A Literature Review on Differential Cryptanalysis of SHA-1

Zhuolun Li
zhuolun.li.21@ucl.ac.uk
University College London

April 3, 2022

Disclaimer This report is submitted as part requirement for the module COMP0057 Research in Information Security at University College London. It is substantially the result of my own work except where explicitly indicated in the text. The report may be freely copied and distributed provided the source is explicitly acknowledged.

Abstract. SHA-1 is a widely used cryptographic hash function that was deprecated by NIST in 2011 due to potential weaknesses [2]. In the past decade, researchers further reduced the complexity of finding collisions in SHA-1 using cryptanalysis techniques. Among various cryptanalysis techniques, differential cryptanalysis is the most important one when finding SHA-1 collisions.

Differential cryptanalysis was firstly used in analysing theoretical weakness for DES (Data Encryption Standard) in 1993 [7]. To apply differential cryptanalysis in SHA-1, a number of techniques and modifications has been introduced. This report reviews the the differential cryptanalysis techniques and attack process of finding SHA-1 collisions, and discusses the result of the existing SHA-1 collision attacks. Additionally, this report briefly discusses the security of newer hash functions and the directions of future works.

1 Introduction

1.1 SHA-1 and Differential Cryptanalysis

A cryptographic hash function takes an arbitrarily long input and produces a fixed-length output. Cryptographic hash functions are used as the fundamental building blocks for many systems, such as content-addressable storage and blockchains. A vital property of a cryptographic hash function is collision resistance, namely, it should be difficult to find two distinct inputs that produce the same output. Before considering any other attacks, cryptographic hash functions should not be vulnerable against the birthday attack by design, which is a brute-force search that takes $2^{n/2}$ computations, where n is the length of the output, according to the birthday paradox. Except for the birthday attack, researchers also use cryptanalysis techniques to analyse hash functions and study potential vulnerabilities.

SHA-1 is a widely used cryptographic hash function in the past two decades. As SHA-1 outputs 160-bit message digests, according to the birthday attack, SHA-1 collision can be found with a cost of 2^{80}, and should not be vulnerable against attacks that cost below 2^{80}. However, SHA-1 had been found to be theoretically vulnerable against differential cryptanalysis with the cost of 2^{69} calls since 2005 [27]. Due to this potential vulnerability, NIST deprecated SHA-1 in 2011 [2].

Following the first theoretical attack by Wang et al. [27], all existing literature for finding SHA-1 collisions used similar differential cryptanalysis approaches. Informally, differential cryptanalysis is a kind of cryptanalysis technique that studies how differences in the input messages affect each of the intermediate states and the final outputs. It studies differences between two paths instead of a single path. Differential cryptanalysis is initially used to analyse DES [7], a symmetric encryption scheme, but it is also applicable to hash functions.

1.2 Progress of SHA-1 Collision Search

After the ground-breaking work by Wang et al. in 2005 with a complexity of 2^{69} [27], more advanced differential cryptanalysis techniques were introduced and applied to SHA-1 or reduced versions of SHA-1 to reduce the complexity of finding identical-prefix collisions [3, 5, 8, 9, 11, 22, 24]. In 2017, Stevens et al. [23] achieved the milestone of producing the first instance of SHA-1 identical-prefix collision attack with a cost of $2^{64.7}$. They also made use of the colliding messages to produce two distinct PDFs with the same SHA-1 message digest. This achievement is the result of the boost of computing power over the years and the development of differential cryptanalysis techniques. More recently, the cost was reduced to $2^{61.2}$ in 2020 by Leurent and Peyrin [16], where they also managed to produce a practical chosen-prefix collision with a complexity of $2^{63.4}$.

The progress of SHA-1 collision attacks can not only be represented in complexity, but also cost of money. Schneier estimated the cost of producing SHA-1 attack in 2012 [20], which indicated that the cost of finding SHA-1 collision should drop from 2.77M US dollars in 2012 to 43K US dollars in 2021, due to the improvement in hardware performance. It turns out that Schneier's prediction was quite accurate, as the cost of Leurent and Peyrin's work in 2020 [16] was close to the predicted number.

