	25
14	$E_{14,16} = 0$, $E_{14,17} = 0$, $E_{14,18} = 0$, $E_{14,19} = 0$, $E_{14,20} = 0$, $E_{14,27} = 0$, $E_{14,28} = 0$	7
15	$E_{15,16} = 1$, $E_{15,17} = 1$, $E_{15,18} = 1$, $E_{15,19} = 1$, $E_{15,20} = 1$, $E_{15,27} = 1$, $E_{15,28} = 1$	7
17	$W_{17,16} = 0$, $W_{17,27} = 0$, $W_{17,14} \neq W_{4,15}$, $W_{17,4} = W_{17,2}$, $W_{17,29} \neq W_{17,20}$	5

Table 6. Starting point for a collision for 27 steps of SHA-256.

i	∇A_i	∇E_i	∇W_i
-4	----------	----------	
-3	----------	----------	
-2	----------	----------	
-1	----------	----------	
0	----------	----------	----------
1	----------	----------	----------
2	----------	----------	----------
3	----------	----------	----------
4	----------	----------	----------
5	----------	----------	----------
6	----------	----------	----------
7	??????????????????????????????	??????????????????????????????	??????????????????????????????x??
8	??????????????????????????????	??????????????????????????????	??????????????????????????????
9	??????????????????????????????	??????????????????????????????	----------
10	----------	??????????????????????????????	----------
11	----------	??????????????????????????????	----------
12	----------	??????????????????????????????	??????????????????????????????
13	----------	??????????????????????????????	----------
14	----------	----------	----------
15	----------	----------	??????????????????????????????
16	----------	----------	----------
17	----------	----------	??????????????????????????????
18	----------	----------	----------
19	----------	----------	----------
20	----------	----------	----------
21	----------	----------	----------
22	----------	----------	----------
23	----------	----------	----------
24	----------	----------	----------
25	----------	----------	----------
26	----------	----------	----------

Table 7. Characteristic for a collision for 27 steps of SHA-256.

i	∇A_i	∇E_i	∇W_i
-4	----------	----------	
-3	----------	----------	
-2	----------	----------	
-1	----------	----------	
0	----------	----------	----------
1	----------	----------	----------
2	----------	----------	----------
3	----------	----------	----------
4	----------	----------	----------
5	----------	----------1-------1------	----------
6	----------	-1--------0--0-10-1----0-0------	----------
7	-------unn--u-----n---nn-uuuu--	101-11---u10u1-0nuu-uuuu1n---n0-	00---1--un-0u-nuuuuu1-nu0n101n--
8	nnnnn-nnnn-------nuu----------	0n0n001001u-1u1n01un010n01n00110	-----u--n---n---------nn--------
9	----un--n--nu------nu-u-------	-1n1n1011u011100nn100u10-10000u-	----------
10	----------	u00000nuuu10uun01u00n00n110-u-u1	----------
11	----------	0n000uuuuu01010111n-uun01n000n01	----------
12	----------	01---1010u01u----111-010-0--110-	------110-u-------n0--u--n-n--nn
13	----------	01-10u1nuuuuu---1110-1nn11---01-	----------
14	----------	-----1-01011----------00--------	----------
15	----------	-----1-001000---------11--------	0u1-nn-n-u-1u---11un0uu10u101u0-
16	----------	----------	----------
17	----------	----------	---0-1nnn---u-1-----10uu0-------
18	----------	----------	----------
19	----------	----------	----------
20	----------	----------	----------
21	----------	----------	----------
22	----------	----------	----------
23	----------	----------	----------
24	----------	----------	----------
25	----------	----------	----------
26	----------	----------	----------

Table 8. Collision for 27 steps of SHA-256.

