A New 256-bit Hash Function DHA-256 : Enhancing the Security of SHA-256

Jesang Lee, Donghoon Chang, Hyun Kim,
Eunjin Lee, Deukjo Hong, Jaechul Sung,
Seokhie Hong, Sangjin Lee

Korea Univ, University of Seoul

Contents

- Motivation of Design of DHA-256
- DHA-256 Algorithm
- Design Principle of DHA-256
- Security Analysis of DHA-256
- Conclusion

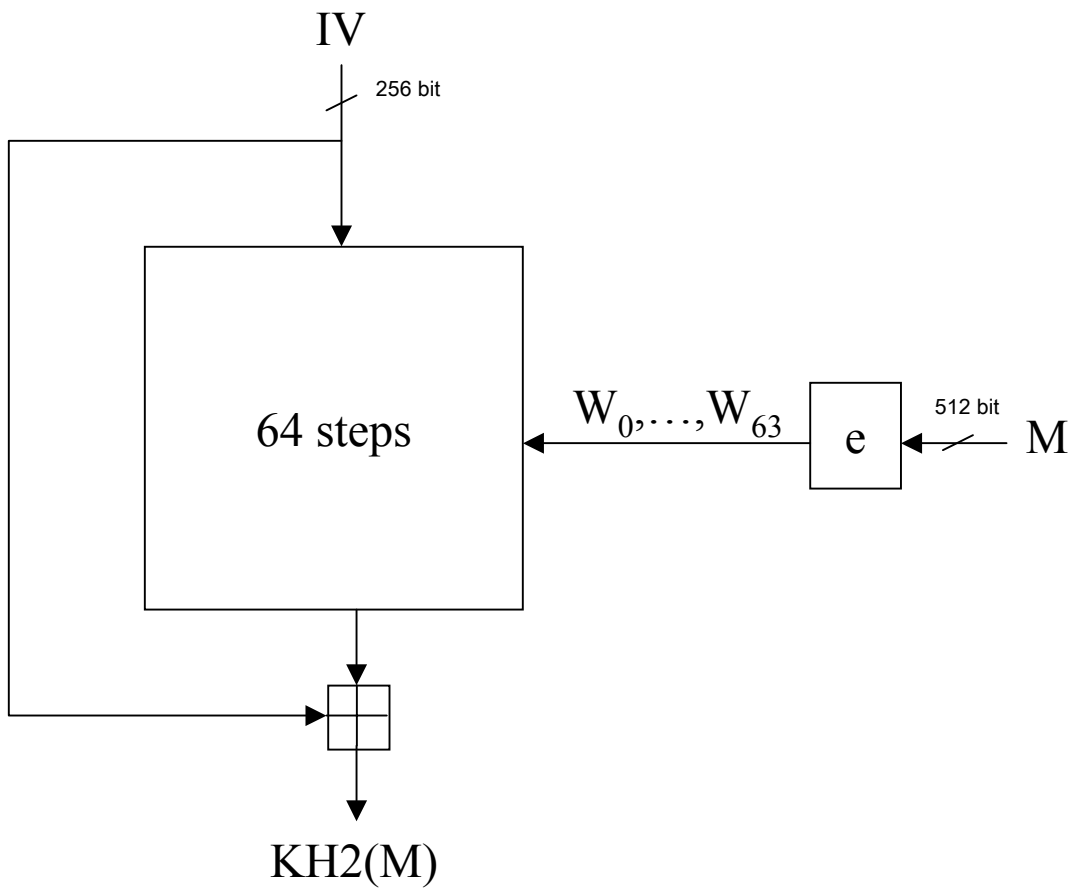
Motivation

- Recently, SHA-0/1 were fully broken by Wang et al..
- In case of SHA-256, the security bound is not so good.
  - ✓ The probability of the inner collision pattern is 2^{-39}.
  - ✓ The repetition number of the pattern is not big.
- In this talk, we suggest DHA-256 to enhance the security of SHA-256.
- The step function and the message expansion of DHA-256 have almost same resources as SHA-256 but provide higher security bound.