1.3 Report Organisation

The report will be organised in the following manner. Firstly, this report provides a background of SHA-1 and differential cryptanalysis by looking at its development. This will be followed by a detailed study of how differential cryptanalysis is applied to SHA-1 and what techniques are developed to speed-up the search for collisions. After the theoretical studies, this report goes into the attack implementation and attack results. In the end, the report discusses the future research directions by looking at differential cryptanalysis in SHA-2 and SHA-3.

2 Background

2.1 SHA-1 Overview

SHA-1 is a cryptographic hash function that takes an arbitrarily long input message and produces a 160-bit message digest. The high-level process of SHA-1 is as follows.

1. The message is padded into the multiply of 512 bits, the padding is a 1 followed by a number of 0 bits depending on the required length.
2. The message is then divided into blocks, each block is 512 bits long.
3. For each block, the 512-bit message and the 160-bit chaining value (CV) of the previous block act as the input of the compression function (h) to produce the output chaining value of this block, where the first chaining value (CV_0) is a fixed initial vector.

$$CV_j = h(CV_{j-1}, M_j),\ CV_0 = 0x67452301EFCDAB8998BADCFE10325476C3D2E1F0$$

4. The output chaining value of the last block becomes the message digest.

The detailed process of the compression function is as follows:

1. The 512-bit message block is divided into 16 words $m_1, m_2, ..., m_{16}$, each word is 32-bit long. The 16 words then go through a linear expansion to obtain 80 words with the following equation:

$$m_i = (m_{i-3} \oplus m_{i-8} \oplus m_{i-14} \oplus m_{i-16})^{\circlearrowleft 1},\ 16 \leq i < 80$$

2. The 160-bit input chaining value is also divided into 5 words represented by a, b, c, d, e.
3. The main loop is 80 rounds, each round takes a word from the expanded message as input to update the chaining value a, b, c, d, e. The pseudocode is given below in algorithm 1.
4. Compute the sum of the output chaining value of the loop and the input chaining value of this block, the result is the output chaining value of this block.

Algorithm 1 SHA-1 Compression Function Main Loop

```
for $0 \leq i \leq 79$ do
    if $i < 20$ then
        f = (b and c) or ((not b) and d)
        k = 0x5A827999
    end if
    if $i < 40$ then
        f = b xor c xor d
        k = 0x6ED9EBA1
    end if
    if $i < 60$ then
        f = (b and c) or (b and d) or(c and d)
        k = 0x8F1BBCDC
    end if
    if $i < 80$ then
        f = b xor c xor d
        k = 0xCA62C1D6
    end if
    $temp = a^{\circlearrowleft 5} + f + e + k + m_i$
    $e = d,\ d = c,\ c = b^{\circlearrowleft 30},\ b = a,\ a = temp$
end for
```

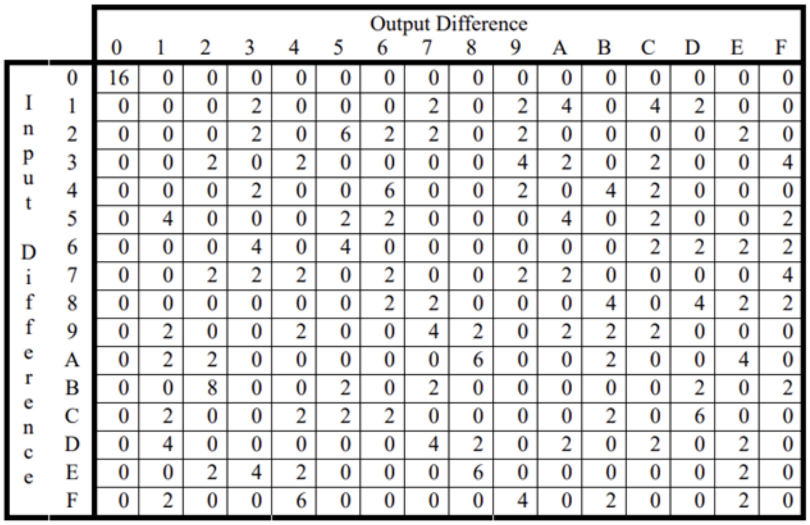
2.2 Differential Cryptanalysis

Differential cryptanalysis is a cryptanalysis technique that was initially invented and used to study the data encryption standard (DES) [6]. Later in 2005, it was also used to break hash functions like MD5 and SHA-0 [28,29]. In the same year, the idea and process of breaking SHA-0 were also applied to constructing the first theoretical attack on SHA-1 [27]. Since then, the works on finding SHA-1 collisions all followed the similar path of using differential cryptanalysis to reduce the search space.

