Second-Preimage Analysis of Reduced SHA-1

Christian Rechberger

[1] Department of Electrical Engineering ESAT/COSIC, Katholieke Universiteit Leuven. Kasteelpark Arenberg 10, B–3001 Heverlee, Belgium.
[2] Interdisciplinary Institute for BroadBand Technology (IBBT), Belgium.

christian.rechberger@esat.kuleuven.be

Abstract. Many applications using cryptographic hash functions do not require collision resistance, but some kind of preimage resistance. That's also the reason why the widely used SHA-1 continues to be recommended in all applications except digital signatures after 2010. Recent work on preimage and second preimage attacks on reduced SHA-1 succeeding up to 48 out of 80 steps (with results barely below the 2^n time complexity of brute-force search) suggest that there is plenty of security margin left. In this paper we show that the security margin is actually somewhat lower, when only second preimages are the goal. We do this by giving two examples, using known differential properties of SHA-1. First, we reduce the complexity of a 2nd-preimage shortcut attack on 34-step SHA-1 from an impractically high complexity to practical complexity. Next, we show a property for up to 61 steps of the SHA-1 compression function that violates some variant of a natural second preimage resistance assumption, adding 13 steps to previously best known results.

Keywords: hash function, cryptanalysis, SHA-1, preimage, second preimage, differential

1 Introduction and overview

After the spectacular collision attacks on MD5 and SHA-1 by Wang *et al.* and follow-up work [7,13,37,40,41,42], implementors reconsider their choices. While starting a very productive phase of research on design and analysis of cryptographic hash functions, the impact of these results in terms of practical and worrying attacks turned out to be less than anticipated (exceptions are e.g. [18,36,38]). In addition to collision resistance, another property of hash functions is crucial for practical security: preimage resistance. Hence, research on preimage attacks and the security margin of hash functions against those attacks seems well motivated, especially if those hash functions are in practical use.

1.1 Motivation: security margin of SHA-1 against preimage style attacks

SHA-1 continues to get recommended by NIST even after 2010 for applications that do not require collision resistance [23]. Hence, SHA-1 will globally remain in practical use for a long time. Even though close to practical collision attacks for SHA-1 are described in [6,40], it's resistance against preimage attacks seems very solid.

1.2 The contribution

Progress in the cryptanalysis of a round-based primitive is often monitored via considering the highest number of rounds for which an attack method violates some assumption about the primitive. For preimage attack, the meet-in-the-middle approach [3,10,15,17,34,35] proved to be successful in doing so. To this end, we devise methods that exhibit non-ideal behavior regarding variants of second preimage resistance for significantly more steps of the SHA-1 compression function (see Sect. 5.2). Another concern is the efficiency of attacks. Also here, we can demonstrate significant efficiency improvements for a step-reduced SHA-1 hash function. Details for this can be found in Section 5.1. As a summary, see Section 1.3 for an overview and a comparison. What is the reason for these improvements? We exclude preimage attacks and specifically use the knowledge of a first preimage to get an advantage as an attacker. The approach we use takes advantage of the existence of differentials with relatively high probability, i.e. it exploits the similar weaknesses that also led to efficient collision search attacks.

1.3 Preview of our results on SHA-1

We summarize our results on the second-preimage resistance of SHA-1 hash function and compression function in Table 1 and 2, respectively. There, they are compared with preimage attacks of De Cannière and Rechberger from Crypto 2008 [8], and to preimage attacks from Aoki and Sasaki, from Crypto 2009 [3]. The method in this paper is sensitive to changes of the Boolean function used in the round transformation, hence we distinguish between round-reduced variants that start from step 0, and those that can start anywhere. Note that [8] is not sensitive to the Boolean function used, and hence the number of rounds can not be reduced or extended with a different choice, In case of [3], the impact of the choice of different starting rounds for the reduced variant is more difficult to assess, but likely to be limited. Interestingly, whereas we can cover many steps of the SHA-1 compression function and still show less than ideal properties of it, we fail to do so for the SHA-1 hash function. The efficiency improvement for 34-step SHA-1 however works for both the compression function and the hash function.

