Analyze SHA-1 in message schedule

Yi-Shiung Yeh [1, *]

Ting-Yu Huang [1, †]

I-Te Chen [2, ‡]

Shih-Chin Chou [1, §]

[1] *Department of Computer Science and Information Engineering*
National Chiao-Tung University
1001 Ta-Hsueh Road
HsinChu
Taiwan 30050
R.O.C.

[2] *General Education Center*
Kaohsiung Medical University
Taiwan
R.O.C.

Abstract

Wang *et al* have found a family of collisions in MD5. They reported their method to find a collision efficiently in SHA, and also to find a collision in SHA-1 within 2^{69} hash steps in February 2005. In fact, we can still discover the decay phenomenon with the application of a message schedule's judgment proposed in this work when inspection how SHA-1 generates message schedule actually. Therefore, we would like to introduce two SHA-1 corrections to enhance the security of SHA-1.

Keywords : *Cryptography, hash functions, SHA-*1*, message schedule, collisions*

Introduction

In 1998, F. Chabaud and A. Joux presented a method to find collisions in SHA with complexity 2^{61} [2]. In 2004 crypto conference and in February

* *E-mail*: `ysyeh@csie.nctu.edu.tw`

† *E-mail*: `tingyu@csie.nctu.edu.tw`

‡ *E-mail*: `itchen@kmu.edu.tw`

§ *E-mail*: `scchou@csie.nctu.edu.tw`

Journal of Discrete Mathematical Sciences & Cryptography
Vol. ( ), No. , pp. 1–7

2005, Wang *et al* [5] developed efficient methods to find collisions in MD5, as well as in SHA-1 with time complexity of 2^{39} and 2^{69} hash steps respectively. Furthermore, Biham and Chen [1] announced new analytical discoveries concerning SHA-1. Their results include a collision in a reduced-round version of SHA-1, which can be found less than 40 rounds.

Suppose the output size of the hash function is n-bit. According to the birthday paradox attack property, we could expect certain collisions after trying $2^{n/2}$ possible input values. Van Oorschot and Wiener [4] have explained how such a brute-force attack might be implemented. That implies any cryptanalysis method with higher complexity than the birthday paradox attack will be regarded as inefficient.

F. Chabaud and A. Joux find collision in SHA with 2^{61} complexity, related to differential cryptanalysis of block ciphers [2]; and their method is theoretically faster than birthday paradox attack. Unfortunately, in SHA-1, their method is unable to detect collision faster than the birthday paradox attack.

One reason from SHA to SHA-1

Firstly, we define notation $x^n = \mathrm{ROTL}^{n \bmod 32(x)}$. The message schedule of w_t of SHA-1 and SHA shall be prepared respectively as follow:

Table 1

Different message block between SHA and SHA-1

SHA	SHA-1
$w_t = m_t \qquad (0 \le t \le 15)$ $w_t = w_{t-3} \oplus w_{t-8} \oplus w_{t-14} \oplus w_{t-16}$ $(16 \le t \le 79)$	$w_t = m_t \qquad (0 \le t \le 15)$ $w_t = \mathrm{ROTL}^1(w_{t-3} \oplus w_{t-8} \oplus w_{t-14}$ $\oplus w_{t-16}) \qquad (16 \le t \le 79)$

The other reason why ROTL^1 function can upgrade the security level is the increase of involved terms of m_t. For example, when comparing w_{27} in both SHA and SHA-1 (shown as follows), only 6 terms involved in SHA compared with 14 terms involved in SHA-1.

