New collision attacks on SHA-1 based on optimal joint local-collision analysis

Marc Stevens

CWI, Amsterdam, The Netherlands
marc@marc-stevens.nl

Abstract. The main contributions of this paper are two-fold.

Firstly, we present a novel direction in the cryptanalysis of the cryptographic hash function SHA-1. Our work builds on previous cryptanalytic efforts on SHA-1 based on combinations of local collisions. Due to dependencies, previous approaches used heuristic corrections when combining the success probabilities and message conditions of the individual local collisions. Although this leads to success probabilities that are seemingly sufficient for feasible collision attacks, this approach most often does not lead to the maximum success probability possible as desired. We introduce novel techniques that enable us to determine the theoretical maximum success probability for a given set of (dependent) local collisions, as well as the smallest set of message conditions that attains this probability. We apply our new techniques and present an implemented open-source near-collision attack on SHA-1 with a complexity equivalent to $2^{57.5}$ SHA-1 compressions.

Secondly, we present an identical-prefix collision attack and a chosen-prefix collision attack on SHA-1 with complexities equivalent to approximately 2^{61} and $2^{77.1}$ SHA-1 compressions, respectively.

1 Introduction

A series of breakthrough attacks on hash functions started in 2004 when the first collisions for MD4, MD5, HAVAL-128 and RIPEMD were presented by Wang et al.[WFLY04, WY05]. This was soon followed by the first SHA-0 collision by Biham et al. [BCJ+05]. Soon thereafter, Wang et al. published a more efficient collision attack on SHA-0 [WYY05c]. In the same year, the first collision attack on full SHA-1 [WYY05b] was presented by Wang et al. with an estimated complexity of 2^{69} compressions. A later unpublished result by Wang et al. claimed an attack with a complexity of 2^{63} compressions [WYY05a] which was later partly verified by Cochran [Coc07]. This was further improved by Mendel et al. with an unpublished attack with a complexity of $2^{60.x}$ compressions [MRR07]. Although later withdrawn, McDonald et al. published an attack with claimed complexity of 2^{52} compressions [MHP09]. Rafi Chen claims to be able to find collisions in

2^{58} [Che11] [1]. For reduced step variants of SHA-1 more progress has been made [CR06, CMR07, Gre10] and collisions have been found for up to 75 steps [GA11].

So far, it seems some kind of barrier has been reached at around 2^{61} SHA-1 compressions. Unfortunately, as Polk et al. [PCTH11] point out, these cryptanalytic advancements are not fully reflected in the literature so far.

2 Our contributions

This paper aims to renew the cryptanalytic efforts to construct a feasible collision attack on SHA-1 and find an actual collision pair. The main contributions of this paper are two-fold.

Firstly, we present a novel direction in the cryptanalysis of SHA-1 that we believe will allow collision attacks with complexity well below the 2^{61} barrier. Collision attacks on SHA-1 are constructed in roughly two parts: a non-linear part (over approximately the first 20 steps) and a linear part (over approximately the last 60 steps). The linear part is constructed using a linear combination of local collisions as described by a disturbance vector [CJ98]. So far, to obtain the success probability of these combinations, the local collisions are first studied independently (e.g., see [MPRR06]) and then combined. As the success probabilities of local collisions can be dependent (e.g., see [Man11]), current approaches make some heuristic corrections when joining probabilities and message conditions. Although this is seemly sufficient to construct feasible collision attack on SHA-1, it may not lead to the desired maximum success probability possible and thereby leads to sub-optimal collision attacks. We introduce novel techniques that enable the computation of the maximum success probability for a given set of (dependent) local collisions, as well as the smallest set of message conditions that attains this probability. That our new approach provides a distinct advantage over the previous approach is showcased in our second contribution.

Our second contribution is an implemented near-collision attack for SHA-1 with a complexity equivalent to $2^{57.5}$ compressions[2]. We show how this near-collision attack can be used to construct a SHA-1 identical-prefix collision attack with a complexity of 2^{61} compressions. Furthermore, we present the first SHA-1 chosen-prefix collision attack with a complexity of $2^{77.1}$ compressions.

Our attack distinguishes itself from previous claims on several aspects. Firstly, we aimed to optimize the complexity over the linear part and (so far) not over the non-linear part. Secondly, our novel direction has resulted in a competitive attack complexity without exploiting nearly all degrees of freedoms. In fact there are well over 50 from the 512 message bits left as degrees of freedom that can be further exploited in future work. Lastly, it is the first public implementation of a SHA-1

[1] We like to note that using our methods we have proven that the highest probability attainable over the last 8 steps is $2^{-8.356}$. But Chen (see Ch. 9.5) actually uses a factor $\frac{100}{3000} \approx 2^{-4.9}$, suggesting his claim may be a factor $2^{3.5}$ too optimistic.

[2] This complexity is not based on a purely theoretical cost analysis, but directly determined from the measured performance over the non-linear part and the (implementation verified) theoretical success probabilities over the linear part, see Sect. 5.1.

collision attack: the source code is available online [Ste12b]. This allows the public verification of the correctness and the complexity of our implementation and we also hope it leads to better understanding and improvements by the scientific community. Due to space considerations, many technical details have been omitted here, but these can be found in [Ste12a]. Despite this, we briefly discuss how the correctness of our implementation as well as our claimed complexity can be verified using our publicly available source code.

3 Preliminaries

Notation. SHA-1 is defined using 32-bit words $X = (x_i)_{i=0}^{31} \in \{0,1\}^{32}$ that are identified with elements $X = \sum_{i=0}^{31} x_i 2^i$ of $\mathbb{Z}/2^{32}\mathbb{Z}$ (for addition and subtraction). A *binary signed digit representation* (BSDR) for $X \in \mathbb{Z}/2^{32}\mathbb{Z}$ is a sequence $Z = (z_i)_{i=0}^{31} \in \{-1,0,1\}^{32}$ for which $X = \sum_{i=0}^{31} z_i 2^i$. We use the following notation: $Z[i] = z_i$, $RL(Z,n)$ and $RR(Z,n)$ (cyclic left and right rotation), $w(Z)$ (Hamming weight), $\sigma(Z) = X = \sum_{i=0}^{31} k_i 2^i \in \mathbb{Z}/2^{32}\mathbb{Z}$.

In collision attacks we consider two related messages M and M'. For any variable X related to the SHA-1 calculation of M, we use X' to denote the corresponding variable for M'. Furthermore, for such a 'matched' variable $X \in \mathbb{Z}/2^{32}\mathbb{Z}$ we define $\delta X = X' - X$ and $\Delta X = (X'[i] - X[i])_{i=0}^{31}$.

SHA-1's compression function. The input for SHA-1's Compress consists of an intermediate hash value $IHV_{\text{in}} = (a,b,c,d,e)$ of five 32-bit words and a 512-bit message block B. The 512-bit message block B is partitioned into 16 consecutive 32-bit strings which are interpreted as 32-bit words $W_0, W_1, \ldots, W_{15}$ (using big-endian), and expanded to $W_0, \ldots, W_{79}$ as follows:

$$W_t = RL(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}, 1), \quad \text{for } 16 \leq t < 80. \tag{1}$$

We describe SHA-1's compression function Compress in an 'unrolled' version. For each step $t = 0, \ldots, 79$ it uses a working state consisting of five 32-bit words Q_t, Q_{t-1}, Q_{t-2}, Q_{t-3} and Q_{t-4} and calculates a new state word Q_{t+1}. The working state is initialized before the first step as

$$(Q_0, Q_{-1}, Q_{-2}, Q_{-3}, Q_{-4}) = (a, b, RR(c,30), RR(d,30), RR(e,30)).$$

For $t = 0, 1, \ldots, 79$ in succession, Q_{t+1} is calculated as follows:

$$\begin{aligned} F_t &= f_t(Q_{t-1}, RL(Q_{t-2},30), RL(Q_{t-3},30)), \\ Q_{t+1} &= F_t + AC_t + W_t + RL(Q_t,5) + RL(Q_{t-4},30). \end{aligned} \tag{2}$$

These 80 steps are grouped in 4 rounds of 20 steps each. Here, AC_t is the constant $\texttt{5a827999}_{16}$, $\texttt{6ed9eba1}_{16}$, $\texttt{8f1bbcdc}_{16}$ or $\texttt{ca62c1d6}_{16}$ for the 1st, 2nd, 3rd and 4th round, respectively. The non-linear function $f_t(X,Y,Z)$ is defined as $(X \wedge Y) \oplus (\overline{X} \wedge Z)$, $X \oplus Y \oplus Z$, $(X \wedge Y) \vee (Z \wedge (X \vee Y))$ or $X \oplus Y \oplus Z$ for the 1st, 2nd, 3rd and 4th round, respectively. Finally, the output intermediate hash value δIHV_{out} is determined as:

$$\delta IHV_{\text{out}} = (a{+}Q_{80},\ b{+}Q_{79},\ c{+}RL(Q_{78},30),\ d{+}RL(Q_{77},30),\ e{+}RL(Q_{76},30)).$$

