Journal of the Faculty of Environmental Science and Technology,
Okayama University
Vol.10, No.1, pp.1-7, February 2005

Linear and Differential Cryptanalysis of SHA-256

WANG Xiao Dong[1], Hirofumi ISHIKAWA[2]

(Received November 29, 2004)

The one-way hash function plays an important role in digital signatures and message authentication from the viewpoint of security. No effective attacking method has been discovered to the algorithm of hash function standard. In this study, we tried to attack SHA-256 in encryption mode using linear and differential cryptanalysis to solve a private key. We deduced that an estimate of the private key would require huge known and chosen plaintexts in both linear and differential cryptanalysis, and that it would be difficult to decipher SHA-256 in view of the required computation.

Key Words: *differential cryptanalysis, encryption mode, hash function, linear cryptanalysis, SHA-256*

1. INTRODUCTION

The one-way hash function plays an important role in digital signatures and message authentication from the viewpoint of security [1, 2]. Although there have been no reports that the previous hash function standard of 160 output bits SHA-1, which was introduced in 1995 as a minor change of SHA-0 [3], may compromise security, its security will be threatened by improvements in the performance of computers.

In 2002, the National Institute of Standard Technology (NIST), U.S., Department of Commerce issued new hash function standards SHA-256, SHA-384 and SHA-512, having respectively 256, 384 and 512 output bits [4]. The structure of SHA-256, SHA-384, SHA-512 adopted a collision intractable and one-way structure as proposed by Merkle-Damgard [5]. The algorithm of SHA-256, 384, 512 were slightly more complicated than that of SHA-1. Since the algorithms of SHA-384 or 512 seem to be essentially the same as that of SHA-256, we examine only SHA-256 in this paper.

Handschuh *et al.* attacked SHA-1 in encryption mode using both linear and differential cryptanalysis [6]. They deduced that an estimate of the private key would require at least 2^{80} known plaintexts in the linear cryptanalytic attack and at least 2^{116} chosen plaintexts in the differential attack.

Matsui of Mitsubishi Electric Corporation devised linear cryptanalysis [7], a kind of known plaintext attack method that made use of bitwise connection of the exclusive-or between plaintext and ciphertext, and succeed in breaking on 8-round DES cipher. Sakamura *et al.* investigated linear cryptanalysis against AES cipher [8].

Biham *et al.* performed differential cryptanalysis in 1993 [9], a kind of chosen plaintext attack method.

Department of Environmental Synthesis and Analysis, Graduate School of Natural Science and Technology, Okayama University, 700-8530, Japan[1], Department of Environmental and Mathematical Science, Faculty of Environmental Science and Technology, Okayama University, 700-8530, Japan[2].

The differential attack is based on the high probability of certain appearance of difference in plaintext leading to that in ciphertext.

In this study, we tried to attack SHA-256 in encryption mode using linear and differential cryptanalysis to solve a private key, and investigated the robustness of SHA-256 against linear and differential cryptanalysis.

$$ch(x,y,z) = (x \wedge y) \oplus (\sim x \wedge z)$$
$$maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$
$$rotr^n(x) : (x >> n) \vee (x << (32-n))$$
$$\sum_0^{256}(x) = rotr^2(x) \oplus rotr^{13}(x) \oplus rotr^{22}(x)$$
$$\sum_1^{256}(x) = rotr^6(x) \oplus rotr^{11}(x) \oplus rotr^{25}(x)$$

Fig. 1 Functions in SHA-256

Bitwise logical word operations are defined as follows:

$x \wedge y$: bitwise logical "and" of x and y

$x \vee y$: bitwise logical "inclusive-or" of x and y

$x \oplus y$: bitwise logical "exclusive-or" of x and y

$\sim x$: bitwise logical "complement" of x

$x <<$: left shift of x

$x >>$: right shift of x

2. Algorithm of SHA-256

In SHA-256, 256 bits of input are divided into eight words (32 bits), *a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*. The 1-step action in SHA-256 consists of word-wise operations. The functions of each word (32 bits) used by SHA-256 are summarized in Fig. 1. The 1-step processing from *a*-*h* to *a'*-*h'* is shown in Fig. 2 where only *a'* and *e'* among the output variables receive the essential conversion, and the others shift their position. For simplicity, we omit the operation of constant addition here. The whole SHA-256 process is composed of 64 times repetition of the above 1-step procession.

$$T1 = h + \sum_1^{256}(e) + ch(e,f,g)$$
$$T2 = \sum_0^{256}(a) + maj(a,b,c)$$
$$h' = g$$
$$g' = f$$
$$f' = e$$
$$e' = d + T1$$
$$d' = c$$
$$c' = b$$
$$b' = a$$
$$a' = T1 + T2$$

Fig. 2 Algorithm of SHA-256

a-*h* denote input words, which *a'*-*h'* denote output words.