The idea of differential cryptanalysis is to have two distinct messages m and m' as inputs of a cryptographic protocol, and observe how the differences propagate at each step. By design, a cryptographic hash function attempts to create an avalanche effect, such that a small difference in input can result in a completely different output, which makes the output pseudorandom. However, this pseudorandomness is achieved by many steps of operations, and at each single step, the difference propagation is not completely random. More concretely, in every single step of a hash function, some output differences are more likely to occur than other output differences, depending on the input difference.

It would be easier to consider a concrete example. The image 1 below is a difference disturbance table. A difference disturbance table is a tool in differential cryptanalysis that is used to illustrate the relationship between the input difference and the output difference of a certain step in a cryptographic scheme. For example, in this example table, one can observe that if the input difference is 1 for two input messages, the output after this step has no chance to be 1, 2, 4, 5, etc., as shown in the second row of the table. Similarly, while an input difference of 1 can result in an output difference of both 9 and A, the probability of ending up in A is higher than ending up in 9. The Differential cryptanalysis makes use of these uneven difference distributions to narrow the search space, such that the probability of finding collisions is higher in this sub-space.

To understand how the uneven difference distributions turn into collision attacks, there are several concepts in differential cryptanalysis to understand in advance.



	Output Difference															
Input Difference	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Image 1: An example difference distribution table [13]

Disturbance Vectors In differential cryptanalysis, the difference between the two inputs m and m' is represented by a disturbance vector (DV), where:

$$DV = m \oplus m'$$

Therefore, each 1 bit in a disturbance vector marks the location where the messages are different. A disturbance vector defines a message space for finding collisions with relatively high probability.

In SHA-1, a disturbance vector has the same length as an expanded message block, and it is a correctly expanded message itself [27], because the xor of two correctly expanded messages must also be a correctly expanded message, which can be deducted using the message expansion equation (see 2.1 compression function step 1). A disturbance vector in SHA-1 also marks the locations where the chaining value starts to get different, because each round in SHA-1 takes one message word only, therefore the chaining value only starts to get different when the input message word is different.

Local Collisions A local collision is a collision that starts and ends in a small number of steps. When the state is different, one can construct the message in the next few steps carefully such that the difference is cancelled out. In the process of finding full collisions for SHA-1, local collisions are used as building blocks to cancel out chaining value differences.

In SHA-1, a local collision is achievable in 6 steps starting at any steps where the message is under controlled [27].

Differential Paths A differential path is a vector that describes in which states the chaining value is different. Informally, a differential path is the combination of a disturbance vector and local collisions. That is because a disturbance vector marks where the chaining values start to get different, and local collisions decide when the differences are cancelled out.

In SHA-1, the step of message expansion decides that it is difficult to control the state through the message after 16 rounds. Therefore, it is difficult to construct an achievable differential path of full 80 rounds since the states of later steps are very uncontrollable.

To conclude, differential cryptanalysis makes use of the uneven difference distributions in the process of a cryptographic scheme to find a good disturbance vector and construct a differential path, then applies these constraints to attack the cryptographic scheme.

3 Finding SHA-1 Collisions with Differential Cryptanalysis

This section reviews the existing works in detail on how differential cryptanalysis is used in finding SHA-1 collisions. Since the existing works all follow a similar attack process, the report introduces the overall attack process first, followed by the discussion of the detailed approaches at each step.

3.1 Attack Process

Since 2005 when Wang et al. published the first theoretical attack [27], later works on SHA-1 collisions followed the overall construction and attempted to optimize the techniques at each step to reduce the attack complexity [16, 23]. The shared attack process overview is as follows.