4 Joint local-collision analysis

4.1 Local collisions and the disturbance vector

In 1998, Chabaud and Joux [CJ98] constructed a collision attack on SHA-0 based on local collisions. A local collision over 6 steps for SHA-0 and SHA-1 consists of a disturbance $\delta Q_{t+1} = 2^b$ created in some step t by a message word bit difference $\delta W_t = 2^b$. This disturbance is corrected over the next five steps, so that after those five steps no differences occur in the five working state words. They were able to interleave many of these local collisions such that the message word differences $(\Delta W_t)_{t=0}^{79}$ conform to the message expansion (cf. Eq. 1). For more convenient analysis, they consider the *disturbance vector* which is a non-zero vector $(DV_t)_{t=0}^{79}$ conform to the message expansion where every '1'-bit $DV_t[b]$ marks the start of a local collision based on the disturbance $\delta W_t[b] = \pm 1$. We denote by $(DW_t)_{t=0}^{79}$ the message word bit differences without sign (i.e., $DW_t = W'_t \oplus W_t$) for a disturbance vector $(DV_t)_{t=0}^{79}$:

$$DW_t := \bigoplus_{(i,r)\in\mathcal{R}} RL(DV_{t-i}, r), \quad \mathcal{R} = \{(0,0),(1,5),(2,0),(3,30),(4,30),(5,30)\}$$

Note that in differential paths we work with differences δW_t instead of DW_t. We say that a message word difference δW_t is *compatible* with DW_t if there are coefficients $c_0, \ldots, c_{31} \in \{-1, 1\}$ such that $\delta W_t = \sum_{j=0}^{31} c_j \cdot DW_t[j]$. The set $\mathcal{W}_t$ of all compatible message word differences given DW_t is defined as:

$$\mathcal{W}_t := \left\{\sigma(X) \;\middle|\; \text{BSDR } X, \;\; X[i] \in \{-DW_t[i], +DW_t[i]\}, \;\; i \in \{0, \ldots, 31\}\right\}$$

4.2 Disturbance vector classes

Manuel [Man11] has classified previously found interesting disturbance vectors into two classes. A disturbance vector from the first class denoted by $\mathrm{I}(K, b)$ is defined by $DV_K = \ldots = DV_{K+14} = 0$ and $DV_{K+15} = 2^b$. Similarly, a disturbance vector from the second class denoted by $\mathrm{II}(K, b)$ is defined by $DV_{K+1} = DV_{K+3} = RL(2^{31}, b)$ and $DV_{K+15} = 2^b$ and $DV_{K+i} = 0$ for $i \in \{0, 2, 4, 5, \ldots, 14\}$. For both classes, the remaining $DV_0, \ldots, DV_{K-1}$ and $DV_{K+16}, \ldots, DV_{79}$ are determined through the (reverse) message expansion relation (Eq. 1).

4.3 Dependencies of local collisions

Local collisions can interact in the following three ways.

- *Message differences.* Firstly, they may use message word differences in the same bit position of the same message word. E.g., consider the disturbance vector for which $DV_{50}[0]$ and $DV_{55}[30]$ are the only '1'-bits. Then as $DW_{55} = DV_{55} \oplus RL(DV_{50}, 30) = 0$, this means the message word differences in step 55 of the two local collisions must be chosen to cancel each other.

– *Working state differences.* Secondly, local collisions starting in the same step directly interact with each other due to carries. E.g., Wang et al. [WYY05b] introduced a bit compression technique. They use opposite signs for two local collisions that start in the same step at two subsequent bit positions (say $DV_{25}[0] = DV_{25}[1] = 1$) to turn it into a single local collision.
– *Boolean function differences.* Thirdly, two 'close' disturbances can interact in the boolean function. E.g., consider the disturbance vector for which $DV_{25}[31]$ and $DV_{26}[31]$ are the only '1'-bits. Then these local collisions interact as in the first case as the message word differences in steps 29 and 30 cancel each other out. Moreover, in step 29 it is also guaranteed that $\delta F_{29} = 0$ as the two disturbances input to the XOR boolean function cancel each other. In contrast, when analyzing these two local collisions independently, each has a probability of 0.5 that the difference δF_{29} has the opposite sign from δW_{29}. The product of the independent success probabilities is thereby *lower* than the maximum joint probability of these two local collisions by a factor $0.5 \cdot 0.5 = 0.25$ (see also [Man11, Table 9]). This particular example does not involve any carries, which in other cases may have a further impact on the maximum success probability.

Although these examples are quite easy to analyze, disturbance vectors have a higher density of disturbances at the beginning and the end. For these higher density areas, it is significantly more difficult to analyze the exact impact of these interactions on the maximum success probability. In this paper we take a new direction in the cryptanalysis of SHA-1 in which we do not analyze these interactions directly, but use a rather general approach to determine the maximum success probability that incorporates these interactions.

4.4 Optimal joint local-collision analysis

We start at the relatively easy and well understood analysis of a single local collision. Given the single bit disturbance $\Delta Q_{t+1}[b] = \pm 1$ created in the first step t, one analyzes the necessary message conditions to cancel this disturbance in the subsequent steps. Most importantly, one determines what the probability is of a successful cancellation under these message conditions. Higher success probabilities are obtained by also considering carries in ΔQ_{t+1} from bit position b to higher positions.

One approach that obtains exact success probabilities is to sum the exact success probabilities of *all* possible differential paths over these 6 steps $t, \ldots, t+5$ with $\delta Q_{t-4} = \ldots = \delta Q_t = 0$, $\delta Q_{t+1} \neq 0$ and $\delta Q_{t+2} = \ldots = \delta Q_{t+6} = 0$ using a given message difference vector $(\delta W_i)_{i=t}^{t+5}$. Although there are quite a few of such differential paths for a single local collision, these can easily be enumerated.

We propose to study combinations of local collisions in a very similar way. That is, we propose to analyze the set of *all* possible differential paths over a given range of steps $t_b, \ldots, t_e$ that contain disturbances as prescribed by the disturbance vector using message word differences δW_t compatible with DW_t. Next, this set is partitioned based on the values for the starting and ending working state

differences and the message word differences. We distinguish thus only on the pre-conditions (the differences in the starting working state and the message words) and the post-condition (the differences in the ending working state) of differential paths that matches how they are used in an actual collision attack. For each partition, we compute the sum of the probabilities of its differential paths. One can thus interpret this total partition probability as the total probability that the ending working state differences are obtained after step t_e given that both the differences in the starting working state at step t_b and the differences in the message words for steps $t_b, \ldots, t_e$ hold. Hence, the desired maximum success probability is the maximum over all total partition probabilities.

4.5 Definitions

More formally, we define a differential path $\mathcal{P}$ over steps $t = t_b, \ldots, t_e$ to be given as $\mathcal{P} = ((\Delta Q_t)_{t=t_b-4}^{t_e+1}, (\Delta F_t)_{t=t_b}^{t_e}, (\delta W_t)_{t=t_b}^{t_e})$[3], under the following restrictions:

- correct differential steps for $t = t_b, \ldots, t_e$:

$$\sigma(\Delta Q_{t+1}) = \sigma(RL(\Delta Q_t, 5)) + \sigma(RL(\Delta Q_{t-4}, 30)) + \sigma(\Delta F_t) + \delta W_t. \quad (3)$$

- $\Delta F_t[31] \in \{0, 1\}$ and a non-zero value represents $\Delta F_t[31] \in \{-1, +1\}$. [4]

The success probability $\Pr[\mathcal{P}]$ of a differential path $\mathcal{P}$ over steps $t_b, \ldots, t_e$ is informally defined as the probability that the given path $\mathcal{P}$ holds exactly for $(\widehat{Q}_{t_b-4}, \widehat{Q}'_{t_b-4})$, …, $(\widehat{Q}_{t_e+1}, \widehat{Q}'_{t_e+1})$ for uniformly-randomly chosen $\widehat{Q}_{t_b-4}, \ldots, \widehat{Q}_{t_b}$ and $\widehat{W}_{t_b}, \ldots, \widehat{W}_{t_e}$. The $\widehat{Q}'_{t_b-4}, \ldots, \widehat{Q}'_{t_b}$ and $\widehat{W}'_{t_b}, \ldots, \widehat{W}'_{t_e}$ are determined through the first five working state differences $\delta Q_{t_b}, \ldots, \delta Q_{t_e}$ and the message differences δW_i (for $i = t_b, \ldots, t_e$). The remaining $(\widehat{Q}_{t_b+1}, \widehat{Q}'_{t_b+1}), \ldots, (\widehat{Q}_{t_e+1}, \widehat{Q}'_{t_e+1})$ are computed using the step function (Eq. 2). We refer to [Ste12a, Ch. 7.5] for an equivalent definition and how to efficiently determine the probability $\Pr[\mathcal{P}]$.