Outline of DHA-256 1/2

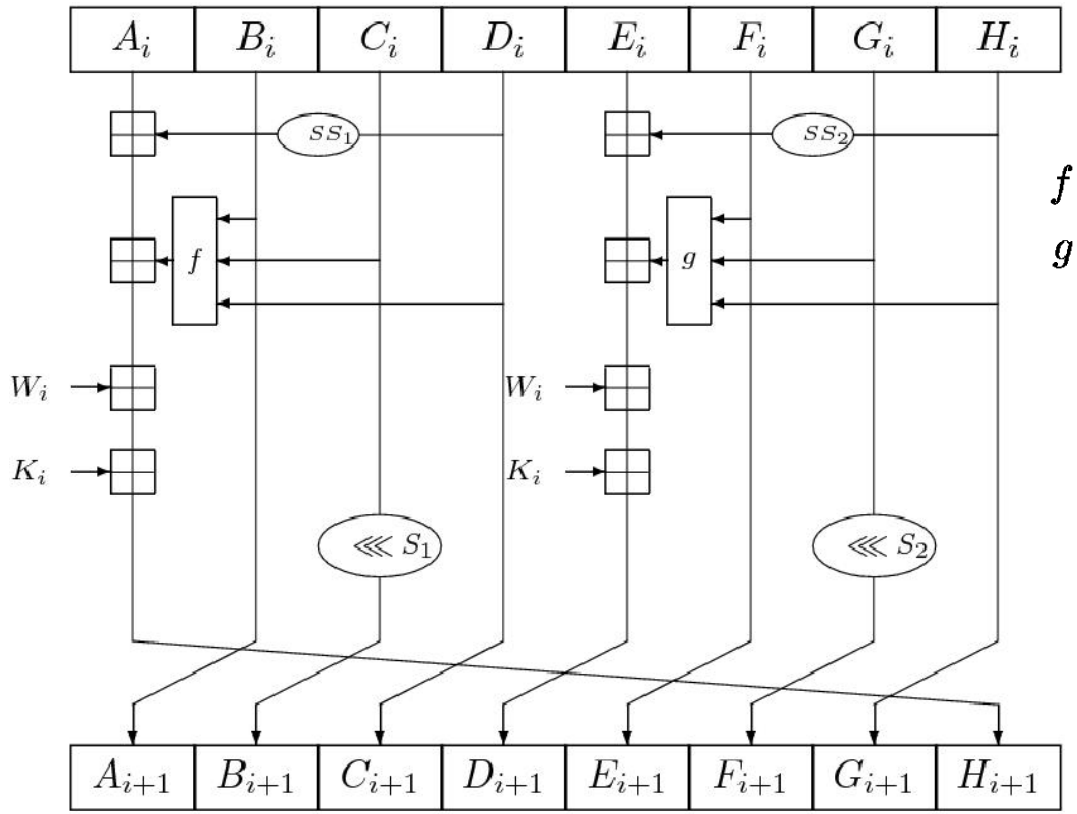
- Message Block Size : 512 bit (16 words)
- Output Size : 256 bit (8 words)
- Consists of 64 Steps

Outline of DHA-256 2/2



DHA-256 1/2

- Step Function & Boolean Functions & Shift Rotations



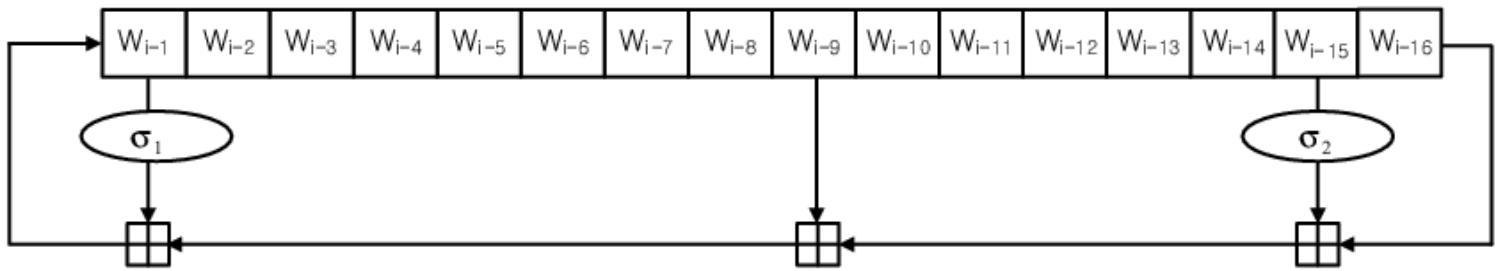
$$f(x,y,z) = (x \wedge y) \vee (\neg x \wedge z)$$
$$g(x,y,z) = (x \wedge y) \oplus (y \wedge z) \oplus (z \wedge x)$$
$$SS_1(x) = x \oplus x^{\lll 11} \oplus x^{\lll 25}$$
$$SS_2(x) = x \oplus x^{\lll 19} \oplus x^{\lll 29}$$
$$S_1(x) = x^{\lll 17}, S_2(x) = x^{\lll 2}$$

DHA-256 2/2

- Message Expansion

$$\sigma_1(x) = x \oplus x^{\lll 7} \oplus x^{\lll 22}$$
$$\sigma_2(x) = x \oplus x^{\lll 13} \oplus x^{\lll 27}$$

$$W_i = \sigma_1(W_{i-1}) + W_{i-9} + \sigma_2(W_{i-15}) + W_{i-16} \; (16 \le i \le 63)$$