1.4 Related work

This approach was already proposed for MD4 in its basic form by Yu *et al.* [44]. There, a characteristic through all 48 steps of MD4 with probability 2^{-56} was used to state that one in 2^{56} messages is a weak message with respect to a 2nd-preimage attack. Leurent noted [19] that for long messages, this can be turned

Table 1. Comparison of various variants of preimage attacks on the SHA-1 hash function with reduced number of rounds.

rounds	complexity time/memory/prob.	type	technique	source
34 (00-33)	$2^{77}/2^{15}/ > 0.5$	2nd-preimage	imp. msg. + P^{3}graph	[8]
34 (00-33)	$2^{42.42}$/negl./ > 0.5	2nd-preimage	differential	Sect. 5
44 (00-43)	$2^{157}/2^{21}/ > 0.5$	preimage	imp. msg. + P^{3}graph	[8]
45 (00-44)	$2^{159}/2^{21}/ > 0.5$	2nd-preimage	imp. msg. + P^{3}graph	[8]
48 (00-47)	$2^{159.3}/2^{40}/ > 0.5$	preimage	MITM	[3]
48 (00-47)	$2^{159.8}$/negl./ > 0.5	preimage	MITM	[3]
48 (00-47)	$2^{159.27}$/negl./ > 0.5	preimage	optimized brute force	[27]

Table 2. Comparison of various variants of preimage attacks on the SHA-1 compression function with reduced number of rounds.

rounds	complexity time/memory/prob.	type	technique	source
34 (00-33)	$2^{69}/ - / > 0.5$	preimage	imp. msg.	[8]
34 (00-33)	1 / negl. /$2^{-42.25}$	2nd-preimage	differential	Sect. 5
45 (00-44)	$2^{157}/ - / > 0.5$	preimage	imp. msg.	[8]
48 (00-47)	$2^{156.7}/2^{40}/ > 0.5$	preimage	MITM	[3]
48 (00-47)	$2^{157.7}$/negl. / > 0.5	preimage	MITM	[3]
61 (18-79)	1 / negl. /$2^{-159.42}$	2nd-preimage	differential	Sect. 5

into an attack actually finding a 2nd-preimage with complexity 2^{56}. Considering second preimage attacks on HMAC when instantiated with concrete hash functions, Kim *et al.* [16] give e.g. results for MD5 up to 33 out of the 64 steps, and for SHA-1 for up to 42 steps.

Relations among various notions of preimage-style resistance requirements are studied in numerous work, e.g. [30,33,39]. Using the notation of [33], we study the aSec property of SHA-1, and show that the SHA-1 compression function is not ideally aSec-secure for up to 61 steps. An example of a construction that explicitly uses the second preimage resistance of a compression function appears in [2].

1.5 Outline of the paper

We start with a simple definition of second preimage resistance for iterated hash functions in Section 2, followed by a description of SHA-1 in Section 3. The idea of the attack is presented in Section 4. We apply the ideas to step-reduced SHA-1 and show an attack on the compression function and the hash function SHA-1 in Section 5. Finally, we discuss our findings and open problems in Section 6.

2 Definitions

Let an iterated hash function F be built by iterating a compression function $f : \{0,1\}^l \times \{0,1\}^n \rightarrow \{0,1\}^n$ as follows:

- Split the message m of arbitrary length into k blocks x_i of size l.
- Set h_0 to a pre-specified IV
- Compute $\forall x_i : h_i = f(h_{i-1}, x_i)$
- Output $F(m) = h_k$

A basic informal definition of second preimage resistance of a hash function is as follows:

Definition 1. *Given $F(\cdot)$, m, it should be hard to find an $m^* \neq m$ such that $F(m^*) = F(m)$. For a hash function with n-bit output size, every guess for an m^* should have success probability of 2^{-n}, and the work to find an m^* should be no less than 2^n.*

Def. 1 applies analogously to a compression function, i.e. with a fixed length instead of arbitrary length input. For a more formal treatment, we refer to [30,33,39].

3 Description of SHA-1

SHA-1 is an iterative hash function that processes up to 2^{55} 512-bit input message blocks and produces a 160-bit hash value. Like many hash functions used today, it is based on the design principle of MD4, pioneered by Rivest [32]. In the following we briefly describe the SHA-1 hash function. It basically consists of two parts: the message expansion and the state update transformation. A detailed description of the hash function is given in [24].

Table 3. Notation

notation	description
$X \oplus Y$	bit-wise XOR of X and Y
$X + Y$	addition of X and Y modulo 2^{32}
X	arbitrary 32-bit word
X^2	pair of words, shortcut for (X, X^*)
M_i	input message word i (32 bits)
W_i	expanded input message word t (32 bits)
$X \lll n$	bit-rotation of X by n positions to the left, $0 \leq n \leq 31$
$X \ggg n$	bit-rotation of X by n positions to the right, $0 \leq n \leq 31$
N	number of steps of the compression function

3.1 Message expansion

The message expansion of SHA-1 is a linear expansion of the 16 message words (denoted by M_i) to 80 expanded message words W_i.

$$W_i = \begin{cases} M_i, & \text{for } \ 0 \leq i \leq 15, \\ (W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16}) \lll 1 & \text{for } 16 \leq i \leq 79 \ . \end{cases} \tag{1}$$