The first step is to find a good disturbance vector. To find a good disturbance vector, hamming weight is used as an important measurement [5, 27]. Hamming weight is the number of 1 bits in the disturbance vector. Disturbance vectors with lower hamming weight are preferred, because it means lower search space. At the same time, the location of differences should be carefully selected such that the probability of finding collisions is high. Finding an achievable disturbance vector requires imposing several constraints, in order to relax the constraints in finding disturbance vectors, the existing works try to find two-block collisions instead of one-block collisions, which means the collisions are achieved in two message blocks.

Secondly, once the disturbance vector is set, a differential path is required. Constructing a differential path can further reduce the search space. However, constructing a valid differential path is non-trivial. In 2005, Wang et al. manually constructed the first differential path for SHA-1 by specifying several heuristics and relying on intuition [27]. Starting from 2006, some automated searching algorithms were developed, which will be discussed in the subsection below regarding differential path constructions.

Thirdly, with the disturbance vector and the differential path, the attack conditions can be determined. The disturbance vector and the differential path are encoded into constraints to prepare for the collisions search. Some of these constraints are on the message bits suggested by the disturbance vector, some are on the intermediate state bits suggested by the differential path.

Lastly, to further improve the efficiency of finding collisions, some additional speed-up techniques are applied. In particular, message modification [26, 28] is the one that can help. Message modification is a technique that modifies certain carefully selected bits of the message to produces controllable changes up to some steps in the hash function. This technique and some of its variants were used to speed up the attacks, which will be discussed in the corresponding section.

3.2 Two-block Collisions

Selecting a good disturbance vector is difficult, because, in SHA-1, a disturbance vector needs to satisfy some conditions, in order to make sure that it is possible to construct local collisions that cancel the differences and produce a collision within a limited number of steps [10, 27]. However, when these conditions are imposed, it is very difficult to find a disturbance vector with a low hamming weight, which means there will be many non-zero bits in the disturbance vector. This means the two messages have a lot of differences, hence a larger search space that is beyond the theoretical bound to compute [27].

To solve this problem, Wang et al. [27] proposed to use two-block collisions in their theoretical attack of finding SHA-1 collisions. Two-block collision was firstly introduced in 2004 in finding collisions for SHA-0 and SHA-1 [5,14,18]. A two-block collision means that the collision is achieved in two consecutive blocks. The state difference at the end of the first block is passed into the second block, which will be cancelled at the end of the second block. The first block is also called the near-collision block, because the output of the first block is close to a collision but not a collision. Compared to one-block collisions, there are more available steps to handle the differences, hence relaxing the constraints when finding searching for a disturbance vector with lower hamming weight. This approach is also used in later research that produces practical attacks [16, 23].