Design Principle 1/5

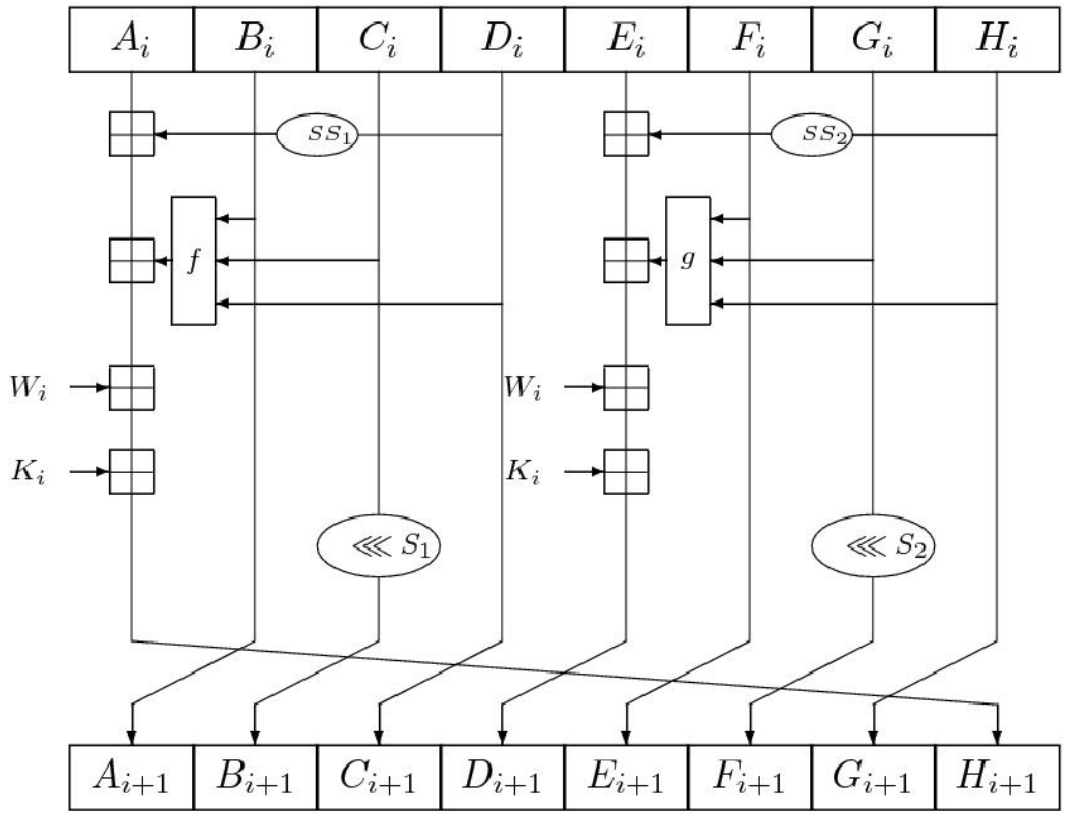
- Shift Rotations

$$SS_1(x) = x \oplus x^{\lll 11} \oplus x^{\lll 25}$$
$$SS_2(x) = x \oplus x^{\lll 19} \oplus x^{\lll 29}$$
$$S_1(x) = x^{\lll 17}, S_2(x) = x^{\lll 2}$$

To define SS_1, SS_2, S_1 and S_2, we search the case which maximizes the diffusion effect after 8 steps when 1 bit difference is injected and satisfies following conditions.

- Intervals of values in SS_1, $SS_2 \geq 4$

- No divisor of 32 in SS_1, SS_2



Design Principle 2/5

- Message Expansion (selection of a, b, c, d) 1/2

$$W_i = \sigma_1(W_{i-a}) + W_{i-b} + \sigma_2(W_{i-c}) + W_{i-d}$$

  - ✓ At first, we choose 1, 15 and 16 because of following reasons :
    - **a=1 : The message word at step (i-1) is used to update the message word at step i. This makes diffusion effect high.**
    - **d=16 : If there is no this condition, the first message word does notinfluence any of the expanded message words.**
    - **c=15 : This makes diffusion effect high in the case of the inverse operation.**