Table 1. *ch* and *maj* functions

x	*y*	*z*	*ch*(*x*,*y*,*z*)	*maj*(*x*,*y*,*z*)
0	0	0	0	0
0	0	1	1	0
0	1	0	0	0
0	1	1	1	1
1	0	0	0	0
1	0	1	0	1
1	1	0	1	1
1	1	1	1	1

0,1: input, output value (bit).

3. Linear Cryptanalysis

We shall treat SHA-256 with encryption mode, namely by inserting a private key in the plaintext as the initial value that was introduced by [6], and try linear cryptanalysis against SHA-256 in this mode by using linear approximation of the input-and-output relation.

Firstly, we can see that for the addition of words $z = x+y$, the successive probability of the i-th bit equation $z_i = x_i \oplus y_i$ is $2^{-(i+1)}$ ($0 \le i \le 31$). The absolute value of the probability subtracted 1/2 is called "bias".

Secondly, we show the input-and-output results of functions *ch*(*x*,*y*,*z*) and *maj*(*x*,*y*,*z*) in Table 1. As the probabilities of equations $y = ch(x,y,z)$, $z = ch(x,y,z)$ hold 3/4 together, their biases are equal to 2^{-2}. For the function *maj*, the successive probabilities of $x = maj(x,y,z)$, $y = maj(x,y,z)$ and $z = maj(x,y,z)$ are all 3/4 (bias= 2^{-2}). From the expression of a' in Fig. 2, we use the following linear approximation:

$$a'_i = h_i \oplus e_{i+6} \oplus e_{i+11} \oplus e_{i+25} \oplus ch(e_i, f_i, g_i) \oplus a_{i+2} \oplus a_{i+13} \oplus a_{i+22} \oplus maj(a_i, b_i, c_i),$$

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	bias
$a_2 \oplus a_{13} \oplus a_{22}$	b_0			$e_6 \oplus e_{11} \oplus e_{25}$	f_0		h_0	2^{-3}
a_0								2^{-1}
	b_0							2^{-1}
		c_0						2^{-1}
			d_0					

Fig. 3. Linear approximation starting from a_0

that leads to a 4-step linear approximation starting from a_0, which is shown in Fig. 3. Its bias is calculated as $2^{2-1}*b_{ch}*b_{maj}=2*2^{-2}*2^{-2}=2^{-3}$ by Piling-up Lemma [7]. In the same way, linear approximation:

$$e'_i = d_i \oplus h_i \oplus e_{i+6} \oplus e_{i+11} \oplus e_{i+25} \oplus ch(e_i, f_i, g_i),$$

leads to a 4-step linear approximation starting from e_0 (Fig. 4), the bias of which is the same as that of its first step.

The definition in Fig. 2 brings into an equation:

$$a_0' \oplus e_0' = (T1_0 + T2_0) \oplus (d_0 + T1_0) = T1_0 \oplus d_0,$$

which leads to a 7-step approximation in Fig. 5. The bias of the last step in Fig. 5 is as same as that of the first step in Fig. 3. The bias of the whole the 7-steps is calculated as follows:

$$bias = 2^{-2} * 2^{-1} * 2^{-1} * 2^{-1} * 2^{-2} * 2^{-1} * 2^{-3} * 2^{6} = 2^{-5}.$$

4. Differential Cryptanalysis

We try to attack SHA-256 in encryption mode, solving the private key by means of differential cryptanalysis.

Table 2. Differences by *ch* and *maj* functions

x	*y*	*z*	*ch*	*maj*
0	0	0	0	0
0	0	1	0/1	0/1
0	1	0	0/1	0/1
0	1	1	1	0/1
1	0	0	0/1	0/1
1	0	1	0/1	0/1
1	1	0	0/1	0/1
1	1	1	0/1	1

"0", "1" or "0/1" denotes that the difference is always 0, always 1 or 0 or 1 with probability 1/2, respectively.

We tabulate the differences of function output (*ch*(*x*,*y*,*z*), *maj*(*x*,*y*,*z*)) depending on the differences of 3-inputs in Table 2. When we alternate any one of variables (*x*, *y*, *z*), the value of *ch*(*x*,*y*,*z*) or *maj*(*x*,*y*,*z*) changes with probability 2^{-1}. The difference of the i+k-th output bit in $\sum_0^{256}$, $\sum_1^{256}$ changes with probability 1 when the i-th input bit changes ($k = 10,19,30$ for $\sum_0^{256}$, $k = 7,21,26$ for $\sum_1^{256}$).