h_0	6a09e667 bb67ae85 3c6ef372 a54ff53a 510e527f 9b05688c 1f83d9ab 5be0cd19
m	725a0370 0daa9f1b 071d92df ec8282c1 7913134a bc2eb291 02d33a84 278dfd29 0c40f8ea d8bd68a0 0ce670c5 5ec7155d 9f6407a8 729fbfe8 aa7c7c08 607ae76d
m^*	725a0370 0daa9f1b 071d92df ec8282c1 7913134a bc2eb291 02d33a84 27460e6d 08c8fbea d8bd68a0 0ce670c5 5ec7155d 9f4425fb 729fbfe8 aa7c7c08 2d32d129
Δm	00000000 00000000 00000000 00000000 00000000 00000000 00000000 00cbf344 04880300 00000000 00000000 00000000 00202253 00000000 00000000 4d483644
h_1	5864015f 133494fa fa42bb35 94bc44f9 29eabb36 9e461e33 2eab27f8 106467c9

Table 9. Set of conditions for the collision for 27 steps.

i	set of conditions	
5	$E_{5,6} = 1$, $E_{5,15} = 1$	2
6	$A_{6,2} = A_{5,2}$, $A_{6,3} \neq A_{5,3}$, $A_{6,7} = A_{5,7}$, $A_{6,8} \neq A_{5,8}$, $A_{6,19} \neq A_{5,19}$, $A_{6,22} \neq A_{5,22}$, $A_{6,23} = A_{5,23}$, $E_{6,6} = 0$, $E_{6,8} = 0$, $E_{6,13} = 1$, $E_{6,15} = 0$, $E_{6,16} = 1$, $E_{6,18} = 0$, $E_{6,21} = 0$, $E_{6,30} = 1$, $E_{6,2} = E_{5,2}$, $E_{6,9} \neq E_{5,9}$, $E_{6,14} = E_{5,14}$	18
7	$A_{7,2} = 1$, $A_{7,3} = 1$, $A_{7,4} = 1$, $A_{7,5} = 1$, $A_{7,7} = 0$, $A_{7,8} = 0$, $A_{7,12} = 0$, $A_{7,19} = 1$, $A_{7,22} = 0$, $A_{7,23} = 0$, $A_{7,24} = 1$, $A_{7,10} \neq A_{7,1}$, $A_{7,11} \neq A_{6,11}$, $A_{7,25} \neq A_{6,25}$, $A_{7,31} \neq A_{7,10}$, $A_{7,31} \neq A_{7,11}$, $A_{7,25} = A_{7,14}$, $A_{7,26} = A_{7,15}$, $A_{7,27} = A_{7,16}$, $A_{7,28} = A_{7,16}$, $A_{7,28} = A_{7,17}$, $A_{7,29} = A_{7,17}$, $A_{7,31} \neq A_{7,20}$, $E_{7,1} = 0$, $E_{7,2} = 0$, $E_{7,6} = 0$, $E_{7,7} = 1$, $E_{7,8} = 1$, $E_{7,9} = 1$, $E_{7,10} = 1$, $E_{7,11} = 1$, $E_{7,13} = 1$, $E_{7,14} = 1$, $E_{7,15} = 0$, $E_{7,16} = 0$, $E_{7,18} = 1$, $E_{7,19} = 1$, $E_{7,20} = 0$, $E_{7,21} = 1$, $E_{7,22} = 1$, $E_{7,26} = 1$, $E_{7,27} = 1$, $E_{7,29} = 1$, $E_{7,30} = 0$, $E_{7,31} = 1$, $E_{7,5} = E_{6,5}$, $E_{7,12} = E_{6,12}$, $E_{7,28} = E_{6,28}$, $E_{7,5} = E_{7,0}$, $E_{7,23} = E_{7,4}$, $E_{7,28} \neq E_{7,23}$, $W_{7,2} = 0$, $W_{7,3} = 1$, $W_{7,4} = 0$, $W_{7,5} = 1$, $W_{7,6} = 0$, $W_{7,7} = 0$, $W_{7,8} = 1$, $W_{7,9} = 0$, $W_{7,11} = 1$, $W_{7,12} = 