Design Principle 3/5

- Message Expansion (selection of b, σ_1, σ_2) 2/2

$$W_i = \sigma_1(W_{i-a}) + W_{i-b} + \sigma_2(W_{i-c}) + W_{i-d}$$

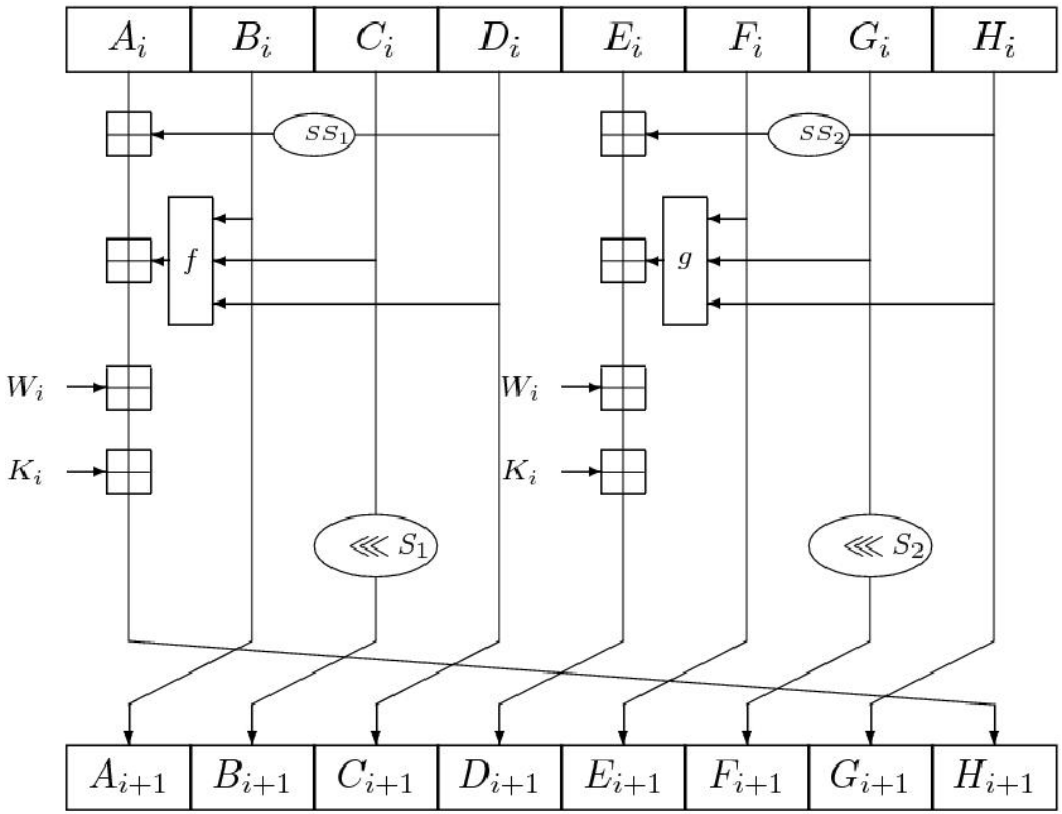
$$\sigma_1(x) = x \oplus x^{\lll 7} \oplus x^{\lll 22}$$

$$\sigma_2(x) = x \oplus x^{\lll 13} \oplus x^{\lll 27}$$

  - ✓ Second, we search b=9 and the shift rotations of σ_1, σ_2 :
    - Recent attacks on SHA-0/1 are attacks with iterating an inner collision pattern. The complexity is related to the number of repetition of the pattern.
    - We search b and the shift rotations of σ_1, σ_2 so that the number of repetition of the pattern is small in step 24 ~ 55.
    - According to Wang et al.'s, it may be possible to find a collision satisfying conditions in step 0 ~ 23 and last 8 steps by using near collisions.

Design Principle 4/5

- Updating chaining variable

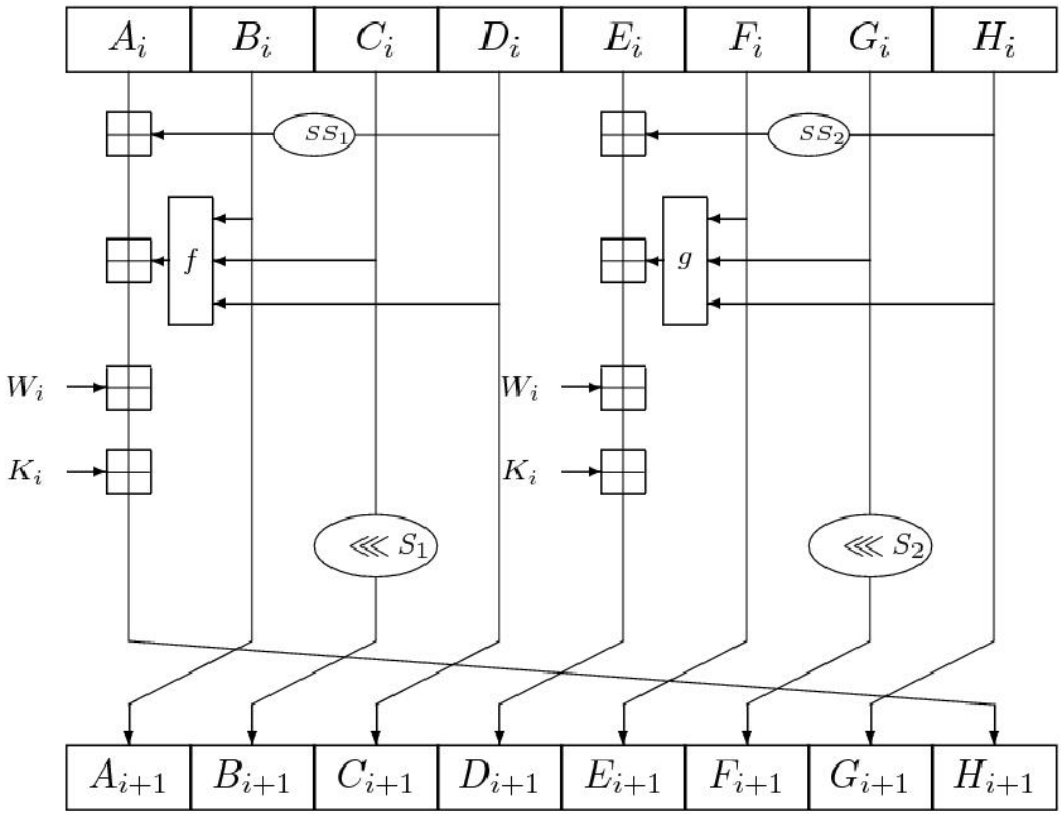


Each message is used twice in a step : this makes it difficult to construct valid differential characteristics.