Table 2

w_{27} in SHA and SHA-1

SHA involved 6 terms	$w_{27} = m_2 \oplus m_3 \oplus m_4 \oplus m_7 \oplus m_8 \oplus m_{15}$
SHA-1 involved 14 terms	$w_{27} = m_2^4 \oplus m_3^2 \oplus m_4^4 \oplus (m_5^2 \oplus m_5^3) \oplus m_7^3$ $\oplus m_8^2 \oplus (m_{10}^2 \oplus m_{10}^4) \oplus (m_{11}^1 \oplus m_{11}^2)$ $\oplus (m_{13}^1 \oplus m_{13}^3) \oplus m_{15}^4$

w_{27} becomes independent to m_5 in the end even though m_5 has been involved twice in SHA. But in SHA-1, m_5 is involved under ROTL function thus m_5^2 and m_5^3 will not be eliminated. Below is a figure comparing the number of terms involved in message schedules of both SHA and SHA-1. x-axis presents the index, and y-axis presents the number of terms.

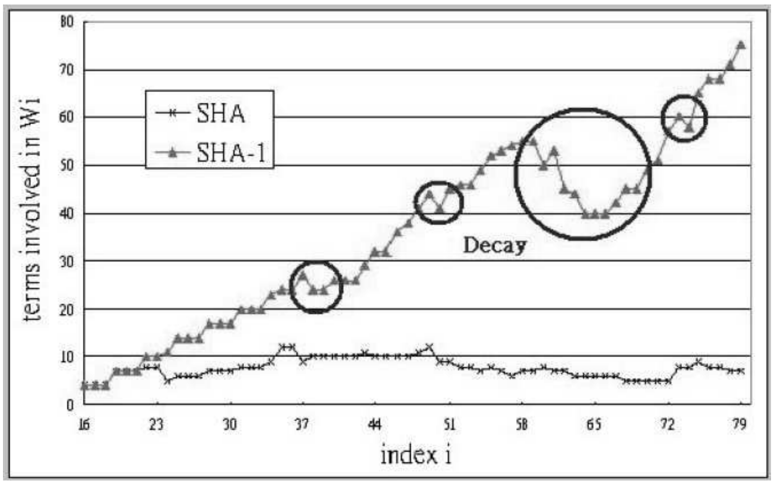

Figure 1

Terms involved between SHA and SHA-1

Not only the paper "Differential collisions in SHA-0" shows the security level of SHA-1 is greatly higher than SHA [2], but also in Figure 1 that shows the terms involved in SHA-1 is much more than in SHA. Furthermore, we also find, however, the decay phenomenon in message schedule, which points out the existence of some inefficient calculations in SHA-1. If the inefficient calculations could be modified such that the decay phenomenon postpone, much more terms will be involved in later w_t.

First modification scheme of SHA-1 (SHA-m1)

Firstly, we re-write the original recursive equation into a general form:

$$w_t = m_t\,, \qquad 0 \le t \le 15$$

$$w_t = \mathrm{ROTL}^1(w_{t-t_1} \oplus w_{t-t_2} \oplus w_{t-t_3} \oplus w_{t-t_4})\,, \qquad 16 \le t \le 79\,.$$

And, we define some notations with convenience and generality. Let $m^{(i)}$ be an input block, $i = 0, \ldots, 15$; and $w_j, j = 0, \ldots, 79$ be the message words.

Table 3

Notations of proposed scheme

ROTL^b	Left rotation of b bits
ROTR^b	Right rotation of b bits
m_i^b	Left rotation of b bits on m_i, $i = 0, \ldots, 15$
w_i^b	Left rotation of b bits on w_i, $i = 0, \ldots, 79$
$m_j^{b_1 \ldots b_{p_j}}$	$m_j^{b_1} \oplus m_j^{b_2} \oplus \ldots m_j^{b_{p_j}}$, $j = 0, \ldots, 15$
$w_j^{b_1 \ldots b_{q_j}}$	$= w_j^{b_1} \oplus w_j^{b_2} \oplus \ldots w_j^{b_{q_j}}$, $j = 0, \ldots, 79$
$\lvert m_j^{b_1 \ldots b_{p_j}} \rvert$	$= \lvert\{b_1, \ldots, b_{pj}\}\rvert = p_1, j = 0, \ldots, 15$

As a result, the original SHA-1 algorithm's (t_1, t_2, t_3, t_4) equals to $(3, 8, 14, 16)$ according to following basic constraints:

(a) $1 \leq t_1 \leq t_2 \leq t_3 \leq t_4 = 16$.