As we are interested in differential paths with prescribed disturbances, we define the set $\mathcal{Q}_t$ as the set of all allowed differences ΔQ_t given $(DV_i)_{i=0}^{79}$:

$$\mathcal{Q}_t := \left\{ \text{BSDR } Y \;\middle|\; \begin{smallmatrix} \sigma(Y)=\sigma(Z), \\ Z[i]\in\{-DV_{t-1}[i], DV_{t-1}[i]\},\ i=0,\ldots,31 \end{smallmatrix} \right\}.$$

We are now ready to define the set of *all* possible differential paths over steps $t_b, \ldots, t_e$ that we will base our analysis on:

$$\mathcal{D}_{[t_b,t_e]} := \left\{ \widehat{\mathcal{P}} \;\middle|\; \Delta\widehat{Q}_i \in \mathcal{Q}_i,\ \delta\widehat{W}_j \in \mathcal{W}_j,\ \Pr[\widehat{\mathcal{P}}] > 0 \right\}$$

We define three functions ψ, ϕ and ω that return beginning working state differences, ending working state differences and message word differences:

[3] In practice, we use a strictly smaller representation wherein ΔQ_{t_b-4} and δQ_{t_e+1} are replaced by $\delta(RL(Q_{t-4}, 30))$ and δQ_{t_e+1}, respectively. We use a simplification here to ease presentation.

[4] Here both -1 and $+1$ result in the same contribution in $\sigma(\Delta F_t)$.

$$\psi(\mathcal{P}) = (\Delta Q_i)_{i=t_b-4}^{t_b}, \quad \omega(\mathcal{P}) = (\delta W_i)_{i=t_b}^{t_e},$$
$$\phi(\mathcal{P}) = (d_i)_{i=t_e-3}^{t_e+1}, \text{ where } d_i = \begin{cases} \sigma(RL(\Delta Q_i, 30)), & i = t_e-3, t_e-2, t_e-1; \\ \delta Q_i, & i = t_e, t_e+1. \end{cases}$$

We have chosen this particular definition for the ending working state differences $\phi(\mathcal{P})$ as this matches δIHV_{out} exactly. We denote by $\psi(\mathcal{D})$, $\phi(\mathcal{D})$ and $\omega(\mathcal{D})$ the sets found by applying ψ, ϕ or ω to all differential paths in the set $\mathcal{D}$.

For a given disturbance vector $(DV_t)_{t=0}^{79}$, the desired maximum success probability over steps $t_b, \ldots, t_e$ denoted by $\text{FDC}_{[t_b,t_e]}\left((DV_t)_{t=0}^{79}\right)$ is:

$$\text{FDC}_{[t_b,t_e]}\left((DV_t)_{t=0}^{79}\right) = \max_{b,e,w} \sum_{\substack{\widehat{\mathcal{P}} \in \mathcal{D}_{[t_b,t_e]} \\ \psi(\widehat{\mathcal{P}})=b,\ \phi(\widehat{\mathcal{P}})=e,\ \omega(\widehat{\mathcal{P}})=w}} \Pr[\widehat{\mathcal{P}}] \cdot c(b),$$

where $c(b) = c((\Delta Q_i)_{i=t_b-4}^{t_e})$ is the correction factor $c(b) = \prod_{i=t_b-4}^{t_b-2} 2^{w(\Delta \widehat{Q}_i)}$. This correction factor $c(b)$ ensures that FDC is the maximum success probability assuming all working state bit conditions are fulfilled for Q_{t_b-4}, Q_{t_b-3} and Q_{t_b-2}.[5] This is due to the fact that a collision attack fulfills working state bit conditions step by step, using message freedoms to speed up the attack, until these freedoms cannot be exploited anymore. At that point, it is more beneficial to compute all remaining steps and verify whether the desired δIHV_{out} is obtained. FDC returns the maximum success probability obtainable for these remaining steps.

4.6 Differential path reduction

Unfortunately, analyzing a single local collision in the above manner is very feasible, whereas analyzing multiple local collisions quickly results in a prohibitively large set of possible differential paths. We exploit the large amount of redundancy among the possible differential paths to be able to efficiently compute the desired maximum success probability even when there are many local collisions.

Note that we are only interested in the total success probability for given pre- and post-conditions and not in the differential paths themselves per se. We therefore propose to break up a differential path $\mathcal{P}$ into two valid differential paths $\widehat{\mathcal{P}}$ and $\widetilde{\mathcal{P}}$ with the following properties:

- $\widehat{\mathcal{P}}$ and $\widetilde{\mathcal{P}}$ are 'disjoint' and 'add' to $\mathcal{P}$. More specifically, we want that either $\Delta \widehat{Q}_i[b]$ or $\Delta \widetilde{Q}_i[b]$ to be equal to $\Delta Q_i[b]$ and the other to be zero (or all three to be zero). The same holds for $\Delta F_i[b]$. Furthermore, $\delta W_i = \delta \widehat{W}_i + \delta \widetilde{W}_i$;
- the success probabilities of $\widehat{\mathcal{P}}$ and $\widetilde{\mathcal{P}}$ are independent: $\Pr[\mathcal{P}] = \Pr[\widehat{\mathcal{P}}] \cdot \Pr[\widetilde{\mathcal{P}}]$;
- $\psi(\mathcal{P}) = \psi(\widehat{\mathcal{P}})$ and $\phi(\mathcal{P}) = \phi(\widehat{\mathcal{P}})$;
- the success probability $\Pr[\widehat{\mathcal{P}}]$ is maximal under the above restraints.

[5] Note that if bit conditions up to Q_{t_b-2} are fulfilled then ΔF_{t_b-1} has been ensured, but not ΔF_{t_b}.

Algorithm 4-1 Iterative construction of reduced differential path sets

1. Let $\widehat{t}$ be some step in the range $[t_b, t_e]$.
2. Construct the entire set $\mathcal{D}_{[\widehat{t},\widehat{t}]}$ of all possible differential paths over step $\widehat{t}$.
3. Compute $\mathcal{R}_{[\widehat{t},\widehat{t}]} = \{\text{Reduce}(\mathcal{P}) \mid \mathcal{P} \in \mathcal{D}_{[\widehat{t},\widehat{t}]}\}$.
4. For $i = \widehat{t}, \widehat{t}+1, \ldots, t_e - 1$, using the set $\mathcal{R}_{[\widehat{t},i]}$ we compute: $\mathcal{R}_{[\widehat{t},i+1]}$:
   (a) Let $A := \emptyset$.
   (b) For all $\mathcal{P} \in \mathcal{R}_{[\widehat{t},i]}$ and for all choices $\Delta Q_{i+2} \in \mathcal{Q}_{i+2}$, $\delta W_{i+1} \in \mathcal{W}_{i+1}$, $\Delta F_{i+1} \in \{-1, 0, 1\}^{31} \times \{0, 1\}$ let $\widehat{\mathcal{P}}$ be the differential path over steps $\widehat{t}, \ldots, i+1$ given as $\mathcal{P}$ appended with ΔQ_{i+2}, ΔF_{i+1} and δW_{i+1}.
   If $\Pr[\widehat{\mathcal{P}}] > 0$ then let $A := A \cup \{\text{Reduce}(\widehat{\mathcal{P}})\}$.
   (c) $\mathcal{R}_{[\widehat{t},i+1]} := A$.
5. For $i = \widehat{t}, \widehat{t}-1, \ldots, t_b + 1$, using the set $\mathcal{R}_{[i,t_e]}$ we compute $\mathcal{R}_{[i-1,t_e]}$:
   (a) Let $A := \emptyset$.
   (b) For all $\mathcal{P} \in \mathcal{R}_{[i,t_e]}$ and for all choices $\Delta Q_{i-5} \in \mathcal{Q}_{i-5}$, $\delta W_{i-1} \in \mathcal{W}_{i-1}$, $\Delta F_{i-1} \in \{-1, 0, 1\}^{31} \times \{0, 1\}$ let $\widehat{\mathcal{P}}$ be the differential path over steps $i-1, \ldots, t_e$ given as $\mathcal{P}$ prepended with ΔQ_{i-5}, ΔF_{i-1} and δW_{i-1}.
   If $\Pr[\widehat{\mathcal{P}}] > 0$ then let $A := A \cup \{\text{Reduce}(\widehat{\mathcal{P}})\}$.
   (c) $\mathcal{R}_{[i-1,t_e]} := A$.
6. Output $\mathcal{R}_{[t_b,t_e]}$.