The updated words are also used in next step : this makes the diffusion effect high.

Design Principle 5/5

- Boolean functions $f(x,y,z) = (x \wedge y) \vee (\neg x \wedge z)$
  $g(x,y,z) = (x \wedge y) \oplus (y \wedge z) \oplus (z \wedge x)$



We choose boolean functions satisfying SAC : this makes it easy to quantify the security bound against recent collision attacks using inner collision pattern.

Security Analysis against Wang et al.'s attack 1/9

- Attack results on SHA-0/1 were introduced at Crypto'05
  - ✓ SHA-0 : Complexity 2^{33}
  - ✓ SHA-1 : Complexity 2^{63}
- Analysis Principle
  - ✓ Find the best inner collision pattern.
  - ✓ Minimize the frequency of the pattern in step 21~58 influenced by the message expansion algorithm.

Security Analysis against Wang et al.'s attack 2/9

- We found 9 step inner collision pattern : Prob. is 2^{-64}
  - ✓ This pattern is the best pattern with the respect to the probability.
  - ✓ When message differences satisfy following condition, this message differences become an inner collision pattern.

$$W_{i+2} = SS_2(SS_1(W_i))$$
$$W_{i+5} = SS_1(W_i^{\lll 17}) + SS_2(W_i^{\lll 2})$$
$$W_{i+8} = W_i^{\lll 19}$$

Security Analysis against Wang et al.'s attack 3/9

- The best 9-step inner collision pattern 1/4
  - ✓ Differences of Message Words

$$
\begin{aligned}
W_i &= 10000000000000000000000000000000 \\
W_{i+2} &= 10110001001001000000110010000000 \\
W_{i+5} &= 00000100000000010011000000001010 \\
W_{i+8} &= 00000000000001000000000000000000
\end{aligned}
$$

round	A	B	C	D	E	F	G	H
i	0	0	0	80000000	0	0	0	80000000
$i+1$	0	0	80000000	90040000	0	0	80000000	81000400
$i+2$	0	00010000	90040000	0	0	00000002	81000400	0
$i+3$	00010000	00012008	0	0	00000002	04001002	0	0
$i+4$	00012008	0	0	00000002	04001002	0	0	00010000
$i+5$	0	0	00000002	0	0	0	00010000	0
$i+6$	0	00040000	0	0	0	00040000	0	0
$i+7$	00040000	0	0	0	00040000	0	0	0
$i+8$	0	0	0	0	0	0	0	0

Security Analysis against Wang et al.'s attack 4/9

- The best 9-step inner collision pattern 2/4
  - ✓ Input words of Boolean functions
    - f : B,C,D ➡ The number of bit positions having 1 bit difference is 14.
    - g : F,G,H ➡ The number of bit positions having 1 bit difference is 14.
    - Total = 28 bit positions. Therefore the probability that boolean functions output specific differences is 2^{-28}.

round	A	B	C	D	E	F	G	H
i	0	0	0	80000000	0	0	0	80000000
$i+1$	0	0	80000000	90040000	0	0	80000000	81000400
$i+2$	0	00010000	90040000	0	0	00000002	81000400	0
$i+3$	00010000	00012008	0	0	00000002	04001002	0	0
$i+4$	00012008	0	0	00000002	04001002	0	0	00010000
$i+5$	0	0	00000002	0	0	0	00010000	0
$i+6$	0	00040000	0	0	0	00040000	0	0
$i+7$	00040000	0	0	0	00040000	0	0	0
$i+8$	0	0	0	0	0	0	0	0

Security Analysis against Wang et al.'s attack 5/9

- The best 9-step inner collision pattern 3/4
  - ✓ addition parts
    - A, E, the output of boolean functions : there are 36 bit positions having non-zero difference.
    - Therefore the probability that specific differences hold is 2^{-36}.

round	A	B	C	D	E	F	G	H
i	0	0	0	80000000	0	0	0	80000000
$i+1$	0	0	80000000	90040000	0	0	80000000	81000400
$i+2$	0	00010000	90040000	0	0	00000002	81000400	0
$i+3$	00010000	00012008	0	0	00000002	04001002	0	0
$i+4$	00012008	0	0	00000002	04001002	0	0	00010000
$i+5$	0	0	00000002	0	0	0	00010000	0
$i+6$	0	00040000	0	0	0	00040000	0	0
$i+7$	00040000	0	0	0	00040000	0	0	0
$i+8$	0	0	0	0	0	0	0	0

Security Analysis against Wang et al.'s attack 6/9

- The best 9-step inner collision pattern 4/4
  - ✓ Total probability
    - Boolean parts : 28 bit positions
    - Addition parts : 36 bit positions
    - Total is 64 bit position. Therefore the probability of inner collision pattern is 2^{-64}.