3.2 State update transformation

The state update transformation of SHA-1 consists of 4 rounds of 20 steps each. In each step the expanded message word W_i is used to update the 5 chaining variables A_i, B_i, C_i, D_i, E_i as follows:

$$\begin{aligned} A_{i+1} &= E_i + A_i \lll 5 + f(B_i, C_i, D_i) + K_j + W_i \\ B_{i+1} &= A_i \\ C_{i+1} &= B_i \ggg 2 \\ D_{i+1} &= C_i \\ E_{i+1} &= D_i \end{aligned}$$

Note that the function f depends on the actual round: round 1 (steps 0 to 19) use f_{IF} and round 3 (steps 40 to 59) use f_{MAJ}. The function f_{XOR} is applied in round 2 (steps 20 to 39) and round 4 (steps 60 to 79). The functions are defined as follows:

$$f_{IF}(B, C, D) = B \wedge C \oplus \overline{B} \wedge D \tag{2}$$

$$f_{MAJ}(B, C, D) = B \wedge C \oplus B \wedge D \oplus C \wedge D \tag{3}$$

$$f_{XOR}(B, C, D) = B \oplus C \oplus D \ . \tag{4}$$

After the last step of the state update transformation, the chaining variables A_0, B_0, C_0, D_0, E_0 and the output values of the last step $A_{80}, B_{80}, C_{80}, D_{80}, E_{80}$ are combined using word-wise modular addition, resulting in the final value of one iteration (feed forward). The result is the final hash value or the initial value for the next message block.

Note that $B_i = A_{i-1}$, $C_i = A_{i-2} \ggg 2$, $D_i = A_{i-3} \ggg 2$, $E_i = A_{i-4} \ggg 2$. This also implies that the chaining inputs fill all A_j for $-4 \leq j \leq 0$. Thus it suffices to consider the state variable A, which we will for the remainder of this paper.

4 Violating second preimage resistance properties with differentials

Assuming the existence of a differential with a certain probability $p > 2^{-n}$, there are two ways to use such a differential in 2nd-preimage attacks. One is to simply use this differential for a single attempt to find a second preimage by being given the first preimage. With $p > 2^{-n}$, this shows less than ideal behavior of the function, even though on average it hardly speeds up the search for an actual second preimage. The second way is to apply this differential in an iterated hash function on individual message blocks, and thereby increasing this probability to actually find a second preimage. In this setting, if the number of message blocks that can be tried is larger than p^{-1} a second preimage can be expected with high probability.

For the description of our approach, we use the framework developed for SHA-1 characteristics by De Cannière and Rechberger [7], and adapt it to the second preimage setting at hand. In the following, we briefly recall those parts that are needed later on.

The expected difference between a particular pair of words X^2 will be denoted by ∇X. For every bit in this pair, we write 'x' if we expect a difference between the same bits of both words, and we write '-' if we do not expect a difference between those two bits.

Let us assume that we are given a complete characteristic for N-step SHA-1, specified by $\nabla A_{-4}, \ldots, \nabla A_N$ and $\nabla W_0, \ldots, \nabla W_{N-1}$, detailing for every bit and every word in the computation, whether or not we expect a difference at a particular bit position. Our goal is to estimate how much effort it would take to, given a message, find another message which follows this characteristic, assuming a simple depth-first search algorithm which tries to determine the pairs of message words L_i^2 one by one starting from L_0^2. In order to estimate the work factor of this algorithm, we will compute the expected number of visited nodes in the search tree. But first another definition, which is needed to estimate the work factor.

Definition 2 ([7]). *The* uncontrolled probability $P_u(i)$ *of a characteristic at step i is the probability that the output* A_{i+1}^2 *of step i follows the characteristic, given that all input pairs do as well, i.e.,*

$$P_u(i) = P\left(A_{i+1}^2 \in \nabla A_{i+1} \mid A_{i-j}^2 \in \nabla A_{i-j} \text{ for } 0 \le j < 5, \text{ and } W_i^2 \in \nabla W_i\right) .$$

With the definition above, we can now easily express the number of nodes $N_s(i)$ visited at each step of the compression function during the second preimage search.