One can interpret $\widehat{\mathcal{P}}$ as the differential path $\mathcal{P}$ with all differences removed that do not interact with the differences that constitute the starting and ending working state differences $\psi(\mathcal{P})$ and $\phi(\mathcal{P})$. We denote $\widehat{\mathcal{P}}$ as $\text{Reduce}(\mathcal{P})$ and $\widetilde{\mathcal{P}}$ as $\mathcal{P} - \widehat{\mathcal{P}}$. In our proposed methodology, instead of directly computing the differential paths in $\mathcal{D}_{[t_b,t_e]}$ and their probabilities, we propose to work with the set of reduced differential paths $\mathcal{R}_{[t_b,t_e]} := \{\text{Reduce}(\mathcal{P}) \mid \mathcal{P} \in \mathcal{D}_{[t_b,t_e]}\}$ and cumulative probabilities $p_{(\mathcal{P},w)}$ for each reduced differential path $\mathcal{P}$ and w defined as:

$$p_{(\mathcal{P},w)} = \sum_{\substack{\mathcal{P}' \in \mathcal{D}_{[t_b,t_e]} \\ \mathcal{P}=\text{Reduce}(\mathcal{P}'), w=\omega(\mathcal{P}')}} \Pr[\mathcal{P}' - \mathcal{P}].$$

These cumulative probabilities have an easy interpretation using the equation:

$$\Pr[\mathcal{P}] \cdot p_{(\mathcal{P},w)} = \sum_{\substack{\mathcal{P}' \in \mathcal{D}_{[t_b,t_e]} \\ \mathcal{P}=\text{Reduce}(\mathcal{P}'), w=\omega(\mathcal{P}')}} \Pr[\mathcal{P}] \cdot \Pr[\mathcal{P}' - \mathcal{P}] = \sum_{\substack{\mathcal{P}' \in \mathcal{D}_{[t_b,t_e]} \\ \mathcal{P}=\text{Reduce}(\mathcal{P}'), w=\omega(\mathcal{P}')}} \Pr[\mathcal{P}']$$

As the working state differences $\phi(\mathcal{P})$ and $\psi(\mathcal{P})$ are unaffected by $\text{Reduce}(\mathcal{P})$, the set of reduced differential paths and the cumulative probabilities are sufficient to determine the total success probability of any partition (b, e, w) of $\mathcal{D}_{[20,79]}$.

Moreover, the set $\mathcal{R}_{[t_b,t_e]}$ of reduced differential paths can be computed efficiently in an iterative manner as shown in Alg. 4-1. The cumulative probabilities can also be computed iteratively, but unfortunately the number of possible message difference vectors $w \in (\mathcal{W}_i)_{i=t_b}^{t_e}$ still grows exponentially in the number of local collisions over these steps.

Message difference vector classes To solve the problem of the exponential growth of possible message difference vectors, we consider classes $\overline{w}$ of message difference vectors w over steps $i, \ldots, j$, where any two $w \neq w'$ are in the same class $\overline{w}$ if and only if $p_{(\mathcal{P},w)} = p_{(\mathcal{P},w')}$ for all $\mathcal{P} \in \mathcal{R}_{[i,j]}$. It then suffices to compute the cumulative probabilities for only one representative $w \in \overline{w}$ for each class $\overline{w}$ over steps $t_b, \ldots, t_e$.

Let $\overline{W}_{[i,j]}$ be the set of all message difference vector classes $\overline{w}$ over steps $i, \ldots, j$. An important insight is that for any class $\overline{w}_{[i,j]} \in \overline{W}_{[i,j]}$ and any two $w, w' \in \overline{w}_{[i,j]}$ it holds that the extensions $w||\delta W_{j+1}$ and $w'||\delta W_{j+1}$ of w and w' with a difference δW_{j+1} are both in the same class $\overline{w}_{[i,j+1]} \in \overline{W}_{[i,j+1]}$. An analogous statement holds for prepending a δW_{i-1} to w and w'. These insights imply that it is sufficient to consider only one representative of each class in $\overline{W}_{[i,j]}$ to determine the sets $\overline{W}_{[i-1,j]}$ and $\overline{W}_{[i,j+1]}$.

In conclusion, with our two key techniques of differential path reduction and message difference vector classes, we are able to efficiently compute $\mathrm{FDC}_{[t_b,t_e]}$.

4.7 Results

We have computed $\mathrm{FDC}_{[20,79]}$ for several interesting disturbance vectors. These results are shown in Sect. B and show the maximum success probability of these disturbance vectors over the last 60 steps. Although the total complexity of a collision attack also depends on the complexity over the non-linear part, these results provide important insights which of these disturbance vectors may possibly lead to the fastest collision attack.

4.8 Improvements for the last few steps of SHA-1

A common approach in constructing SHA-1 collision attacks is to remove the conditions for the last few steps as this will decrease the attack's overall complexity. The heuristic behind this effect is that for the last few steps some other differential paths that do not follow the disturbance vector actually have a higher success probability. Our approach can be adjusted by extending the sets $\mathcal{Q}_{76}, \ldots, \mathcal{Q}_{80}$ with differences ΔQ_i from these more likely alternative differential paths. We denote by $\mathrm{FDC}'_{[t_b,t_e]}$, $\mathcal{D}'_{[t_b,t_e]}$ and $\mathcal{R}'_{[t_b,t_e]}$ the respective function and sets wherein the extended sets $\mathcal{Q}'_{76}, \ldots, \mathcal{Q}'_{80}$ are used instead of $\mathcal{Q}_{76}, \ldots, \mathcal{Q}_{80}$. Algorithms that efficiently determine such extended sets $\mathcal{Q}'_{76}, \ldots, \mathcal{Q}'_{80}$ using ideas similar to the analysis in Sect. 4 are omitted here, but can be found in [Ste12a, Ch. 7.5].

5 New collision attacks on SHA-1

5.1 Open-source near-collision attack

In this section we present our near-collision attack on SHA-1 with an average complexity of $2^{57.5}$ compressions. Our near-collision attack is based on disturbance vector II(52,0). Below we describe how we used our new approach from

Table 5-1. SHA-1 near-collision differential path - round 1

t	Bitconditions: $\mathfrak{q}_t[31]\ldots\mathfrak{q}_t[0]$	ΔW_t
$-4,-3,-2$	`........ ........ ........ ........`	
-1	`...1.... ........ ........ ....0...`	
0	`.^.0.1.. .....0.1 ...00.10 .1..1..1`	$\{1, 26, 27\}$
1	`.0.+^-^^ ^^^^^1^0 ^^^11^10 .0..1.+0`	$\{\overline{4}, \overline{30}, 31\}$
2	`1-...+-- -------- -------- --.-1.+0`	$\{\overline{2}, 3, \overline{4}, \overline{26}, 28, 29, 31\}$
3	`.-.-.0.1 11111111 11110++1 +-1-00-0`	$\{\overline{2}, 26, 27, 28, \overline{29}\}$
4	`.-...1.0 11111111 1111-+++ ++0.1.+1`	$\{1, \overline{3}, 4, 26, \overline{27}, \overline{28}, \overline{29}, 31\}$
5	`.-...0.. ........ ......0. .+.+10+0`	$\{\overline{4}, \overline{29}\}$
6	`.-.+.... ........ ......01 100-.0+.`	$\{2, 3, 4, 26, \overline{29}\}$
7	`-1...1.. ........ ........ ...0.0..`	$\{2, 4, \overline{26}, \overline{27}, 29, 30, 31\}$
8	`1.1-.1.. ........ ........ .....1..`	$\{\overline{1}, 26, \overline{27}\}$
9	`..-..0.. ........ ........ ........`	$\{\overline{4}, 30, 31\}$
10	`^...00.. ........ ........ .......1`	$\{\overline{2}, \overline{3}, 4, \overline{26}, \overline{28}, 29, 31\}$
11	`..-.1... ........ ........ .......0`	$\{2, \overline{26}, 27, \overline{29}\}$
12	`0-..1... ........ ........ ......!.`	$\{\overline{3}, 4, \overline{26}, \overline{27}, \overline{28}, 29, 31\}$
13	`+..01... ........ ........ ........`	$\{\overline{4}, 28, 29, 31\}$
14	`..-1.... ........ ........ ......!.`	$\{\overline{2}, 3\}$
15	`+.0.1... ........ ........ ......!^`	$\{\overline{4}, \overline{27}, \overline{28}, \overline{29}, 31\}$
16	`+-0.0... ........ ........ ......!.`	$\{3, \overline{4}, \overline{27}\}$
17	`+..1.... ........ ........ ......^.`	$\{\overline{4}, \overline{27}, \overline{28}, \overline{29}, 30\}$
18	`-.+0.... ........ ........ ........`	$\{\overline{2}, 4, \overline{27}\}$
19	`-....... ........ ........ ........`	$\{4, 28, 29, \overline{30}\}$
20	`..+..... ........ ........ ........`	

Note: ΔW_t uses a compact notation, e.g., $\Delta W_t = +2^5 - 2^{10}$ is notated as $\{5, \overline{10}\}$.