For the addition of words $z = x + y$, when x_i changes, only z_i changes with probability 2^{-1}, and also exactly k-consequentive z_j $(i \le j \le i+k-1)$ changes with probability with $2^{-(k+1)}$ because of carrying bits in addition.

We summarize the probability of differences when only the i-th bit of each word (*a*-*h*) changes in Table 3.

We construct a 2-step difference in Fig. 6; the first step has a probability 1 due to the characteristic of *ch*, while the second step has a small probability due to the action of $\sum_0^{256}(a)$ and $\sum_1^{256}(e)$.

We make one more expression. To avoid the influence of the addition process of $\sum_1^{256}(e_i) = e_{26} \oplus e_{21} \oplus e_7$, we choose an input bit e_i which leads to the highest (31-th) bit as output: input bit e_5, e_{10} or e_{24} according to i+26, i+21 or i+7 respectively. Prolonging the 1 step differential expression starting from e_5 upward, we get a 4-step differential expression starting from b_5 (Fig. 7). Its probability is calculated as

$$2^{-1} * 2^{-1} * 2^{-1} * 2^{-5} = 2^{-8}.$$

Table 3. Differences by i-th bit of each input word

input	output	function	probability
a_i	$a_{i+10}, a_{i+19}, a_{i+30}$	$\sum_0^{256}(a)$	1
	a_i	$maj(a, b, c)$	2^{-1}
	b_i	$b = a$	1
b_i	a_i	$maj(a, b, c)$	2^{-1}
	c_i	$c = b$	1
c_i	a_i	$maj(a, b, c)$	2^{-1}
	d_i	$d = c$	1
d_i	e_i	$e = d + T1$	2^{-1}
e_i	$a_{i+26}, a_{i+21}, a_{i+7}$	$\sum_1^{256}(e)$	1
	a_i	$ch(e, f, g)$	2^{-1}
	$e_{i+26}, e_{i+21}, e_{i+7}$	$\sum_1^{256}(e)$	1
	e_i	$ch(e, f, g)$	2^{-1}
	f_i	$f = e$	1
f_i	a_i	$ch(e, f, g)$	2^{-1}
	e_i	$ch(e, f, g)$	2^{-1}
	g_i	$g = f$	1
g_i	a_i	$ch(e, f, g)$	2^{-1}
	e_i	$ch(e, f, g)$	2^{-1}
	h_i	$h = g$	1
h_i	a_i	$T1 = h + \sum_0^{256}(a) + ch(e, f, g)$	2^{-1}
	e_i	$T1 = h + \sum_0^{256}(a) + ch(e, f, g)$	2^{-1}

a	b	c	d	e	f	g	h	bias
			d_0	$e_6 \oplus e_{11} \oplus e_{25}$	f_0		h_0	2^{-2}
				e_0				2^{-1}
					f_0			2^{-1}
						g_0		2^{-1}
							h_0	

Fig. 4 Linear approximation starting from e_0

a	b	c	d	e	f	g	h	bias
$a_2 \oplus a_{13} \oplus a_{22}$	b_0		d_0					2^{-2}
a_0				e_0				2^{-1}
	b_0				f_0			2^{-1}
		c_0				g_0		2^{-1}
			d_0				h_0	2^{-2}
				e_0	$f_6 \oplus f_{11} \oplus f_{25}$	g_0		2^{-1}
					f_0	$g_6 \oplus g_{11} \oplus g_{25}$	h_0	2^{-3}
a_0	$b_2 \oplus b_{13} \oplus b_{22}$		d_0		$f_6 \oplus f_{11} \oplus f_{25}$		$h_6 \oplus h_{11} \oplus h_{25}$	

Fig. 5 Linear approximation starting from $a_0 \oplus e_0$

a	b	c	d	e	f	g	h	probability
0	0	0	0	0	f_{31}	g_{31}	0	
a_{31}	0	0	0	e_{31}	0	g_{31}	h_{31}	1
$a_{6,9,18,20}$ $_{25,29,31}$	e_{31}	0	0	$e_{6,20,25,31}$	f_{31}	0	h_{31}	2^{-11}

Fig. 6 2-steps difference expression starting from f_{31}, g_{31} on the bases of the characteristic of *ch*

"0" means that the input value does not change, and "x_i", "$x_{i,j}$" means that the input value differs in only *i*-th bit or *i* and *j*-th bit bits of *x*.

a	b	c	d	e	f	g	h	probability
0	b_5	0	0	0	0	0	0	
0	0	c_5	0	0	0	0	0	2^{-1}
0	0	0	d_5	0	0	0	0	2^{-1}
0	0	0	0	e_5	0	0	0	2^{-1}
$a_{12,26,31}$	0	0	0	$e_{12,26,31}$	f_5	0	0	2^{-5}

Fig. 7 4-steps difference expression starting from b_5

5. Discussion

The linear cryptanalysis against 7 steps in SHA-256 requires $2^8(1/4(2^{-5})^{-2})$ known plaintexts to attain the success rate of 84.1% being calculated by Lemma in [7], since its bias is estimated as 2^{-5}. Therefore the bias of the whole 64-steps for SHA-256 is presumed to be less than $2^{-45}((2^{-5})^9)$.