Taking into account that the average number of children of a node at step i is $P_u(i)$, and that the search stops as soon as step N is reached, we can derive the following recursive relation:

$$N_s(i) = \begin{cases} 1 & \text{if } i = N\,, \\ N_s(i+1) \cdot P_u^{-1}(i) & \text{if } i < N\,. \end{cases}$$

The total work factor is then given by

$$N_w = \sum_{i=1}^{N} N_s(i)\,. \tag{5}$$

It is now easy to see that we have two different quantities that define the search for a second preimage. One is the number of step computations N_w, which should be noticeably below $2^n \cdot N$ to be considered an attack. The other one is the number of distinct message blocks N_m that need to be tried during the search:

$$N_m = \prod_{j=1}^{N} P_u(j)^{-1} = N_s(0)\,. \tag{6}$$

Note that N_m could theoretically be above 2^n, while the resulting work factor can still be below an equivalent of 2^n compression function computations. This is because the tree-based model of the search takes *early-stop strategies* into account. However, this only works if in addition to the first preimage, also all intermediate chaining values that lead to the target hash are already available to the attacker. This may be the case in certain settings, but is certainly not a standard assumption for second preimage attacks.

Without this additional assumption on data available to an attacker, the workfactor is in fact

$$N_w = N \cdot \prod_{j=1}^{N} P_u(j)^{-1}\,. \tag{7}$$

We will refer to this as setting 2, and will use setting 1 (and Eq. 5) when we assume the availability of internal chaining inputs.

5 Application to SHA-1

In order to find attacks on the SHA-1 compression function, or the SHA-1 hash function, characteristics need to be found that result in a workfactor N_w which should be noticeably below $2^n \cdot N$. The search algorithms we used are based on methods developed in the early cryptanalysis of SHA-1 regarding collision attacks [4,20,26,31] with the improvement that exact probabilities as described in [7] instead of Hamming weights are used to prune and rank them. More recent characteristic search algorithms (e.g. [7,12,21,43]) which exploit the fact that non-linear propagation of differences with low probability can be useful in collision attacks do not appear to be applicable to the setting considered in this paper. Depending on whether the hash- or the compression function is considered, the chaining input $\nabla A_{-4} \ldots \nabla A_0$ is allowed to have a difference or not.

In order to explain various aspects of the method, we consider two case studies. The first is the SHA-1 hash reduced to the first 34 steps and discussed in Section 5.1. There we show that better attack complexities can be obtained. The second is the SHA-1 compression function reduced to 61 steps and discussed in Section 5.2. There we aim for having results on a higher number of steps.

5.1 Hash function attacks: 34-step SHA-1 as a case study

To illustrate the techniques, we consider SHA-1 reduced to the first 34 steps, and walk through the attack reasoning. We aim for a second-preimage attack on the hash functions, i.e., we require from a characteristic that input- and output chaining do not have a difference. The best characteristic we found for our purpose is the same as the one used by Biham *et al.* [4, Tab. 1] for a collision attack, and is also related to those used in Kim *et al.* [16, Tab. 6], and in [28, Tab. 6]. First, we recompute the probabilities $P_u(i)$ of the differential specified by the message difference m', and the chaining output co' (a zero difference). What we are interested in is the probability that, given an m' from a uniform distribution, $F(m) = F(m \oplus m')$. A good lower bound for this probability is the probability of the particular characteristic as shown in Table 4, which is $2^{-42.42}$.

Table 4. Characteristic with probability $2^{-42.42}$ used for the 34-step (0-33) attack. $P_u(i)$ is written as a log_2, and $N_s(i)$ is written as log_2 as well.

i	∇A_i	∇W_i	$P_u(i)$	$N_s(i)$
-4	--------------------------------			
-3	--------------------------------			
-2	--------------------------------			
-1	--------------------------------			
0	--------------------------------	------------------------------x-	1	42.42
1	------------------------------x-	-------------------------x------	2	41.42
2	--------------------------------	--------------------------------	3	39.42
3	------------------------------x-	x------------------------x------	2	36.42
4	--------------------------------	x-------------------------------	3	34.42
5	------------------------------x-	-------------------------x------	2	31.42
6	--------------------------------	x------------------------------x	2.42	29.42
7	-------------------------------x	-------------------------xx-----	3	27.00
8	--------------------------------	x-----------------------------xx	4	24.00
9	--------------------------------	-x----------------------------x-	2	20.00
10	------------------------------x-	xx-----------------------x------	3	18.00
11	--------------------------------	xx----------------------------x-	4	15.00
12	--------------------------------	x-------------------------------	0	11.00
13	--------------------------------	x-------------------------------	0	11.00
14	--------------------------------	x-----------------------------x-	1	11.00
15	------------------------------x-	-------------------------x------	2	10.00
16	--------------------------------	------------------------------x-	3	8.00
17	--------------------------------	x-------------------------------	0	5.00
18	--------------------------------	x-------------------------------	0	5.00
19	--------------------------------	x-------------------------------	0	5.00
20	--------------------------------	------------------------------x-	1	5.00
21	------------------------------x-	-------------------------x------	1	4.00
22	--------------------------------	--------------------------------	1	3.00
23	------------------------------x-	x------------------------x------	1	2.00
24	--------------------------------	x-----------------------------x-	1	1.00
25	--------------------------------	--------------------------------	0	0.00
26	--------------------------------	x-------------------------------	0	0.00
27	--------------------------------	x-------------------------------	0	0.00
28	--------------------------------	--------------------------------	0	0.00
29	--------------------------------	--------------------------------	0	0.00
30	--------------------------------	--------------------------------	0	0.00
31	--------------------------------	--------------------------------	0	0.00
32	--------------------------------	--------------------------------	0	0.00
33	--------------------------------	--------------------------------	0	0.00
34	--------------------------------			