Sect. 4 to determine which message bitrelations and δIHV_{out} to use and how we constructed the first round differential path. Collision search algorithms and various improvements using message modification techniques have already been covered extensively in the literature. We refer to our open-source implementation [Ste12b, Ste12a] for these details due to space considerations.

To apply our analysis of Sect. 4, we have chosen to use $t_b = 20$ (and $t_e = 79$). We use the improvements mentioned in Sect. 4.8 as this leads to higher success probabilities by a factor $2^{1.2}$. Let $\mathcal{D}' := \mathcal{D}'_{[20,79]}$, for $b \in \psi(\mathcal{D}')$, $e \in \phi(\mathcal{D}')$ and $w \in \omega(\mathcal{D}')$ we define $p_{b,e,w}$ and $p_{\max}$ as:

$$p_{b,e,w} = \sum_{\substack{\widehat{\mathcal{P}} \in \mathcal{D}' \\ \psi(\widehat{\mathcal{P}})=b,\ \phi(\widehat{\mathcal{P}})=e,\ \omega(\widehat{\mathcal{P}})=w}} \Pr[\widehat{\mathcal{P}}] \cdot c(b), \quad p_{\max} = \max_{b,e,w} p_{b,e,w}.$$

We algorithmically find a differential path over the first 20 steps that starts from $\delta IHV_{\text{in}} = 0$ and ends with working state differences $b \in \psi(\mathcal{D}')$ for which there are e and w such that $p_{b,e,w} = p_{\max}(=\text{FDC}'_{[20,79]}(\text{II}(52,0)))$. The differential path over the first round that we selected for our near-collision attack is shown in Tbl. 5-1 and fixes a specific value $\widehat{b}$ and specific message differences $\delta\widehat{W}_0, \ldots, \delta\widehat{W}_{19}$.

Table 5-2. SHA-1 near-collision attack target δIHV_{diff} values

$$\begin{aligned}
\mathcal{I}_1 = \big\{&(2^{11}+2^4-2^2, 2^6, 2^{31}, 2^1, 2^{31}), (2^{12}+2^{11}+2^9+2^4-2^2, 2^7+2^6+2^4, 2^{31}, 2^1, 2^{31}),\\
&(2^{12}+2^3+2^1, 2^7, 0, 2^1, 2^{31}), (2^{12}+2^{11}+2^4-2^2, 2^7+2^6, 2^{31}, 2^1, 2^{31}),\\
&(2^{12}+2^4-2^1, 2^7, 0, 2^1, 2^{31}), (2^{11}+2^9+2^4-2^2, 2^6+2^4, 2^{31}, 2^1, 2^{31}),\\
&(2^{12}+2^9+2^3+2^1, 2^7+2^4, 0, 2^1, 2^{31}), (2^{12}+2^9+2^4-2^1, 2^7+2^4, 0, 2^1, 2^{31})\big\};\\
\mathcal{I}_2 = \big\{&(v_1+c_1\cdot 2^3-c_2\cdot 2^5, v_2, v_3, v_4-c_3\cdot 2^2, v_5) \;\big|\; (v_i)_{i=1}^5 \in \mathcal{I}_1,\ c_1, c_2, c_3 \in \{0,1\}\big\};\\
\mathcal{I}_3 = \big\{&(v_1-c\cdot 2^{13}, v_2-c\cdot 2^8, v_3, v_4, v_5) \;\big|\; (v_i)_{i=1}^5 \in \mathcal{I}_2,\ c \in \{0,1\}\big\};\\
\widetilde{\mathcal{I}} = \big\{&(v_1-c\cdot 2^9, v_2-c\cdot 2^4, v_3, v_4, v_5) \;\big|\; (v_i)_{i=1}^5 \in \mathcal{I}_3,\ c \in \{0,1\}\big\};
\end{aligned}$$

Note: the resulting set $\widetilde{\mathcal{I}}$ has 192 unique target δIHV_{diff} values. Furthermore, for any $\delta IHV_{\text{diff}} \in \widetilde{\mathcal{I}}$ also $-\delta IHV_{\text{diff}} \in \widetilde{\mathcal{I}}$.

To maximize the success probability, we only accept δIHV_{out} in the set $\{e \in \phi(\mathcal{D}') \mid \exists w : p_{\widehat{b},e,w} > 0.9 p_{\max}\}$. We can further decrease overall complexity by only allowing w that maximize the number of $e = \delta IHV_{\text{out}}$ with $p_{\widehat{b},e,w} > 0.9 p_{\max}$. The near-collision attack gains a speed up due to the fact that it always has several chances of finding a target δIHV_{out}. Note that a possible second near-collision attack (for an identical-prefix collision attack) does not have the benefit of the speedup as it targets one specific $\delta IHV_{\text{out}} = 0$. More formally, for each $w \in \omega(\mathcal{D}')$, we count the number N_w of values e for which $p_{\widehat{b},e,w} > 0.9 p_{\max}$. Let $N_{\max} := \max_w N_w$ (which is 6 in our case) then we limit the allowed message difference vectors to the set $\mathfrak{W}_{[20,79]} = \{w \mid N_w = N_{\max}\}$. Hence, we only accept values for δIHV_{out} in the set $\{e \in \phi(\mathcal{D}') \mid \exists w \in \mathfrak{W}_{[20,79]} : p_{\widehat{b},e,w} > 0.9 p_{\max}\}$. In this manner we have found 192 target δIHV_{out}-values (see Tbl. 5-2).

With the differential path and the set of allowed δIHV_{out} known, we only need the message bit relations to construct a collision attack. We translate the set $\mathfrak{W}_{[20,79]}$ and the vector $(\delta\widehat{W}_i)_{i=0}^{19}$ into a smallest sufficient set of linear bit relations on the message words bits using linear algebra (see Sect. A).

Using the differential path, the message bitrelations and the set of allowed δIHV_{out}, we have implemented a near-collision attack. For more details, we refer to the source code which is available online at [Ste12b]. For more convenient analysis, the attack is split in four subsequent stages:

1. The first stage finds a message block pair satisfying the message bitrelations and which results in $\delta Q_i = 0$ for $i = 29, 30, 31, 32, 33$. This stage is the most complex and contains all speed ups using message modification techniques.
2. The second stage is to find a message block pair that satisfies the message bitrelations and results in $\delta Q_i = 0$ for $i = 49, 50, 51, 52, 53$.
3. The third stage is to find a message block pair that satisfies the message bitrelations and results in $\delta Q_i = 0$ for $i = 57, 58, 59, 60, 61$.
4. The fourth and final stage is to find a message block pair that results in one of the 192 target δIHV_{out} in Tbl. 5-2.

The last three stages cannot use any freedoms anymore and thereby either are or are not successful with some probability. The average total complexity

Table 5-3. Example message pair each consisting of an identical-prefix block and a near-collision block satisfying our differential path up to step 66.