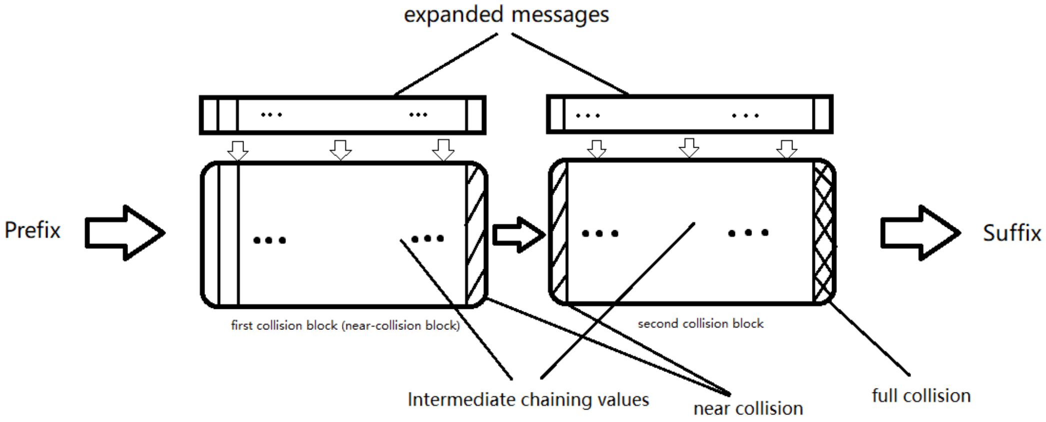


Image 2: two-block collision illustration

3.3 Disturbance Vector Selection

When Wang et al. published the first theoretical attack in 2005 [27], they used the heuristic that the non-zero bits of the disturbance vector are likely to be in consecutive positions. In 2009, an improved version of disturbance search was proposed by Yajima et al. [30] that applied additional searching conditions that makes use of the venerability of the Boolean functions in the compression function. However, these approaches are not used to construct practical attacks due to the more efficient algorithm proposed later.

The best known approach to find a good disturbance is joint-local-collision analysis (JLCA) [22]. Instead of only trying to build disturbance vectors with low hamming weights, JLCA takes one step further by evaluating the possible differential paths up to a specified step given a disturbance vector. Namely, in JLCA, the standard for finding disturbance vectors in JLCA is that they should have low hamming weights and there should be good differential paths given this disturbance vector. This approach takes in the allowed differences in both the intermediate chaining values and the input messages to go through all possibilities of differential paths up to some steps, and evaluate the probability of successfully finding collisions. This approach is used by Stevens et al. in 2017 [23] and Leurent and Peyrin in 2020 [16] when constructing practical attacks.

3.4 Differential Path Construction

In SHA-1, a differential path is usually constructed for the first 16 or 22 steps only [23, 27], because in later steps one does not have control over the message, as they are from the expansion algorithms. In the first 16 steps, the message is fully under control, hence the states can be easily controlled, after which the state will make uncontrollable changes. However, one can extend a few more steps of control by applying some special techniques, these techniques include JLCA (see 3.3), which can run through more steps for searching differential paths, and also a technique called boomerangs which will be discussed in section 3.6.

Since the existing works use two-block collisions, they had to construct two differential paths, one for each collision block. The conditions for a finding differential path for the first block and for the second block are different [23]. For the first collision block, one can make use of the previous block to pass in a desired input chaining value, hence reducing the difficulty of finding a good differential path. However, the second block must start at a specific chaining value that is passed in from the first block, thereby increasing the computation needed.

As mentioned above in section 3.1, The first differential path for SHA-1 was constructed by Wang et al. manually [27]. They search for differential paths by creating several strategies in finding differential paths that make use of the differential properties of SHA-1. However, their approach is not detailed and it is partly based on intuition. Although the work of Wang et al. was groundbreaking, their method of creating differential paths was not used in later works. But it is worth noticing that in their approach, they add -1 as a possible value to the bits in differential paths, and this concept was used in the automated algorithms created by later works. The reason to add -1 as a possible value is the fact that not only does whether there is a difference matter, but what is the exact difference also matters. In their representation, 0 bits still represent no differences in that bit position; 1 or -1 bits not only means there is a difference, but also whether their value is (1,0), or (0,1).

The first automated search for SHA-1 differential paths was created in 2006 by De Cannière and Rechberger [9]. The approach is operated in bit pairs, namely, the two bits at the same location for two distinct input messages. starting from no value restrictions on the bit pairs, it continuously applies restrictions on a bit pair and propagates the effect to other bit pairs. It stops a branch when contradictions of the restrictions are found, then backtracks and goes for another restriction. Using this approach, De Cannière and Rechberge successfully created a collision on a reduced version of SHA-1.

More recent approaches are meet-in-the-middle approaches [21, 25, 31]. These approaches make use of the message expansion process of SHA-1. The message expansion process (see section 2.1, step 1 of the compression function) can not only go forward but also go backward, which means it is possible to create m_i for $i < 0$. More importantly, when more than 80 words are created, any continuous 80 words are a valid expanded message since they all satisfy the equation. With this property, Yajima et al. [31] expand the message in both directions and create two differential paths at two ends respectively, then attempts to connect the two paths in the middle by iterating through all possibilities. Stevens [21, 25] also made use of this property to create differential path searching algorithms. His approach evaluates a large set of possible differential paths step by step, and only keeps a number of good paths at each step. His approach was used in finding practical collisions in 2017 [23] and 2020 [16].