(b) $\gcd(t_1, t_2, t_3) = 1$.

There are $C\binom{15}{3} = 455$ possibilities to assign (t_1, t_2, t_3), where $1 \leq t_1 \leq t_2 \leq t_3 \leq 15$. We list part of experiments in Table 4 and comparison between SHA-m1 and SHA-1 in Figure 2. And according to our experiments, the best choice is $(t_1, t_2, t_3 = \{1, 2, 11\}$.

Table 4

Parts of experiments for choosing $\{t_1, t_2, t_3\}$

t_1	t_2	t_3	Total terms	Maximum number of involved terms in w_t	Average terms involved of all w_t
...					
1	2	10	7279	175	113.4844
1	2	11	8670	212	135.2188
1	2	12	7189	182	112.0781
...					

SHA-m1 algorithm costs as much time as SHA-1. However in Figure 2, the terms involved in SHA-m1 are significantly more than SHA-1 and the decay phenomenon, postpone.

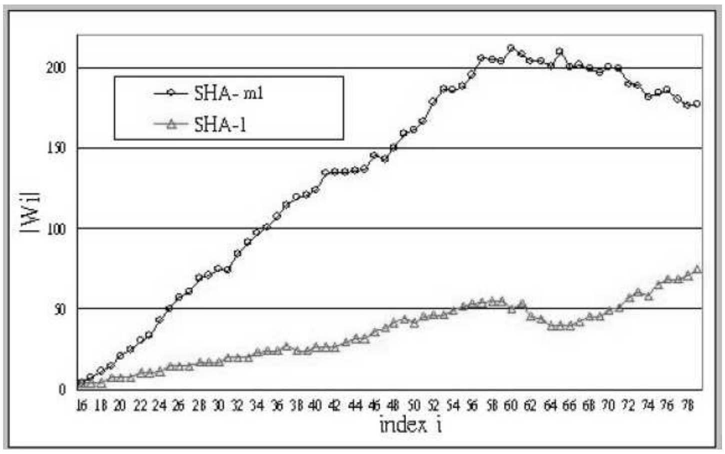

Figure 2

Comparison between SHA-1 and SHA-m1

Second trial of SHA-1

Another viewpoint to modify SHA-1 is based on the ROTL^1 function. We re-write the original equation as following and summarize 3 conclusions:

$$w_t = m_t\,, \qquad 0 \le t \le 15$$

$$w_t = \mathrm{ROTL}^b(w_{t-3} \oplus w_{t-8} \oplus w_{t-14} \oplus w_{t-16})\,, \quad 16 \le t \le 79\,.$$

1. ROTL^b and ROTL^{32-b} cause the same effect;
2. The smaller $\gcd(32, b)$ is, the more involved terms will be; and
3. ROTL^n and ROTL^m will cause the same effect if $\gcd(n, 32) = \gcd(m, 32)$.

We classify four groups as listed in Table 5. The original SHA-1 is one of the 24 experiments with the most terms involved. The same experiments on SHA-m1 are classified into 5 groups by the largest common divisor of 32 and the variable b. As a result, rotating one bit is the best choice already both in SHA-1 and SHA-m1.

Third modification scheme of SHA-1 (SHA-m2)

We re-write the w_t in another form:

$$w_t = m_t\,, \qquad 0 \le t \le 15$$

$$w_t = (w_{t-3})^{b_1} \oplus (w_{t-8})^{b_2} \oplus (w_{t-14})^{b_3} \oplus (w_{t-16})^{b_4}\,, \quad 16 \le t \le 79$$

where $0 \le b_1, b_2, b_3, b_4 \le 31$.