Taking into account also other, strongly related characteristics with lower probability (see [22,25,28,29] for details), we would arrive at an improved probability of $2^{-42.25}$. The second-preimage finding algorithm hence needs to traverse the first preimage of a length of about $2^{42.25}$ (N_m) message blocks in order to succeed with good probability. The memory requirements for this are negligible as the first preimage can be processed in an on-line manner. In setting 1, when intermediate chaining values are also given, most of the time only the first few step transformations are computed. Hence the computational resources needed in terms of computing step transformations are about an equivalent of $2^{37.87}$ computations of 34-step SHA-1 (N_w according to Eq. 5), taking the early stop technique into account. Without this assumption, the computational effort is hence about $2^{42.25}$ (N_w according to Eq. 7).

Comparisons with results obtained by De Cannière/Rechberger. On one hand, this may be compared with the result from [8], where memory of order 2^{15} and an equivalent of about 2^{77} computations are needed to find a second preimage of 34-step SHA-1 with good probability (a first preimage may be as small as 2^5 message blocks with this approach, but longer first preimages do not help to improve the attack).

Comparison with the generic Kelsey/Schneier 2nd-preimage attack. On the other hand, this may be compared with the generic method of Kelsey and Schneier. In [14], Kelsey and Schneier describe a second preimage attack on iterated hash functions that is independent of the actual compression function. The approach finds a second preimage for a 2^k-message-block message with about $k \times 2^{n/2+1} + 2^{n-k+1}$ work. It was then later generalized to also take, among other aspects, multiple targets into account [1]. Those attacks do not concern our results on the SHA-1 compression function, but need to be taken into account when considering the SHA-1 hash function. The new 2nd-preimage result we described above needs about $2^{42.25}$ message blocks in order to succeed with good probability, i.e. $k = 42.25$. Using the Kelsey/Schneier approach, the resulting attack complexity is of order $42.25 \times 2^{160/2+1} + 2^{160-42.25+1} \approx 2^{118.75}$. Hence, even by neglecting some constants in time complexities comparison, it seems safe to conclude that the proposed differential based method is considerable faster.

5.2 Compression function attacks: 61-step SHA-1

To further illustrate that the availability of a first preimage helps to improve upon current preimage attacks on reduced SHA-1, we also seek to increase the number of steps in which results can be obtained. For this, we relax our requirements on 2nd-preimage attacks in three ways:

1. No practical complexity or probability, better than the ideal 2^{-n} is enough.
2. We do no longer require it to beat the generic Kelsey/Schneier result, i.e. the result will only be valid for the compression function rather the hash function (as Kelsey/Schneier does not apply there).
3. Any choice of consecutive steps is allowed instead of starting with step 0.

By exploiting all those relaxations, we demonstrate attacks for up to 61 steps, thereby having reached more steps than in any compression function attack on SHA-1 before. We used the characteristic given in Table 5. The product of all uncontrolled probabilities P_u suggests a probability of $2^{-158.42}$. However, this does not take the feed-forward operation into account. For the previous example, this was ignored safely, as no probabilistic events happen during the feed forward operation. As can be seen in Table 5 however, we do have a single bit difference

Table 5. Characteristic with probability $2^{-158.42}$ used for the 61-step (18-79) attack. $P_u(i)$ is written as log_2