3.5 Attack conditions encoding

With disturbance vectors and differential paths selected, the next step is to encode them into constraints on the search for collisions. These conditions are also called systems of equations [23], which consist of equations about the messages and equations about the intermediate states. The equations about the messages are derived from the disturbance vectors, and the equations about the states are from the differential paths.

In general, the more constraints the system of equations gives, the faster it is to search through the possibilities, because the search space is lower and the search can stop earlier in the cases of contradictions. But having too many constraints is not necessarily good. When Stevens et al. [23] encoded their equations, they applied too many constraints at the first time, which led to the failure of their first attack attempt. They went through all allowed possibilities and none of the messages fulfilled all constraints. That is because the differential path they used only makes sure that the probability of finding a collision is high up to some steps. When too many constraints are applied, the chance that some of the constraints are bad or contradictory to each other gets higher.

3.6 Speed-up Approaches

To further speed up the attack, the existing works applied message modifications in the searching process [16,23,27]. Message modification is a technique that modifies the input messages in a way that the constraints are not affected up to some steps [26,28]. Namely, the modified bits do not involve in the system of equations. When one solution is found up to some steps, this technique efficiently allows to generate more solutions up to the same step with very little cost.

The first message modification approach is called neutral bits that is proposed by Biham and Chen in 2004 [4]. Fora bit in the message, if flipping this bit does not affect the conditions up to some step with a high probability, these bits are called neutral bits.

In 2007, a new variant of message modification called boomerangs was proposed by Joux and Peyrin [15], which was later used in the SHA-1 practical collision works [16,23]. The technique of boomerangs flips several bits that are carefully selected, such that the intermediate chaining values barely change in the first 16 steps, where the message is in control. The differences produced by such modification can only be controlled up to the first 16 steps, after which the differences will be uncontrollable. However, this is not concerning because no conditions are imposed in later steps. Compared to neutral bits, boomerangs can further delay the uncontrollable changes [16].

4 Attack Implementations

Once all conditions are specified, and having the speed-up approaches introduced, the final step is to implement the attack. There are two implementations that managed to produce actual collisions for full SHA-1. The first one is by Stevens et al. in 2017 [23], and the second one is by Leurent and Peyrin in 2020 [16]. Even though the first theoretical attack was already proposed in 2005 [27] and the overall methodologies of the attack are similar, it was not until 12 years later then the first practical attack has been implemented. That is because the boost of computing power over the years plays an important role in the success as well. In 2012, Schneier [20] estimated the cost to produce SHA-1 collisions, the development of hardware was estimated to help to decrease the cost from 2.77M US dollars in 2012 to 43K US dollars in 2021, without considering the progress of the attack method. In reality, Leurent and Peyrin managed to produce a chosen-prefix collision with 75k US dollars in 2020, which is fairly close to the estimation, considering it is a chosen-prefix attack in 2020.

Stevens et al. [23] implemented a distributed network that searches for different values simultaneously. They searched for collisions with both CPUs and GPUs, using them to work on different parts of the computation. They computed partial solutions of the first few steps with CPUs, and extend the potential solutions up to more steps with GPUs, then check whether they result in a full collision with CPUs. Using different hardware in different steps is because the types of computations are different. For example, in the first few steps where there are constraints encoded, there are more general computations that CPUs are better at.

The latest attack from Leurent and Peyrin [16] followed the work from Stevens et al. in most of the steps. But by mainly improving on the hardware implementation and design of the computation network, for example, they found some additional boomerangs to speed up the attack. They also managed to further reduce the cost of the attack and the complexity compared to the work by Stevens et al., and also produced a chosen-prefix collision attack. One of the key reasons they managed to reduce the cost of attack is their choices of hardware. For example, considering that the GPUs do not require frequent communications, they managed to reduce the cost by using cheaper GPUs to reduce cost.

5 Attack Results

The two practical works from Stevens et al. [23] and Leurent and Peyrin [16] successfully constructed colliding messages. Additionally, they made use of the colliding messages to create attacks on the application layer.