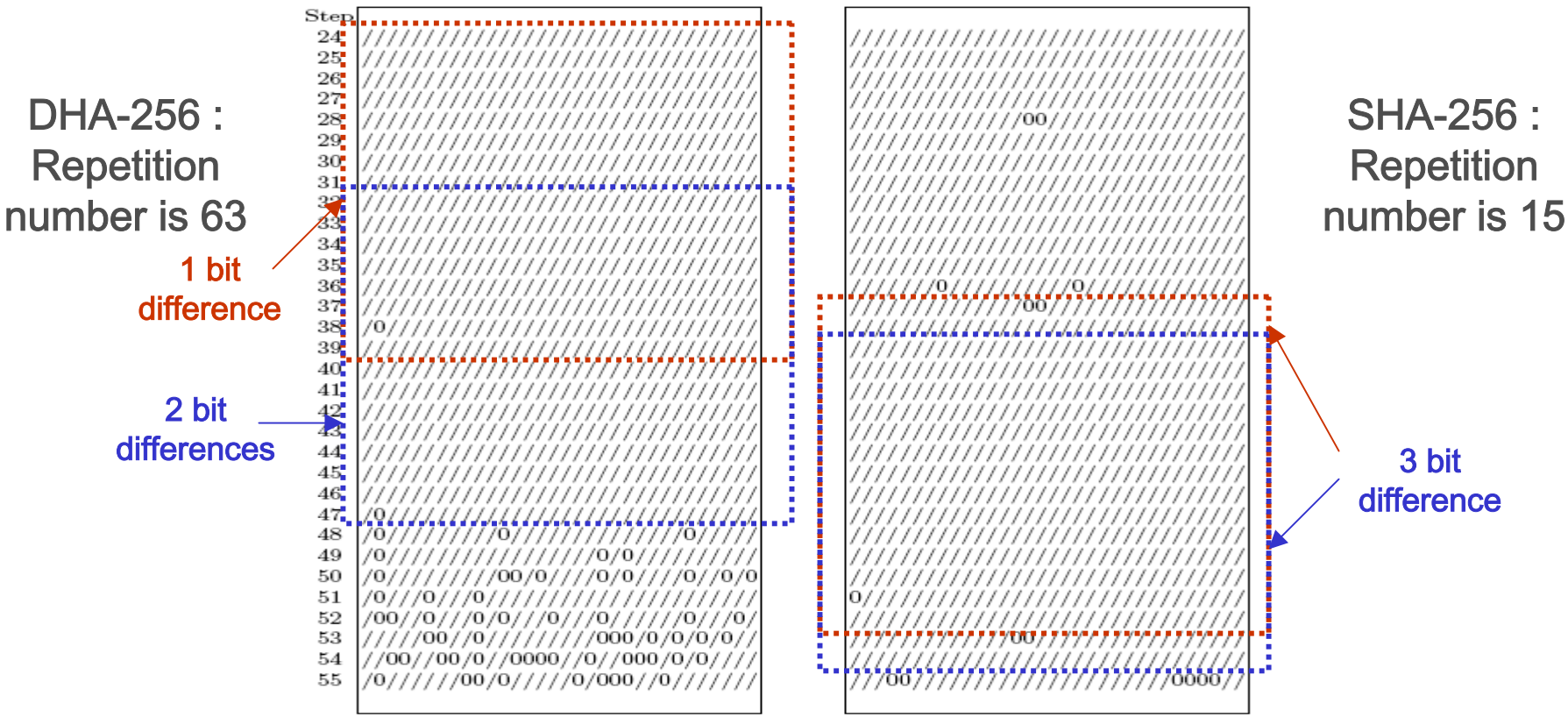
round	A	B	C	D	E	F	G	H
i	0	0	0	80000000	0	0	0	80000000
$i+1$	0	0	80000000	90040000	0	0	80000000	81000400
$i+2$	0	00010000	90040000	0	0	00000002	81000400	0
$i+3$	00010000	00012008	0	0	00000002	04001002	0	0
$i+4$	00012008	0	0	00000002	04001002	0	0	00010000
$i+5$	0	0	00000002	0	0	0	00010000	0
$i+6$	0	00040000	0	0	0	00040000	0	0
$i+7$	00040000	0	0	0	00040000	0	0	0
$i+8$	0	0	0	0	0	0	0	0

Security Analysis against Wang et al.'s attack 7/9

- Repetition of the Inner collision pattern 1/2
  - ✓ We check how many the pattern occurs in step 24~55.
  - ✓ We consider all cases that 16 consecutive words have one, two, three bit differences.
  - ✓ In this case, there exists 16 consecutive words such that the minimum repetition number of the pattern in step 24~55 is 63.
    - In case of SHA-256, the minimum repetition number of the pattern in step 24~55 is 15.