Table 5

Four groups of SHA-1 on $w_t = \mathbf{ROTL}^b(w_{t-3} \oplus w_{t-8} \oplus w_{t-14} \oplus w_{t-16})$

	$\gcd(b, 32)$	variations	Total terms
$b = \{1,2,3,5,6,7,9,10,11,13, 14,15,17,18,19,21,22, 23,25,26,27,29,30,31\}$	$\{1,2\}$	$\{31,15\}$	2271
$b = \{4,12,20,28\}$	4	7	1733
$b = \{8,24\}$	8	3	1265
$b = 16$	16	1	725

Based on the results in second trial, we make one supposition that "The largest number of 'Terms involved in w_t' will appear when b_1, b_2, b_3, and b_4 are all odds". Hence, the time complexity to determine b_1, b_2, b_3, and b_4 is reduced from 32^4 to 16^4. We obtain two results:

1. the maximal number of 'Terms involved in w_t' founded in 1280 experiments is 2509; one of them is $\{b_1, b_2, b_3, b_4\} = \{1,3,9,3\}$.
2. the minimum number of 'Terms involved in w_t' founded in 256 experiments is 1023; one of them is $\{b_1, b_2, b_3, b_4\} = \{1,1,3,7\}$.

We develop SHA-m2 by using one of the best choice $\{b_1, b_2, b_3, b_4\} = \{1,3,9,3\}$ and show the comparison between SHA-1 and SHA-m2 as follows:

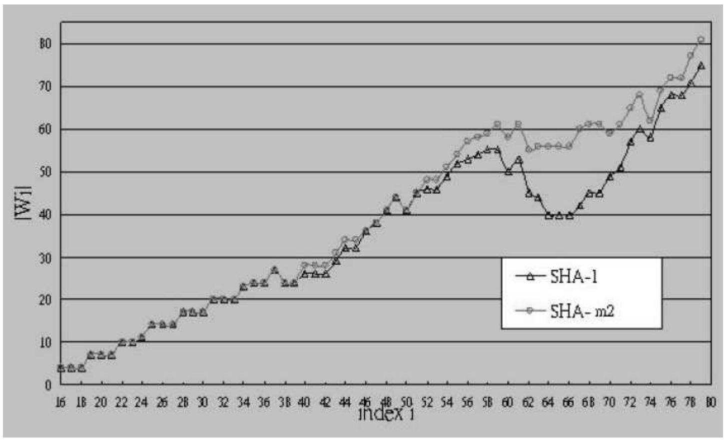

Figure 3

Comparison $|w_i|$ between SHA-1 and SHA-m2

Conclusion

In order to increase the 'Terms involved in w_t', we develop two algorithms as SHA-m1 and SHA-m2 by modifying recursive equations and the number of shift-rotated bit of SHA-1. The more nonlinear terms are involved, the more terms of f_t and $a = \mathrm{ROTL}^5(a) + f_t(b, c, d) + e + K_t + W_t$ [3] will be effective. Because the increase of the nonlinear terms really helps to enhance the security level of original SHA-1, this analysis could also be used in all SHA-serials or other hash functions. Basing on our result, we can further develop the more secure one-way hash function such as SHA-1024 or SHA-2048. In the future, we will try $w_t = (w_{t-t_1})^{b_1} \oplus (w_{t-t_2})^{b_2} \oplus (w_{t-t_3})^{b_3} \oplus (w_{t-16})^{b_4}$ to make the optimal development.

References

[1] E. Biham and R. Chen, Near-collisions of SHA-0, *Advances in Cryptology – Crypto'2004*, LNCS 3152, Springer-Verlag, 2004.

[2] F. Chabaud and A. Joux, *Differential Collisions in SHA-0*, Crypto'98, H. Krawczyk (ed.), LNCS 1462, 1998, pp 56-71.

[3] NIST, *FIPS 180-2, Secure Hash Standard*, US Department of Commerce, Washington D.C., August 2002.

[4] P. van Oorschot and M. Wiener, Parallel collision search with application to hash functions and discrete logarithms, in *Proceedings of 2nd ACM Conference on Computer and Communication Security*, 1994.

[5] X. Wang, D. Feng, X. Lai and H. Yu, *Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD*, Cryptology ePrint Archive.

Received July, 2005