By creating a two-block SHA-1 collision, Stevens et al. [23] managed to craft two PDF files (see image 3) with distinct appearance that have the same SHA-1 message digest. More interestingly, with the same colliding message, their approach can be extended to build colliding PDF files with any distinct appearances.

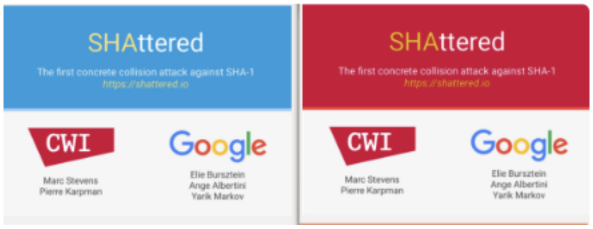


Image 3: screenshots of two colliding PDF files [1]

To do so, they put all the data in the identical suffix of both files and choose what data to display in the blocks of collisions. For example, for the above two PDF files, they have both the blue image and the red image stored in both PDF files, since one can put any identical data after the colliding blocks. In the two colliding blocks, they use the difference to specify which element in the suffix should be displayed. Therefore, this approach is applicable to any PDF file.

Even more exploitation can be done with the chosen-prefix collision produced by Leurent and Peyrin [16]. With the collision, they managed to compromise the PGP Web of Trust. They managed to collide the identity certificates issued by the Web of Trust, such that the PGP key for one user can be transferred to a forged key of another user. That was done by the attacker submitting a key and a crafted JPEG image such that the certificate of the victim collides with the certificate of the attacker's key plus the image. This is also the first known attack that makes use of the colliding messages to compromise a real-world application.

6 Future Works

In less than 20 years, the existing works have successfully proved SHA-1 is not secure, not only theoretically but also practically. The direction of future works has moved to finding collisions for other hash functions, for example, SHA-2 and SHA-3. Fortunately, SHA-2 and SHA-3 seem to be resistant to differential cryptanalysis, since there is not much progress in attacking SHA-2 and SHA-3 over the past ten years. There are works on reduced versions of SHA-2 and SHA-3 [12, 17, 19], but they are still far from being vulnerabilities.

A potential reason is that differential cryptanalysis is not suitable to attack SHA-2 and SHA-3, as it is initially designed to attack DES. Therefore, it is reasonable to consider developing new cryptanalysis techniques that are more effective against newer hash functions.

Quantum-resistant hash functions and quantum attacks are also an important direction for future works, as the development of quantum computers is rapid. The effect of quantum computers on hash functions should be studied more carefully, in order to respond to the related threats in advance.

7 Conclusion

This report summarises the existing collision attacks of SHA-1. These attacks are constructed using advanced differential cryptanalysis techniques to reduce the search space. This report reviews these techniques and goes through the development of SHA-1 collision attacks from theoretical attacks to practical attacks in less than 20 years. It is also interesting to see that with two colliding messages, one can craft different colliding PDF files and have the ability to compromise real-world applications. It is clear that SHA-1 is unsafe and should not be used anymore. Fortunately, there are safe alternatives available, for example, SHA-2, SHA-3, ECDSA and so on.

Even though SHA-1 it is deprecated in 2011, it was still widely used after deprecation and it still acts as an important part of many applications such as PGP and Git. Therefore, it is also alarming that applications should not assume a component is safe forever, everything should be upgradable and constantly updated to prevent security issues.