i	∇A_i	∇W_i	$P_u(i)$
- 4	--------------------------------		
-3	------------------------------x-		
-2	--------------------------------		
-1	--------------------------------		
0	--------------------------------	x-------------------------------	1.00
1	--------------------------------	x-------------------------------	0.00
2	--------------------------------	--------------------------------	0.00
3	--------------------------------	--------------------------------	0.00
4	--------------------------------	--------------------------------	0.00
5	--------------------------------	-x------------------------------	1.00
6	-x------------------------------	----------------------------x---	1.00
7	--------------------------------	-x----------------------------x-	2.00
8	------------------------------x-	x--x---------------------x------	2.00
9	x-------------------------------	-x-x-----------------------x---x	5.00
10	-x-----------------------------x	---x---------------------xx-x---	4.00
11	--------------------------------	xxx---------------------------x-	4.00
12	-------------------------------x	xxxx----------------------x---x-	5.00
13	x-----------------------------x-	-xxx---------------------x-x---x	6.00
14	x-------------------------------	---x-----------------------x----	4.00
15	------------------------------x-	-xx----------------------x-----x	4.00
16	-------------------------------x	xx------------------------x---x-	3.00
17	--------------------------------	x-----------------------------xx	3.00
18	x-----------------------------x-	xxx----------------------x-x--x-	5.00
19	------------------------------x-	xx-----------------------x------	3.00
20	x-----------------------------x-	xxx----------------------x-x--x-	5.00
21	--------------------------------	--x---------------------------xx	4.00
22	x------------------------------x	x-------------------------xx----	5.00
23	--------------------------------	--x----------------------------x	5.00
24	x-------------------------------	xx-------------------------x--x-	5.00
25	------------------------------x-	-xx----------------------x-----x	5.00
26	x------------------------------x	-x------------------------xx--x-	7.00
27	--------------------------------	--x---------------------------xx	6.00
28	------------------------------x-	xx-----------------------x----x-	5.00
29	------------------------------x-	xxx----------------------x----x-	6.00
30	--------------------------------	xxx---------------------------x-	4.00
31	--------------------------------	------------------------------x-	2.00
32	------------------------------x-	-------------------------x------	2.00
33	--------------------------------	x------------------------------x	1.42
34	-------------------------------x	x------------------------xx-----	3.00
35	--------------------------------	x------------------------------x	4.00
36	------------------------------x-	-x-----------------------x----x-	4.00
37	------------------------------x-	xx-----------------------x------	5.00
38	------------------------------x-	-x-----------------------x----x-	4.00
39	--------------------------------	--------------------------------	3.00
40	------------------------------x-	x------------------------x------	2.00
41	--------------------------------	------------------------------xx	2.00
42	-------------------------------x	--------------------------x-----	1.00
43	--------------------------------	x------------------------------x	1.00
44	--------------------------------	xx----------------------------x-	2.00
45	------------------------------x-	-x-----------------------x------	2.00
46	--------------------------------	-x----------------------------x-	2.00
47	--------------------------------	x-----------------------------x-	1.00
48	------------------------------x-	x------------------------x------	1.00
49	--------------------------------	x-----------------------------x-	1.00
50	--------------------------------	x-------------------------------	0.00
51	--------------------------------	x-------------------------------	0.00
52	--------------------------------	x-------------------------------	0.00
53	--------------------------------	--------------------------------	0.00
54	--------------------------------	--------------------------------	0.00
55	--------------------------------	------------------------------x-	1.00
56	------------------------------x-	-------------------------x------	1.00
57	--------------------------------	--------------------------------	1.00
58	------------------------------x-	x------------------------x------	1.00
59	--------------------------------	x-----------------------------x-	1.00
60	--------------------------------	--------------------------------	0.00
61	--------------------------------		

in the chaining input and chaining output. We do require these differences to cancel out during the feed forward operation, which happens with probability 1/2. Hence a lower bound for the probability to indeed have a second preimage is $2^{-159.42}$.

As before, by taking into account also other, strongly related characteristics with lower probability (see [22,25,28,29] for details), we would arrive at an improved probability of $2^{-159.42+1.61} = 2^{-157.81}$. This probability is above the ideal 2^{-160}, hence exhibiting less than ideal 2nd-preimage resistance.

Comparisons. The best results in terms of number of rounds on the SHA-1 compression function following the impossible message approach [8] is 45 steps. Also this approach is not able to take advantage of the relaxation of condition (3) from above. Following the meet-in-the-middle approach, the best result is on 48 steps [3]. There, relaxation of (3) may lead to a slightly better result, but most likely not more than for 1-4 steps.

6 Discussion and open problems

Our results on the second preimage resistance of SHA-1 complement earlier analysis regarding its preimage resistance. Both, attacks for more rounds, and more computationally efficient attacks, can be obtained if the existence of a first preimage (especially if it is long) can be assumed. Our results also complement similar results on the iteration mode [1,14]: also there, better second preimage attacks than preimage attacks were obtained. A lesson to be learned from our results are as follows. In the preimage setting, when it comes to squeezing out the most in terms of number of rounds or in terms of attack complexity, the help provided for an attacker by being given an existing preimage is most of the time not used in earlier preimage-style cryptanalysis of the SHA family.

Overall, applications requiring 2nd-preimage resistance of SHA-1 are not endangered by our results. Even though SHA-1 is arguably one of the more interesting cryptanalytic targets, it will be interesting to see this approach considered for other hash functions as well.