First message	Second message
bc7e393a0470f684 e0a484dea556875a	bc7e393a0470f684 e0a484dea556875a
cddff9c82d02016b 860ee7f911e18418	cddff9c82d02016b 860ee7f911e18418
71bfbff1067095c9 ed44afee78122409	71bfbff1067095c9 ed44afee78122409
a3b2eb2e16c0cfc2 06c5202810383c2b	a3b2eb2e16c0cfc2 06c5202810383c2b
73e6e2c8437fb13e 4e4d5db6e383e01d	7fe6e2ca837fb12e fa4d5daadf83e019
7bea242c2bb63054 6845b1430c2194ab	c7ea24360bb63044 4c45b15fe02194bf
fb5236be2bc91e19 1d11bf8f665ef9ab	f75236bcebc91e09 a911bf934a5ef9af
9f8fe36a402cbf39 d77c1fb43cb00872	238fe372f02cbf29 d77c1fb884b00862

of our near-collision attack is thus the average complexity of the first stage divided by the product of the success probabilities for the last three stages. Our implementation outputs the throughput of the first stage in #/s as 'timeavg 40', and the success probabilities of the last three stages as 'avg 53 stats', 'avg 61 stats' and 'avg 80 stats', respectively. Using these numbers one can easily determine the average complexity in SHA-1 compressions to find a near-collision. With profiling and tuned optimization flags for the compiler and many hours-long runs, we determined an average complexity of the first stage to be $2^{20.91}$ SHA-1 compressions per message block pair. Using our novel analysis for step ranges [33,52], [53,60] and [61,79] and $N_{\max} = 6$, we determined the exact success probabilities for the last three stages, namely, $2^{-20.91}$, 2^{8} and $2^{16.65}$, respectively. These probabilities were verified by our implemented attack. Hence, the total complexity of our near-collision is $2^{11.97} \cdot 2^{20.91} \cdot 2^{8.00} \cdot 2^{16.65} = 2^{57.53}$ SHA-1 compressions. Finally, we like to note that with more than 50 bits of the 512 message bits left as degrees of freedom, there is ample room to further optimize the first stage with message modification techniques.

We provide an example message pair in Tbl. 5-3 that successfully passed the first three stages of our near-collision attack (at a cost of about $2^{40.9}$ compressions).

5.2 Identical-prefix collision attack on SHA-1

The near-collision attack of Sect. 5.1 can directly be used in a two-block identical-prefix collision attack on SHA-1.[6] The second near-collision block of the two blocks cancels the δIHV_{out} resulting from the first near-collision block.

For the second near-collision block, we follow the steps as described in Sect. 5.1 with two modifications. Firstly, in Sect. 5.1 we allow only $\delta IHV_{\text{out}} = 0$ (thus δIHV_{in} is canceled). This leads to $N_{\max} = 1$ and a different set of optimal message

[6] An additional identical-prefix block is used to satisfy a few bitconditions on the IHV (see Tbl. 5-1) and furthermore to simplify implementation and to allow very easy parallelization. It should be possible to remove this prefix block with only a negligible impact on the attack complexity.

difference vectors $\mathfrak{W}_{[20,79]}$. Hence, the total complexity over the last three stages increases by a factor 6. Secondly, instead of using a differential path starting with $\delta IHV_{\text{in}} = 0$ in Sect. 5.1, we use a differential path that starts with the (IHV, IHV') resulting from the first near-collision block.

A lower-bound for the complexity of a complete two-block identical-prefix collision attack based on our current near-collision implementation is about $(1 + 6) \cdot 2^{57.5} \approx 2^{60.3}$ compressions, as the first near-collision attack has the luxury of six allowed values for δIHV_{out} for each possible $(\delta W_t)_{t=0}^{79}$, whereas the second near-collision attack must target one specific δIHV_{out}. As the second-block differential path will differ roughly only up to $\mathfrak{q}_4$, almost all used message modification techniques will be unaffected. Also, there will still be a relative large amount of freedoms left to further apply message modification techniques. Hence, it is reasonable to expect a similar complexity in the first stage (first 32 steps). Nevertheless, leaving room for a small set back, we estimate the average complexity of our identical-prefix collision attack for SHA-1 to be equivalent to 2^{61} SHA-1 compressions.

5.3 Chosen-prefix collision attack

We present a chosen-prefix collision attack on SHA-1 using the second near-collision attack of Sect. 5.2 that does the following. Given chosen prefixes P and P', we first append bit strings S_{r} and S'_{r} such that the bit lengths of $P||S_{\text{r}}$ and $P'||S'_{\text{r}}$ are both equal to $N \cdot 512 - 119$. By processing the first $N - 1$ blocks of $P||S_{\text{r}}$ and $P'||S'_{\text{r}}$, we obtain IHV_{N-1} and IHV'_{N-1}, resp. Furthermore, let B and B' be the last $512 - 119$ bits of $P||S_{\text{r}}$ and $P'||S'_{\text{r}}$, resp. The next step is to perform a birthday search as explained in [vOW99] using a search space V and a step function $f : V \to V$. We define $V = \{0, 1\}^{119}$ and f (based on Tbl. 5-2) as:

$$f(v) = \begin{cases} \phi\big(\text{Compress}(IHV_{N-1}, B||v)\big) & \text{if } w(v) = 0 \bmod 2; \\ \phi\big(\text{Compress}(IHV'_{N-1}, B'||v) - (0, 0, 0, 0, 2^{31})\big) & \text{if } w(v) = 1 \bmod 2, \end{cases}$$

$$\phi(a, b, c, d, e) = (a[i])_{i=19}^{31}||(b[i])_{i=14}^{31}||(c[i])_{i=0}^{30}||(d[i])_{i=7}^{31}||e$$

The probability that a birthday collision results in one of the 192 target δIHV_{out} is found to be approximately $2^{-33.46}$ using Monte Carlo simulations. Therefore, a birthday search collision pair v, w with $f(v) = f(w)$ has a probability of $q = 2^{-33.46-1}$ that $\tau(v) \neq \tau(w)$ and δIHV_N is one of the 192 target δIHV_{out}-values. Using the analysis from [vOW99], this implies that the expected birthday search complexity in SHA-1 compressions is $\sqrt{\pi \cdot |V|/(2 \cdot q)} \approx 2^{77.06}$.

To complete the chosen-prefix collision attack it remains to find a near-collision block that cancels δIHV_N. But as δIHV_N is one of the 192 target δIHV_{out}, we can directly use the construction of the second near-collision block of Sect. 5.2, whose complexity is significantly lower than $2^{77.06}$. Hence, the overall cost of a chosen-prefix collision attack on SHA-1 is dominated by the expected $2^{77.1}$ SHA-1 compressions required for the birthday search.

6 Concluding remarks

We have presented new collision attacks on SHA-1, most importantly an identical-prefix collision attack with an average complexity of 2^{61} compressions. With the construction of these attacks, we focused mostly on obtaining the highest success probability that is theoretically possible over the linear part. Our novel direction in the cryptanalysis of SHA-1 is essentially based on an exhaustive and exact analysis of all possible differential paths that follow the disturbance vector. This is in contrast to previous approaches that combine success probabilities and conditions of individual local collisions with heuristic corrections. In this paper we have introduced the foundations of our novel direction. For a complete and rigorous mathematical treatment we refer to the full version [Ste12a].

As our attacks have still over 50 out of the 512 message bits left as degrees of freedom for further improvements using message modification techniques, we hope that our novel methods provide the necessary advantage to construct attacks with complexity well below 2^{61} compressions and thereby contributes to the search for the long-anticipated first SHA-1 collision.

References

[BCJ+05] Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet, and William Jalby, *Collisions of SHA-0 and Reduced SHA-1*, EUROCRYPT (Ronald Cramer, ed.), Lecture Notes in Computer Science, vol. 3494, Springer, 2005, pp. 36–57.

[Che11] Rafael Chen, *New Techniques for Cryptanalysis of Cryptographic Hash Functions*, Ph.D. thesis, Technion, Aug 2011.

[CJ98] Florent Chabaud and Antoine Joux, *Differential Collisions in SHA-0*, CRYPTO (Hugo Krawczyk, ed.), Lecture Notes in Computer Science, vol. 1462, Springer, 1998, pp. 56–71.

[CMR07] Christophe De Cannière, Florian Mendel, and Christian Rechberger, *Collisions for 70-Step SHA-1: On the Full Cost of Collision Search*, Selected Areas in Cryptography (Carlisle M. Adams, Ali Miri, and Michael J. Wiener, eds.), Lecture Notes in Computer Science, vol. 4876, Springer, 2007, pp. 56–73.

[Coc07] Martin Cochran, *Notes on the Wang et al. 2^{63} SHA-1 Differential Path*, Cryptology ePrint Archive, Report 2007/474, 2007.

[CR06] Christophe De Cannière and Christian Rechberger, *Finding SHA-1 Characteristics: General Results and Applications*, ASIACRYPT (Xuejia Lai and Kefei Chen, eds.), Lecture Notes in Computer Science, vol. 4284, Springer, 2006, pp. 1–20.