1$, $W_{7,13} = 1$, $W_{7,14} = 1$, $W_{7,15} = 1$, $W_{7,16} = 1$, $W_{7,17} = 0$, $W_{7,19} = 1$, $W_{7,20} = 0$, $W_{7,22} = 0$, $W_{7,23} = 1$, $W_{7,26} = 1$, $W_{7,30} = 0$, $W_{7,31} = 0$, $W_{7,21} \neq W_{7,0}$, $W_{7,18} \neq W_{7,1}$, $W_{7,29} \neq W_{7,1}$, $W_{7,21} \neq W_{7,10}$, $W_{7,25} = W_{7,10}$, $W_{7,29} = W_{7,18}$, $W_{7,29} = W_{7,25}$	80
8	$A_{8,11} = 1$, $A_{8,12} = 1$, $A_{8,13} = 0$, $A_{8,22} = 0$, $A_{8,23} = 0$, $A_{8,24} = 0$, $A_{8,25} = 0$, $A_{8,27} = 0$, $A_{8,28} = 0$, $A_{8,29} = 0$, $A_{8,30} = 0$, $A_{8,31} = 0$, $A_{8,10} \neq W_{12,10}$, $A_{8,14} \neq W_{12,14}$, $A_{8,26} \neq W_{12,26}$, $A_{8,19} \neq A_{6,19}$, $A_{8,10} \neq A_{7,10}$, $A_{8,20} = A_{7,20}$, $A_{8,26} \neq A_{7,26}$, $A_{8,10} = A_{8,1}$, $A_{8,16} \neq A_{8,4}$, $A_{8,17} = A_{8,5}$, $A_{8,15} \neq A_{8,6}$, $A_{8,18} = A_{8,6}$, $A_{8,18} \neq A_{8,7}$, $A_{8,19} = A_{8,7}$, $A_{8,20} \neq A_{8,8}$, $E_{8,0} = 0$, $E_{8,1} = 1$, $E_{8,2} = 1$, $E_{8,3} = 0$, $E_{8,4} = 0$, $E_{8,5} = 0$, $E_{8,6} = 1$, $E_{8,7} = 0$, $E_{8,8} = 0$, $E_{8,9} = 0$, $E_{8,10} = 1$, $E_{8,11} = 0$, $E_{8,12} = 0$, $E_{8,13} = 1$, $E_{8,14} = 1$, $E_{8,15} = 0$, $E_{8,16} = 0$, $E_{8,17} = 1$, $E_{8,18} = 1$, $E_{8,19} = 1$, $E_{8,21} = 1$, $E_{8,22} = 1$, $E_{8,23} = 0$, $E_{8,24} = 0$, $E_{8,25} = 1$, $E_{8,26} = 0$, $E_{8,27} = 0$, $E_{8,28} = 0$, $E_{8,29} = 0$, $E_{8,30} = 0$, $E_{8,31} = 0$, $W_{8,8} = 0$, $W_{8,9} = 0$, $W_{8,19} = 0$, $W_{8,23} = 0$, $W_{8,26} = 1$, $W_{8,20} \neq W_{8,5}$, $W_{8,22} = W_{8,5}$, $W_{8,27} = W_{8,6}$, $W_{8,15} = W_{8,11}$, $W_{8,24} \neq W_{8,13}$, $W_{8,30} \neq W_{8,15}$, $W_{8,29} = W_{8,25}$	70
9	$A_{9,8} = 1$, $A_{9,10} = 1$, $A_{9,11} = 0$, $A_{9,19} = 1$, $A_{9,20} = 0$, $A_{9,23} = 0$, $A_{9,26} = 0$, $A_{9,27} = 1$, $A_{9,13} \neq A_{7,13}$, $A_{9,4} = A_{8,4}$, $A_{9,7} \neq A_{8,7}$, $A_{9,12} \neq A_{9,0}$, $A_{9,22} \neq A_{9,1}$, $A_{9,15} \neq A_{9,3}$, $A_{9,16} \neq A_{9,4}$, $A_{9,15} \neq A_{9,6}$, $A_{9,18} \neq A_{9,7}$, $A_{9,30} \neq A_{9,7}$, $A_{9,29} \neq A_{9,9}$, $A_{9,30} \neq A_{9,21}$, $A_{9,31} \neq A_{9,22}$, $E_{9,1} = 1$, $E_{9,2} = 0$, $E_{9,3} = 