Acknowledgements. The work in this paper has been supported in part by the European Commission under contract ICT-2007-216646 (ECRYPT II) and in part by the IAP Programme P6/26 BCRYPT of the Belgian State (Belgian Science Policy).

References

1. Andreeva, E., Bouillaguet, C., Fouque, P.A., Hoch, J.J., Kelsey, J., Shamir, A., Zimmer, S.: Second Preimage Attacks on Dithered Hash Functions. In: Smart, N. (ed.) Eurocrypt 2008. LNCS, Springer (2008)
2. Andreeva, E., Preneel, B.: A New Three-Property-Secure Hash Function. In: Selected Areas in Cryptography. Lecture Notes in Computer Science, Springer (2008)
3. Aoki, K., Sasaki, Y.: Meet-in-the-Middle Preimage Attacks Against Reduced SHA-0 and SHA-1. In: Halevi [11], pp. 70–89
4. Biham, E., Chen, R., Joux, A., Carribault, P., Lemuet, C., Jalby, W.: Collisions of SHA-0 and Reduced SHA-1. In: Cramer [5], pp. 36–57
5. Cramer, R. (ed.): Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings, LNCS, vol. 3494. Springer (2005)

6. De Cannière, C., Mendel, F., Rechberger, C.: Collisions for 70-Step SHA-1: On the Full Cost of Collision Search. In: Adams, C.M., Miri, A., Wiener, M.J. (eds.) Selected Areas in Cryptography. LNCS, vol. 4876, pp. 56–73. Springer (2007)
7. 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)
8. De Cannière, C., Rechberger, C.: Preimages for Reduced SHA-0 and SHA-1. In: Wagner, D. (ed.) CRYPTO. LNCS, vol. 5157, pp. 179–202. Springer (2008)
9. Dunkelman, O. (ed.): Fast Software Encryption, 16th International Workshop, FSE 2009, Leuven, Belgium, February 22-25, 2009, Revised Selected Papers, Lecture Notes in Computer Science, vol. 5665. Springer (2009)
10. Guo, J., Ling, S., Rechberger, C., Wang, H.: Advanced Meet-in-the-Middle Preimage Attacks: First Results on Full Tiger, and Improved Results on MD4 and SHA-2. Cryptology ePrint Archive, Report 2010/016 (2010), `http://eprint.iacr.org/`
11. Halevi, S. (ed.): Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2009. Proceedings. LNCS, Springer (2009)
12. Hawkes, P., Paddon, M., Rose, G.: Automated Search for Round 1 Differentials for SHA-1: Work in Progress. NIST - Second Cryptographic Hash Workshop, August 24-25 (2006)
13. Joux, A., Peyrin, T.: Hash Functions and the (Amplified) Boomerang Attack. In: Menezes, A. (ed.) CRYPTO. LNCS, vol. 4622, pp. 244–263. Springer (2007)
14. Kelsey, J., Schneier, B.: Second Preimages on n-Bit Hash Functions for Much Less than 2^{n} Work. In: Cramer [5], pp. 474–490
15. Khovratovich, D., Nikolic, I., Weinmann, R.P.: Meet-in-the-Middle Attacks on SHA-3 Candidates. In: Dunkelman [9], pp. 228–245
16. Kim, J., Biryukov, A., Preneel, B., Hong, S.: On the Security of HMAC and NMAC Based on HAVAL, MD4, MD5, SHA-0 and SHA-1 (Extended Abstract). In: Prisco, R.D., Yung, M. (eds.) SCN. Lecture Notes in Computer Science, vol. 4116, pp. 242–256. Springer (2006)
17. Knudsen, L.R., Mathiassen, J.E., Muller, F., Thomsen, S.S.: Cryptanalysis of MD2. J. Cryptology 23(1), 72–90 (2010)
18. Leurent, G.: Message Freedom in MD4 and MD5 Collisions: Application to APOP. In: Biryukov, A. (ed.) FSE. LNCS, vol. 4593, pp. 309–328. Springer (2007)
19. Leurent, G.: MD4 is Not One-Way. In: Nyberg, K. (ed.) FSE. LNCS, vol. 5086, pp. 412–428. Springer (2008)
20. Matusiewicz, K., Pieprzyk, J.: Finding Good Differential Patterns for Attacks on SHA-1. In: Ytrehus, Ø. (ed.) WCC. Lecture Notes in Computer Science, vol. 3969, pp. 164–177. Springer (2005)
21. McDonald, C., Pieprzyk, J., Hawkes, P.: SHA-1 collisions now 2^{52}. Eurocrypt 2009 Rump Session (2009)
22. Mendel, F., Pramstaller, N., Rechberger, C., Rijmen, V.: The Impact of Carries on the Complexity of Collision Attacks on SHA-1. In: Robshaw, M.J.B. (ed.) FSE. LNCS, vol. 4047, pp. 278–292. Springer (2006)
23. National Institute of Standards and Technology: NIST's Policy on Hash Functions (2008), available online at `http://csrc.nist.gov/groups/ST/hash/policy.html`
24. National Institute of Standards and Technology (NIST): FIPS-180-2: Secure Hash Standard (August 2002), available online at `http://www.itl.nist.gov/fipspubs/`
25. Peyrin, T.: Analyse de fonctions de hachage cryptographiques. Ph.D. thesis (2008)