Security Analysis against Wang et al.'s attack 8/9

- Repetition of Inner collision pattern 2/2



Security Analysis against Wang et al.'s attack 9/9

- Comparison between DHA-256 and SHA-256

	DHA-256	SHA-256
Inner Collision Pattern	2^{-64}	2^{-39}
Frequency of the pattern (24~55)	63	15
The upper bound of attack success	2^{-4032}	2^{-585}

Conclusion

- DHA-256 uses each message word twice in a step.
- Using this idea, we constructed a step function such that the probability of the inner collision pattern is low.
- We also improved the message expansion of SHA-256 such that the repetition number of the pattern of DHA-256 is higher than that of SHA-256.

A New 256-bit Hash Function DHA-256 : Enhancing the Security of SHA-256

Jesang Lee[1], Donghoon Chang[1], Hyun Kim[1], Eunjin Lee[1], Deukjo Hong[1], Jaechul Sung[2], Seokhie Hong[1], Sangjin Lee[1]

[1] Center for Information and Security Technologies
Korea University, Seoul, Korea
{dogcraft, dhchang, bkcloud18, walgadak, hongdj, hsh, sangjin}@cist.korea.ac.kr
[2] Department of Mathematics, University of Seoul, Korea
jcsung@uos.ac.kr

Abstract. DHA(Double Hash Algorithm)-256 is a dedicated hash function with message length of 512 bits and output length of 256 bits. "Double" means that each message word which is generated by the message expansion algorithm is used twice in a step. Our Design goal is to enhance the security of SHA-256. The step function and the message expansion of DHA-256 has almost same resource as SHA-256 but provides higher security bound against recent proposed attacks.

1 Introduction

Since introduction of MD4 [16], the design strategy of MD4 has been most popular for designing dedicated hash functions. MD5 [15], RIPEMD [13], RIPEMD-128,160 [7], HAVAL [26] and SHA-0,1,224,256,384,512 [11, 12] are well-known hash functions which follow such design strategy. All of them except MD4 [6], HAVAL [17] and SHA-0 [5, 18, 19] had not been totally broken until Wang *et al.* proposed a new collision-finding method for hash function [21, 23], while other existing attacks depend largely on the shift rotation and the input order of message words.

In 2004 and 2005, Wang *et al.* announced collision attacks on MD4, MD5, RIPEMD, HAVAL, SHA-0 and SHA-1 [20, 21, 23–25]. Comparing with existing attack methods, their attacks are noticeably improved because they break down the wall of the shift rotation values and the input order of message words which make it difficult to analyze hash functions. They also demonstrate the superiority of their attack method by showing improved results on MD4 and HAVAL and SHA-0 which were already analyzed.

Hash functions analyzed by Wang *et al.* are classified into 2 groups according to message word input methods as Table 1. Hash functions in Group-I use the "Message Word Re-Ordering" method. In the case of Group-II, input message words are determined by a message word expansion function. Wang *et al.*'s analyzing methods for each Group are as follows:

Group-I(Re-ordering)	Group-II(Expansion)
MD4, MD5, RIPEMD 3-pass HAVAL	SHA-0, SHA-1

Table 1. Classification according to the Message Word Input Methods

Group-I : Each message word is applied one time to each round. Therefore if we give a difference to one message word, message words having as many differences as the number of rounds in a compression function are applied to the entire compression function. In the case of hash functions is composed of three rounds, the attack scenario is as follows. First, find a differential characteristic (whose input and output chaining variables' differences are zeros) of the third round. Then construct a differential characteristic of first two rounds (round 1, round 2) with the difference of message words used at the third round characteristic. It is possible to construct effectively and easily a characteristic with properties of boolean functions and the relation between the addition difference and the XOR difference. Using this 3-round-characteristic, hash functions (except for MD5) in Group-I are all analyzed.

In the case of MD5, which is composed of four rounds, a structural weakness of its step function is used to construct last two round characteristic. If all input chaining variables have differences only at the most significant bit position, these differences are preserved with high probability [4]. This structural property is also used to find a pseudo-collision which has the non-zero difference of initial values. The pseudo-collision related works were studied by Gilbert and Handschuh [8]. Wang *et al.* chose the difference of message words which doesn't destroy the structural property in last two rounds. Then they constructed first two round differential characteristic which makes message pair collide in second round with using message differences used in last two rounds. Finally they could find two block collision (using MD5 compression function two times) with the entire four round differential characteristic.

Wang *et al.*'s analyzing results on hash fucntions in Group-I are summarized as follows :

- Three round hash function such as hash functions in Group-I may not be secure against Wang *et al.*'s attack.
- Four round hash function such as MD5 must have no structural weakness of the step function in order to be secure against the type of Wang *et al.*'s attacks.
- There is no general attack method to analyze hash function having rounds more than three rounds.
- There is no measure to quantify the security of hash functions such as hash functions in Group-I.