[GA11] E.A. Grechnikov and A.V. Adinetz, *Collision for 75-step SHA-1: Intensive Parallelization with GPU*, Cryptology ePrint Archive, Report 2011/641, 2011.

[Gre10] E.A. Grechnikov, *Collisions for 72-step and 73-step SHA-1: Improvements in the Method of Characteristics*, Cryptology ePrint Archive, Report 2010/413, 2010.

[Man11] Stéphane Manuel, *Classification and generation of disturbance vectors for collision attacks against SHA-1*, Des. Codes Cryptography **59** (2011), no. 1-3, 247–263.

[MHP09] Cameron McDonald, Philip Hawkes, and Josef Pieprzyk, *Differential Path for SHA-1 with complexity $O(2^{52})$*, Cryptology ePrint Archive, Report 2009/259, 2009.

[MPRR06] Florian Mendel, Norbert Pramstaller, Christian Rechberger, and Vincent Rijmen, *The Impact of Carries on the Complexity of Collision Attacks on SHA-1*, FSE (Matthew J. B. Robshaw, ed.), Lecture Notes in Computer Science, vol. 4047, Springer, 2006, pp. 278–292.

[MRR07] Florian Mendel, Christian Rechberger, and Vincent Rijmen, *Update on SHA-1*, Rump session of CRYPTO 2007, 2007.

[PCTH11] T. Polk, L. Chen, S. Turner, and P. Hoffman, *Security Considerations for the SHA-0 and SHA-1 Message-Digest Algorithms*, Internet Request for Comments, March 2011, RFC 6194.

[Ste12a] Marc Stevens, *Attacks on Hash Functions and Applications*, Ph.D. thesis, Leiden University, June 2012.

[Ste12b] Marc Stevens, *SHA-1 near collision attack source code*, 2012, `https://hashclash.googlecode.com/files/sha1_nearcoll_attack.zip`.

[vOW99] Paul C. van Oorschot and Michael J. Wiener, *Parallel Collision Search with Cryptanalytic Applications*, J. Cryptology **12** (1999), no. 1, 1–28.

[WFLY04] Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu, *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*, Cryptology ePrint Archive, Report 2004/199, 2004.

[WY05] Xiaoyun Wang and Hongbo Yu, *How to Break MD5 and Other Hash Functions*, EUROCRYPT (Ronald Cramer, ed.), Lecture Notes in Computer Science, vol. 3494, Springer, 2005, pp. 19–35.

[WYY05a] Xiaoyun Wang, Andrew C. Yao, and Frances Yao, *Cryptanalysis on SHA-1*, NIST Cryptographic Hash Workshop Presentation, 2005.

[WYY05b] Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu, *Finding Collisions in the Full SHA-1*, CRYPTO (Victor Shoup, ed.), Lecture Notes in Computer Science, vol. 3621, Springer, 2005, pp. 17–36.

[WYY05c] Xiaoyun Wang, Hongbo Yu, and Yiqun Lisa Yin, *Efficient Collision Search Attacks on SHA-0*, CRYPTO (Victor Shoup, ed.), Lecture Notes in Computer Science, vol. 3621, Springer, 2005, pp. 1–16.

A Deriving message bitrelations

For each $\widehat{w} = (\delta\widehat{W}_i)_{i=20}^{79} \in \mathfrak{W}_{[20,79]}$ we define the set $\mathcal{V}_{\widehat{w}}$ as the set of all $(W_i)_{i=0}^{79}$ that 'result' in $\widehat{w}$, i.e., $(W_i \oplus DW_i) - W_i = \delta\widehat{W}_i$ for all $i \in \{20, \ldots, 79\}$. Let the set $\mathcal{V} = \bigcup_{w \in \mathfrak{W}_{[20,79]}} \mathcal{V}_w$ consist of all $(W_t)_{t=0}^{79}$ that are compatible with some $w \in \mathfrak{W}_{[20,79]}$. Furthermore, let $\mathcal{V}'$ be the set consisting of all elements of $\mathcal{V}$ mapped to $\mathbb{F}_2^{32\cdot 80}$. We search for an affine subspace $y + \mathcal{U} \subseteq \mathcal{V}'$ which is as large as possible. Choose any basis of $\mathcal{U}^\perp$ of size k and let the k rows of the matrix $A_{[20,79]} \in \mathbb{F}_2^{k \times (32\cdot 80)}$ consist of the k basis vectors of $\mathcal{U}^\perp$. It follows that $x \in \mathcal{U} \Leftrightarrow A_{[20,79]} \cdot x = 0$ and thus $x \in y + \mathcal{U} \Leftrightarrow A_{[20,79]} \cdot x = A_{[20,79]} \cdot y$. The

matrix equation $A_{[20,79]} \cdot x = c_{[20,79]}$ with $c_{[20,79]} = A_{[20,79]} \cdot y$ describes sufficient linear bit relations for steps 20 up to 79.[7]

The set $\mathfrak{W}_{[0,19]} = \{(\delta\widehat{W}_i)_{i=0}^{19}\}$ similarly leads to a matrix equation $A_{[0,19]} \cdot x = c_{[0,19]}$. The two matrix equations can be combined into a single matrix equation $A_{[0,79]} \cdot x = c_{[0,79]}$ that defines our message search space. Finally, this matrix equation over the $32 \cdot 80$ message words bits is reduced using the message expansion relation to a matrix equation over the 512 message block bits.

B SHA-1 disturbance vector analysis

Tbl. B-1 is based on the disturbance vector cost function $\mathrm{FDC}_{[t_b,t_e],u}$ that is defined as similar to $\mathrm{FDC}_{[t_b,t_e]}$, but under the additional constraint that only up to u carries are allowed in the working state differences ΔQ_i. More formally, we define:

$$\mathcal{Q}_{t,u} := \left\{ \text{BSDR } Y \;\middle|\; \begin{array}{c} \sigma(Y)=\sigma(Z), \\ Z[i]\in\{-DV_{t-1}[i], DV_{t-1}[i]\},\ i=0,\ldots,31, \\ w(Y)\leq u+\min_{X\in\mathcal{Q}_t} w(X). \end{array} \right\};$$

$$\mathcal{D}_{[t_b,t_e],u} := \left\{ \widehat{\mathcal{P}} \;\middle|\; \Delta\widehat{Q_i} \in \mathcal{Q}_{i,u},\ \delta\widehat{W_j} \in \mathcal{W}_j,\ \Pr[\widehat{\mathcal{P}}] > 0 \right\};$$

$$\mathrm{FDC}_{[t_b,t_e],u}\left((DV_t)_{t=0}^{79}\right) = \max_{b,e,w} \sum_{\substack{\widehat{\mathcal{P}}\in\mathcal{D}_{[t_b,t_e],u} \\ \psi(\widehat{\mathcal{P}})=b,\ \phi(\widehat{\mathcal{P}})=e,\ \omega(\widehat{\mathcal{P}})=w}} \Pr[\widehat{\mathcal{P}}] \cdot c(b),$$

where $c(b) = c((\Delta Q_i)_{i=t_b-4}^{t_e})$ is the correction factor $c(b) = \prod_{i=t_b-4}^{t_b-2} 2^{w(\Delta\widehat{Q}_i)}$.

The tables below contain notes $\epsilon = 0, 1/8, 1/4, 1/2$ for each entry. This note indicates whether in our algorithms to compute $\mathrm{FDC}_{[t_b,t_e],u}$ we removed certain message difference vectors w that had a 'total success probability of w' less than ϵ times the highest 'total success probability over all w''. Although, we won't go into the details of the notationally heavy definition of this 'total success probability', it is clear that choosing $\epsilon = 0$ will cause no message difference vector to be removed. Choosing $\epsilon > 0$ will result in that the maximum taken in $\mathrm{FDC}_{[t_b,t_e],u}$ will actually be taken over a subset of all values w. Hence, choosing $\epsilon > 0$ can only affect the outcome in a negative way, i.e., a smaller maximum success probability. Although for ϵ close to 1, this removal of message difference vectors does affect the outcome (in a negative way), we have not seen this happen for $\epsilon \leq 0.5$ for all selected studied cases. Choosing $\epsilon > 0$ allows us to compute lower-bounds for $\mathrm{FDC}_{[t_b,t_e],u}$ for disturbance vectors and values for u that were otherwise prohibitive for our particular machine due to memory requirements. We argue that for up to $\epsilon \leq 0.5$ these values are not just lower-bounds, but in fact the correct outcome for $\mathrm{FDC}_{[t_b,t_e],u}$, which is backed-up by the fact that for increasing u these outcomes increase as expected and no sudden decrease is seen (or, when taking the $-\log_2$, decrease as expected and no sudden increase is seen).