0$, $E_{9,4} = 0$, $E_{9,5} = 0$, $E_{9,6} = 1$, $E_{9,8} = 0$, $E_{9,9} = 1$, $E_{9,10} = 1$, $E_{9,11} = 0$, $E_{9,12} = 0$, $E_{9,13} = 1$, $E_{9,14} = 0$, $E_{9,15} = 0$, $E_{9,16} = 0$, $E_{9,17} = 0$, $E_{9,18} = 1$, $E_{9,19} = 1$, $E_{9,20} = 1$, $E_{9,21} = 0$, $E_{9,22} = 1$, $E_{9,23} = 1$, $E_{9,24} = 1$, $E_{9,25} = 0$, $E_{9,26} = 1$, $E_{9,27} = 0$, $E_{9,28} = 1$, $E_{9,29} = 0$, $E_{9,30} = 1$	50
10	$A_{10,8} \neq A_{8,8}$, $A_{10,10} \neq A_{8,10}$, $A_{10,19} \neq A_{8,19}$, $A_{10,20} = A_{8,20}$, $A_{10,26} \neq A_{8,26}$, $A_{10,12} = A_{9,12}$, $A_{10,13} = A_{9,13}$, $A_{10,22} = A_{9,22}$, $A_{10,24} \neq A_{9,24}$, $A_{10,25} \neq A_{9,25}$, $A_{10,28} = A_{9,28}$, $A_{10,29} = A_{9,29}$, $A_{10,30} \neq A_{9,30}$, $A_{10,31} \neq A_{9,31}$, $E_{10,0} = 1$, $E_{10,1} = 1$, $E_{10,3} = 1$, $E_{10,5} = 0$, $E_{10,6} = 1$, $E_{10,7} = 1$, $E_{10,8} = 0$, $E_{10,9} = 0$, $E_{10,10} = 0$, $E_{10,11} = 0$, $E_{10,12} = 0$, $E_{10,13} = 0$, $E_{10,14} = 1$, $E_{10,15} = 1$, $E_{10,16} = 0$, $E_{10,17} = 0$, $E_{10,18} = 1$, $E_{10,19} = 1$, $E_{10,20} = 0$, $E_{10,21} = 1$, $E_{10,22} = 1$, $E_{10,23} = 1$, $E_{10,24} = 1$, $E_{10,25} = 0$, $E_{10,26} = 0$, $E_{10,27} = 0$, $E_{10,28} = 0$, $E_{10,29} = 0$, $E_{10,30} = 0$, $E_{10,31} = 1$	44
11	$A_{11,8} = A_{10,8}$, $A_{11,10} = A_{10,10}$, $A_{11,11} \neq A_{10,11}$, $A_{11,19} \neq A_{10,19}$, $A_{11,20} \neq A_{10,20}$, $A_{11,23} = A_{10,23}$, $A_{11,26} \neq A_{10,26}$, $A_{11,27} \neq A_{10,27}$, $E_{11,0} = 1$, $E_{11,1} = 0$, $E_{11,2} = 0$, $E_{11,3} = 0$, $E_{11,4} = 0$, $E_{11,5} = 0$, $E_{11,6} = 0$, $E_{11,7} = 1$, $E_{11,8} = 0$, $E_{11,9} = 0$, $E_{11,10} = 1$, $E_{11,11} = 1$, $E_{11,13} = 0$, $E_{11,14} = 1$, $E_{11,15} = 1$, $E_{11,16} = 1$, $E_{11,17} = 0$, $E_{11,18} = 1$, $E_{11,19} = 0$, $E_{11,20} = 1$, $E_{11,21} = 0$, $E_{11,22} = 1$, $E_{11,23} = 1$, $E_{11,24} = 1$, $E_{11,25} = 1$, $E_{11,26} = 1$, $E_{11,27} = 0$, $E_{11,28} = 0$, $E_{11,29} = 0$, $E_{11,30} = 0$, $E_{11,31} = 0$	39