26. Pramstaller, N., Rechberger, C., Rijmen, V.: Exploiting Coding Theory for Collision Attacks on SHA-1. In: Smart, N.P. (ed.) IMA Int. Conf. Lecture Notes in Computer Science, vol. 3796, pp. 78–95. Springer (2005)
27. Rechberger, C.: Preimage Search for a Class of Block Cipher based Hash Functions with Less Computation. Unpublished manuscript (2008)
28. Rechberger, C., Rijmen, V.: On Authentication with HMAC and Non-random Properties. In: Dietrich, S., Dhamija, R. (eds.) Financial Cryptography. Lecture Notes in Computer Science, vol. 4886, pp. 119–133. Springer (2007)
29. Rechberger, C., Rijmen, V.: New Results on NMAC/HMAC when Instantiated with Popular Hash Functions, journal = J. UCS 14(3), 347–376 (2008)
30. Reyhanitabar, M.R., Susilo, W., Mu, Y.: Enhanced Target Collision Resistant Hash Functions Revisited. In: Dunkelman [9], pp. 327–344
31. Rijmen, V., Oswald, E.: Update on SHA-1. In: Menezes, A. (ed.) CT-RSA. LNCS, vol. 3376, pp. 58–71. Springer (2005)
32. Rivest, R.L.: The MD4 Message Digest Algorithm. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO. LNCS, vol. 537, pp. 303–311. Springer (1990)
33. Rogaway, P., Shrimpton, T.: Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In: Roy, B.K., Meier, W. (eds.) FSE. LNCS, vol. 3017, pp. 371–388. Springer (2004)
34. Sasaki, Y., Aoki, K.: Preimage Attacks on 3, 4, and 5-Pass HAVAL. In: Pieprzyk, J. (ed.) ASIACRYPT. LNCS, vol. 5350, pp. 253–271. Springer (2008)
35. Sasaki, Y., Aoki, K.: Finding Preimages in Full MD5 Faster Than Exhaustive Search. In: Joux, A. (ed.) EUROCRYPT. LNCS, vol. 5479, pp. 134–152. Springer (2009)
36. Sasaki, Y., Wang, L., Ohta, K., Kunihiro, N.: Security of MD5 Challenge and Response: Extension of APOP A Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings. In: CT-RSA. Lecture Notes in Computer Science, vol. 4964. Springer (2008)
37. Stevens, M., Lenstra, A.K., de Weger, B.: Chosen-Prefix Collisions for MD5 and Colliding X.509 Certificates for Different Identities. In: Naor, M. (ed.) EUROCRYPT. LNCS, vol. 4515, pp. 1–22. Springer (2007)
38. Stevens, M., Sotirov, A., Appelbaum, J., Lenstra, A., Molnar, D., Osvik, D.A., de Weger, B.: Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate. In: Halevi [11], pp. 55–69
39. Stinson, D.R.: Some Observations on the Theory of Cryptographic Hash Functions. Des. Codes Cryptography 38(2), 259–277 (2006)
40. 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)
41. Wang, X., Yu, H.: How to Break MD5 and Other Hash Functions. In: Cramer [5], pp. 19–35
42. Yajima, J., Iwasaki, T., Naito, Y., Sasaki, Y., Shimoyama, T., Peyrin, T., Kunihiro, N., Ohta, K.: A Strict Evaluation on the Number of Conditions for SHA-1 Collision Search (2009)
43. Yajima, J., Sasaki, Y., Naito, Y., Iwasaki, T., Shimoyama, T., Kunihiro, N., Ohta, K.: A New Strategy for Finding a Differential Path of SHA-1. In: Pieprzyk, J., Ghodosi, H., Dawson, E. (eds.) ACISP. LNCS, vol. 4586, pp. 45–58. Springer (2007)
44. Yu, H., Wang, G., Zhang, G., Wang, X.: The Second-Preimage Attack on MD4. In: Desmedt, Y., Wang, H., Mu, Y., Li, Y. (eds.) CANS. LNCS, vol. 3810, pp. 1–12. Springer (2005)