References

1. SHAttered. https://shattered.io/, accessed: 2022-4-2
2. Barker, E., Roginsky, A.: Transitions: Recommendation for transitioning the use of cryptographic algorithms and key lengths (Jan 2011), https://csrc.nist.gov/publications/detail/sp/800-131a/archive/2011-01-13
3. Biham, E.: New results on sha-0 and sha-1. Crypto 2004 Rump Session (2004)
4. Biham, E., Chen, R.: Near-collisions of sha-0. In: Annual International Cryptology Conference. pp. 290–305. Springer (2004)
5. Biham, E., Chen, R., Joux, A., Carribault, P., Lemuet, C., Jalby, W.: Collisions of sha-0 and reduced sha-1. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 36–57. Springer (2005)
6. Biham, E., Shamir, A.: Differential cryptanalysis of the full 16-round des. In: Annual international cryptology conference. pp. 487–496. Springer (1992)
7. Biham, E., Shamir, A.: Differential cryptanalysis of the data encryption standard (1993)
8. Cannière, C.D., Mendel, F., Rechberger, C.: Collisions for 70-step sha-1: on the full cost of collision search. In: International Workshop on Selected Areas in Cryptography. pp. 56–73. Springer (2007)
9. Cannière, C.D., Rechberger, C.: Finding sha-1 characteristics: General results and applications. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 1–20. Springer (2006)
10. Chabaud, F., Joux, A.: Differential collisions in sha-0. In: Annual International Cryptology Conference. pp. 56–71. Springer (1998)
11. Grechnikov, E.A.: Collisions for 72-step and 73-step sha-1: Improvements in the method of characteristics. Cryptology ePrint Archive (2010)
12. Guo, J., Liao, G., Liu, G., Liu, M., Qiao, K., Song, L.: Practical collision attacks against round-reduced sha-3. Journal of Cryptology 33(1), 228–270 (2020)
13. Heys, H.M.: A tutorial on linear and differential cryptanalysis. Cryptologia 26(3), 189–221 (2002)
14. Joux, A.: Collisions for sha-0. CRYPTO 2004 rump session (Aug.) (2004)
15. Joux, A., Peyrin, T.: Hash functions and the (amplified) boomerang attack. In: Annual International Cryptology Conference. pp. 244–263. Springer (2007)
16. Leurent, G., Peyrin, T.: Sha-1 is a shambles. In: USENIX 2020-29th USENIX Security Symposium (2020)
17. Mendel, F., Nad, T., Schläffer, M.: Finding sha-2 characteristics: searching through a minefield of contradictions. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 288–307. Springer (2011)
18. Phan, R.C.W.: Impossible differential cryptanalysis of 7-round advanced encryption standard (aes). Information processing letters 91(1), 33–38 (2004)
19. Sanadhya, S.K., Sarkar, P.: New collision attacks against up to 24-step sha-2. In: International conference on cryptology in India. pp. 91–103. Springer (2008)
20. Schneier, B.: When will we see collisions for sha-1? https://www.schneier.com/blog/archives/2012/10/when_will_we_se.html (2012), accessed: 2022-4-1
21. Stevens, M.: Project HashClash. https://marc-stevens.nl/p/hashclash/, accessed: 2022-4-1
22. Stevens, M.: New collision attacks on sha-1 based on optimal joint local-collision analysis. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 245–261. Springer (2013)
23. Stevens, M., Bursztein, E., Karpman, P., Albertini, A., Markov, Y.: The first collision for full sha-1. In: Annual international cryptology conference. pp. 570–596. Springer (2017)
24. Stevens, M., Karpman, P., Peyrin, T.: Freestart collision for full sha-1. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 459–483. Springer (2016)
25. Stevens, M.M.J.: Attacks on hash functions and applications. Leiden University (2012)
26. Wang, X., Lai, X., Feng, D., Chen, H., Yu, X.: Cryptanalysis of the hash functions md4 and ripemd. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 1–18. Springer (2005)
27. Wang, X., Yin, Y.L., Yu, H.: Finding collisions in the full sha-1. In: Annual international cryptology conference. pp. 17–36. Springer (2005)
28. Wang, X., Yu, H.: How to break md5 and other hash functions. In: Annual international conference on the theory and applications of cryptographic techniques. pp. 19–35. Springer (2005)
29. Wang, X., Yu, H., Yin, Y.L.: Efficient collision search attacks on sha-0. In: Annual International Cryptology Conference. pp. 1–16. Springer (2005)
30. 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. IEICE transactions on fundamentals of electronics, communications and computer sciences 92(1), 87–95 (2009)

31. 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: Australasian Conference on Information Security and Privacy. pp. 45–58. Springer (2007)