12	$E_{12,1} = 0$, $E_{12,2} = 1$, $E_{12,3} = 1$, $E_{12,6} = 0$, $E_{12,8} = 0$, $E_{12,9} = 1$, $E_{12,10} = 0$, $E_{12,12} = 1$, $E_{12,13} = 1$, $E_{12,14} = 1$, $E_{12,19} = 1$, $E_{12,20} = 1$, $E_{12,21} = 0$, $E_{12,22} = 1$, $E_{12,23} = 0$, $E_{12,24} = 1$, $E_{12,25} = 0$, $E_{12,26} = 1$, $E_{12,30} = 1$, $E_{12,31} = 0$, $E_{12,11} = W_{12,11}$, $E_{12,27} \neq W_{12,27}$, $E_{12,0} \neq A_{8,0}$, $E_{12,5} = E_{12,0}$, $W_{12,0} = 0$, $W_{12,1} = 0$, $W_{12,4} = 0$, $W_{12,6} = 0$, $W_{12,9} = 1$, $W_{12,12} = 0$, $W_{12,13} = 0$, $W_{12,21} = 1$, $W_{12,23} = 0$, $W_{12,24} = 1$, $W_{12,25} = 1$	35
13	$E_{13,1} = 1$, $E_{13,2} = 0$, $E_{13,6} = 1$, $E_{13,7} = 1$, $E_{13,8} = 0$, $E_{13,9} = 0$, $E_{13,10} = 1$, $E_{13,12} = 0$, $E_{13,13} = 1$, $E_{13,14} = 1$, $E_{13,15} = 1$, $E_{13,19} = 1$, $E_{13,20} = 1$, $E_{13,21} = 1$, $E_{13,22} = 0$, $E_{13,23} = 1$, $E_{13,24} = 0$, $E_{13,25} = 1$, $E_{13,26} = 1$, $E_{13,27} = 0$, $E_{13,28} = 1$, $E_{13,30} = 1$, $E_{13,31} = 0$, $E_{13,5} = E_{13,0}$, $E_{13,17} = E_{13,4}$, $E_{13,18} = E_{13,5}$, $E_{13,29} = E_{13,11}$	27
14	$E_{14,8} = 0$, $E_{14,9} = 0$, $E_{14,20} = 1$, $E_{14,21} = 1$, $E_{14,22} = 0$, $E_{14,23} = 1$, $E_{14,24} = 0$, $E_{14,26} = 1$	8
15	$E_{15,8} = 1$, $E_{15,9} = 1$, $E_{15,19} = 0$, $E_{15,20} = 0$, $E_{15,21} = 0$, $E_{15,22} = 1$, $E_{15,23} = 0$, $E_{15,24} = 0$, $E_{15,26} = 1$, $W_{15,1} = 0$, $W_{15,2} = 1$, $W_{15,3} = 1$, $W_{15,4} = 0$, $W_{15,5} = 1$, $W_{15,6} = 1$, $W_{15,7} = 0$, $W_{15,8} = 1$, $W_{15,9} = 1$, $W_{15,10} = 1$, $W_{15,11} = 0$, $W_{15,12} = 0$, $W_{15,13} = 1$, $W_{15,14} = 1$, $W_{15,15} = 1$, $W_{15,19} = 1$, $W_{15,20} = 1$, $W_{15,22} = 1$, $W_{15,24} = 0$, $W_{15,26} = 0$, $W_{15,27} = 0$, $W_{15,29} = 1$, $W_{15,30} = 1$, $W_{15,31} = 0$, $W_{15,23} \neq W_{15,0}$, $W_{15,25} \neq W_{15,0}$, $W_{15,25} = W_{15,18}$, $W_{15,28} \neq W_{15,21}$	37
17	$W_{17,7} = 0$, $W_{17,8} = 1$, $W_{17,9} = 1$, $W_{17,10} = 0$, $W_{17,11} = 1$, $W_{17,17} = 1$, $W_{17,19} = 1$, $W_{17,23} = 0$, $W_{17,24} = 0$, $W_{17,25} = 0$, $W_{17,26} = 1$, $W_{17,28} = 0$, $W_{17,2} \neq W_{17,0}$, $W_{17,30} = W_{17,0}$, $W_{17,31} = W_{17,1}$, $W_{17,31} = W_{17,6}$, $W_{17,21} = W_{17,12}$, $W_{17,21} = W_{17,14}$, $W_{17,22} = W_{17,15}$, $W_{17,27} \neq W_{17,18}$	20