[7] Although this seems to be impractical, we can compute this efficiently by splitting it into independent parts and using well chosen representations.

Table B-1. Disturbance vector results

	u				
DV	0	1	2	3	7
I(42, 0)	82.68 ϵ=0	78.67 ϵ=0	78.36 ϵ=1/4		
I(43, 0)	82.00 ϵ=0	77.65 ϵ=0	77.31 ϵ=1/8		
I(44, 0)	81.00 ϵ=0	77.41 ϵ=0	77.1 ϵ=0	76.98 ϵ=0	76.89 ϵ=1/8
I(45, 0)	81.00 ϵ=0	76.91 ϵ=0	76.66 ϵ=0	76.54 ϵ=0	76.45 ϵ=1/8
I(46, 0)	79.00 ϵ=0	75.02 ϵ=0	74.92 ϵ=0	74.84 ϵ=0	74.83 ϵ=1/8
I(47, 0)	79.00 ϵ=0	75.15 ϵ=0	74.83 ϵ=0	74.71 ϵ=0	74.61 ϵ=0
I(**48**, **0**)	75.00 ϵ=0	71.84 ϵ=0	71.61 ϵ=0	71.51 ϵ=0	71.42 ϵ=0
I(**49**, **0**)	76.00 ϵ=0	72.59 ϵ=0	72.34 ϵ=0	72.24 ϵ=0	72.15 ϵ=0
I(**50**, **0**)	75.00 ϵ=0	72.02 ϵ=0	71.95 ϵ=0	71.93 ϵ=0	71.92 ϵ=0
I(51, 0)	77.00 ϵ=0	73.76 ϵ=0	73.53 ϵ=0	73.43 ϵ=0	73.34 ϵ=0
I(52, 0)	79.00 ϵ=0	76.26 ϵ=0	76.24 ϵ=0	76.24 ϵ=0	76.24 ϵ=0
I(53, 0)	82.83 ϵ=0	78.86 ϵ=0	78.79 ϵ=0	78.77 ϵ=0	78.77 ϵ=0
I(54, 0)	82.83 ϵ=0	79.60 ϵ=0	79.38 ϵ=0	79.28 ϵ=0	79.19 ϵ=0
I(55, 0)	81.54 ϵ=0	78.67 ϵ=0	78.42 ϵ=0	78.32 ϵ=0	78.23 ϵ=0
I(56, 0)	81.54 ϵ=0	79.10 ϵ=0	79.03 ϵ=0	79.01 ϵ=0	79.01 ϵ=0
I(42, 2)	85.09 ϵ=0	82.17 ϵ=1/4	81.84 ϵ=1/2	81.72 ϵ=1/2	
I(43, 2)	84.42 ϵ=0	81.15 ϵ=1/4	80.78 ϵ=1/2		
I(44, 2)	84.42 ϵ=0	81.92 ϵ=0	81.57 ϵ=1/4	81.45 ϵ=1/2	81.36 ϵ=1/2
I(45, 2)	83.42 ϵ=0	80.80 ϵ=0	80.52 ϵ=0	80.41 ϵ=1/4	80.32 ϵ=1/2
I(46, 2)	80.42 ϵ=0	78.10 ϵ=0	78.00 ϵ=0	77.99 ϵ=1/8	77.99 ϵ=1/4
I(47, 2)	79.68 ϵ=0	77.01 ϵ=0	76.68 ϵ=0	76.56 ϵ=0	76.47 ϵ=1/8
I(48, 2)	76.68 ϵ=0	74.27 ϵ=0	73.99 ϵ=0	73.88 ϵ=0	73.79 ϵ=0
I(49, 2)	77.00 ϵ=0	74.30 ϵ=0	74.02 ϵ=0	73.92 ϵ=0	73.83 ϵ=0
I(50, 2)	77.00 ϵ=0	74.74 ϵ=0	74.63 ϵ=0	74.61 ϵ=0	74.60 ϵ=0
I(51, 2)	80.00 ϵ=0	77.47 ϵ=0	77.21 ϵ=0	77.11 ϵ=0	77.03 ϵ=0
I(52, 2)	82.00 ϵ=0	79.98 ϵ=0	79.93 ϵ=0	79.92 ϵ=0	79.92 ϵ=0
I(53, 2)	84.00 ϵ=0	81.91 ϵ=0	81.80 ϵ=0	81.78 ϵ=0	81.78 ϵ=0
I(54, 2)	84.00 ϵ=0	81.37 ϵ=0	81.06 ϵ=0	80.95 ϵ=0	80.85 ϵ=0
I(55, 2)	84.00 ϵ=0	81.78 ϵ=0	81.53 ϵ=0	81.43 ϵ=0	81.34 ϵ=0
I(56, 2)	82.00 ϵ=0	80.22 ϵ=0	80.13 ϵ=0	80.12 ϵ=0	80.11 ϵ=0
II(44, 0)	87.00 ϵ=0	79.51 ϵ=1/2			
II(45, 0)	83.00 ϵ=0	75.45 ϵ=1/8	74.82 ϵ=1/2		
II(**46**, **0**)	76.00 ϵ=0	71.85 ϵ=0	71.83 ϵ=1/2		
II(47, 0)	81.42 ϵ=0	76.23 ϵ=0	75.87 ϵ=1/2		
II(48, 0)	80.00 ϵ=0	76.11 ϵ=0	75.89 ϵ=0	75.79 ϵ=1/8	
II(49, 0)	80.00 ϵ=0	75.04 ϵ=0	74.72 ϵ=0	74.60 ϵ=0	74.51 ϵ=1/2
II(**50**, **0**)	78.00 ϵ=0	73.52 ϵ=0	73.23 ϵ=0	73.12 ϵ=0	73.02 ϵ=0
II(**51**, **0**)	77.00 ϵ=0	72.55 ϵ=0	72.18 ϵ=0	72.02 ϵ=0	71.88 ϵ=0
II(**52**, **0**)	75.00 ϵ=0	71.88 ϵ=0	71.87 ϵ=0	71.76 ϵ=0	71.75 ϵ=0
II(53, 0)	76.96 ϵ=0	73.65 ϵ=0	73.34 ϵ=1/8	73.23 ϵ=1/8	73.14 ϵ=1/8
II(54, 0)	77.96 ϵ=0	73.97 ϵ=0	73.74 ϵ=1/8	73.64 ϵ=1/8	73.55 ϵ=1/8
II(55, 0)	77.96 ϵ=0	75.22 ϵ=1/8	74.99 ϵ=1/2	74.89 ϵ=1/2	74.80 ϵ=1/2
II(56, 0)	76.96 ϵ=0	74.48 ϵ=1/2	74.18 ϵ=1/2	74.07 ϵ=1/2	73.97 ϵ=1/2
II(45, 2)	85.00 ϵ=0	78.64 ϵ=1/2			
II(46, 2)	82.00 ϵ=0	77.51 ϵ=1/2			
II(47, 2)	85.42 ϵ=0	79.83 ϵ=1/2			
II(48, 2)	83.00 ϵ=0	78.81 ϵ=1/2	78.46 ϵ=1/2		
II(49, 2)	83.00 ϵ=0	78.09 ϵ=0	77.74 ϵ=1/2		
II(50, 2)	81.00 ϵ=0	76.51 ϵ=0	76.16 ϵ=1/8	76.03 ϵ=1/8	
II(51, 2)	82.00 ϵ=0	77.74 ϵ=0	77.36 ϵ=1/8	77.20 ϵ=1/8	
II(52, 2)	82.00 ϵ=0	79.07 ϵ=0	78.96 ϵ=0	78.94 ϵ=0	78.93 ϵ=1/2
II(53, 2)	83.00 ϵ=0	79.60 ϵ=0	79.30 ϵ=0	79.18 ϵ=0	79.09 ϵ=1/8
II(54, 2)	84.00 ϵ=0	80.49 ϵ=0	80.21 ϵ=0	80.10 ϵ=0	80.00 ϵ=1/8
II(55, 2)	84.00 ϵ=0	81.20 ϵ=0	80.88 ϵ=0	80.76 ϵ=0	80.67 ϵ=1/8
II(56, 2)	85.00 ϵ=0	82.69 ϵ=1/4	82.39 ϵ=1/4	82.27 ϵ=1/4	82.18 ϵ=1/4

Note: the columns are the negative $\log_2$ results of the cost function $\mathrm{FDC}_{[20,79],u}$.