Besides the theoretical investigation, we tried to search for better linear approximate expressions by means of computer experiment. The search project started at the bottom line in one of the best 7 step linear approximations in Fig. 5. The program worked on the following conditions:

1. the 8 word inputs (*a-h*) were generated 100,000 times at random.
2. it examined all combinations of l bits in the output side where l ran over 1-15 while the input side remained stable as
$$a_0 \oplus b_2 \oplus b_{13} \oplus b_{22} \oplus d_0 \oplus f_6 \oplus f_{11} \oplus f_{25} \oplus h_6 \oplus h_{11} \oplus h_{25}$$
3. a new linear approximation was recorded when its bias exceeded 2^{-10}.

However, no new linear approximations were found by this computer experiment. When we prolong the linear approximation in Fig. 5 downward, it is difficult to compose a 1-step linear approximation with input bits (6, 11, 25) of h having a bias at more than 2^{-14}.

The differential cryptanalysis against 4 steps in SHA-256 needs 2^8 chosen plaintexts to solve 1-bit (b_5, b_{10} or b_{24}), since its probability is estimated as 2^{-8} (Fig. 7). Therefore, the probability of the whole 64-steps for SHA-256 is presumed to be less than $2^{-128}((2^{-8})^{16})$.

We also tried to search for better differential expressions by computer experiment. The search project started at the bottom line expression in Fig. 7. The program worked on almost the same conditions as in the linear approximation:

1. the 8 word inputs (*a-h*) were generated 100,000 times at random.
2. it examined all combinations of l bits in the output side where l ran over 1-15 while the input side remained stable as
$$a_{12} \oplus a_{26} \oplus a_{31} \oplus e_{12} \oplus e_{26} \oplus e_{31} \oplus f_5 \cdot$$
3. a new differential expression was recorded when its probability exceeded 2^{-10}.

However, this yielded no effective differential expression because it was difficult to compose a 1-step additional expression with input bits ($a_{12,26,31}$ and $e_{12,26,31}$) having a probability of more than 2^{-13} prolonging the differential expression in Fig. 7 downward.

The bit expansion functions $\sum_0^{256}$, $\sum_1^{256}$ have a great influence on decipherment due to the decrease in bias or probability through addition process for linear or differential cryptanalysis.

Consequently, we estimate that linear and differential cryptanalysis against the whole 64 steps in SHA-256 requires more than $2^{88}(1/4(2^{-45})^{-2})$ known plaintexts to attain a success rate of 84.1%, and differential cryptanalysis requires 2^{128} chosen plaintexts, indicating the difficulty of decipherment against SHA-256 encryption in view of the required computation. Further careful research is necessary because there remains much to learn about security following the proposal of this new hash function standard SHA-256, 384, 512.

References

[1] National Institute of Standards and Technology (NIST). Digital Signature Standard (FIPS 186-2), 2001.

[2] American National Standards Institute (ANSI). The Elliptic Curve Digital Signature Algorithm (ANSI X9.62), 1998.

[3] National Institute of Standards and Technology (NIST). Secure Hash Signature Standard (FIPS 180-1), 1995.

[4] National Institute of Standards and Technology (NIST). Secure Hash Signature Standard (FIPS 180-2), 2002.

[5] Rivest RL, The MD4 message digest algorithm. In: Advances in Cryptology CRYPTO'90, LNCS **537**, pp. 303-311. Springer-Verlag, Berlin, 1991.

[6] Handschuh H, Knudsen LR, and Robshaw MJ. Analysis of SHA-1 in encryption mode, In: CT-RSA, Naccache, D. (Ed.), LNCS 2020, Springer-Verlag, Berlin, pp. 70-83, 2001.

[7] Matsui M, Linear cryptanalysis method for DES cipher. In: Advances in Cryptology, EUROCRYPT'93, Helleseth T. (Ed.), LNCS **765**, pp. 386-397, Springer-Verlag, Berlin, 1994.

[8] Sakamura K, Wang XD, Ishikawa H, A study on linear cryptanalysis of AES cipher, *J. Fac Environ Sci & Tech, Okayama Univ.*, **9**, 19-26, 2004.

[9] Biham E, Shamir A. Differential cryptanalysis of the data encryption standards, Springer-Verlag, Berlin, 1993.