Group-II : In the case of SHA-0 and SHA-1, one message word having non-zero difference influences expanded message words so that tens of message words have also non-zero differences. This is because message word at i-th step is made from message words applied at (i-16)-,...,(i-1)-th step. We call this message expansion "LFSR-based Message Expansion". As all expanded message words are

Algorithm	Round	Analyzing Strategy
MD4	3	collision on both the second and the third round
MD5	4	collision on the second round, structural weakness on last two rounds
RIPEMD	3	collision on both the second and the third round
3-pass HAVAL	3	collision on both the second and the third round

Table 2. Analysis of Strategy on Hash Functions in Group-I

generated regularly by LFSR-based message expansion, the property observed in small steps can be expanded over full steps.

In 1997 and 1998, Wang [18, 19] and Chabaud & Joux [5] introduced attack methods to analyze hash functions with LFSR-based Message Expansion. They use an inner collision pattern of 6 consecutive steps to analyze full 80 steps of SHA-0. The attack complexity of their methods depends on the frequency of the repetition of the inner collision pattern. The success probability of the inner collision pattern depends only on the step operation. The frequency of the repetition depends only on the message expansion. In 2004, Biham and Chen[1] improved the complexity of the existing results by introducing the notion "Neutral Bit". They also suggested the method to find collisions more than 2 blocks by using "Near Collision" [2]. In 2005, Biham *et al.* [3] presented real 4 block collision pair by using some near collisions. However it is difficult to analyze SHA-1 only by using these attack methods because differences on different bit positions can not offset only by the existing attack.

Very recently, by improving existing methods, Wang *et al.* lowered the complexity of finding collision of SHA-0 [24] and firstly showed that SHA-1 is not collision resistant [25]. They ignored 3 conditions (of disturbance vectors) needed in existing methods and found disturbance vectors with low hamming weight in round $2 \sim 4$. Characteristics of hash fucntions in Group-II are summarized as follows :

- It is possible to quantify the security of hash function against Wang *et al.*'s attack.
- The possibility of quantifying the security of hash function make it possible to design new hash function with high security.

Our Contribution : SHA-256 [12] also belongs to Group-II. Therefore it is possible to assess the security of SHA-256 against Wang *et al.*'s attack. As SHA-family were designed before Wang *et al.*'s attack is introduced, it is necessary to check whether SHA-family are well-designed and there is room to change the options of SHA-family in order to improve the security of SHA-family. In this paper, we show that SHA-256 is not optimized against Wang *et al.*'s attack. We also suggest a new hash function which is more secure than SHA-256 against Wang *et al.*' attack.

We consider the security of DHA-256 against Wang *et al.*'s attack as follows : When we compute the hamming weight of disturbance corresponding to step $24 \sim 55$, we consider all cases of one, two and three bit differences of the disturbance vector corresponding to 16 consecutive steps among 32 steps from step $24 \sim$

55. Here, we consider only step 24∼55 to measure precisely the security lower bound of DHA-256 because that conditions at step 17-23 may be corrected by modification method like SHA-0 [24] and SHA-1 [25].

Conditions at step 56 ∼ 63 can be ignored by the analyzing method which uses a near collision. In this respect, we can present a lower bound of the attack complexity of finding collision pair with using Wang *et al.*'s attack method. In

	SHA-256	DHA-256
Probability of Inner Collision Pattern	2^{-39} [9]	2^{-64}(this paper)
Hamming Weight of Disturbance(at step 24-55)	15(this paper)	63(this paper)
Total Probability	2^{-585}	2^{-4032}

Table 3. Security Comparison between SHA-256 and DHA-256

Table 3, SHA-256 does not give security as much as we believe. In the case of SHA-256, Hawkes *et al.* showed that the probability of the best inner collision pattern is 2^{-39} [9]. The result of this paper shows that there exists a disturbance vector whose hamming weight corresponding to step 24 ∼ 55 is 15. On the other hand, in the case of DHA-256, the probability of the best inner collision pattern is 2^{-64} in section 4.1. There exists a disturbance vector whose hamming weight corresponding to step 24 ∼ 55 is 70. These results means that it is more difficult to find the collision of DHA-256 with using Wang *et al.*' attack method when comparing with SHA-256. These results also show that components of SHA-256 have to be changed to improve the security of SHA-256 against recent Wang *et al.*'s attack.

The organization of this paper is as follows. In section 2, we describe DHA-256 algorithm. Then, in section 3 and 4, we show the design principle and the security analysis of DHA-256. In section 5, we compare between DHA-256 and SHA-256 with the number of operation used in each algorithm. Finally, we conclude.

2 The Description of DHA-256 Algorithm

In this section, we briefly describe the hash function DHA-256 and we introduce notations used in this paper.

$X^{\lll s}$	left rotation X by s bits
X∧Y	bitwise logical AND operation of X and Y
X∨Y	bitwise OR operation of X and Y
X⊕Y	bitwise XOR operation of X and Y

Table 4. Basic Notions of DHA-256

2.1 Input Block Length and Padding

An input message is processed by 512-bit block. This hash function pads a message by appending a single bit 1 next to the least significant bit of the

message, followed by zero or more bit 0's until the length of the message is 448 modulo 512, and finally appends message length modulo 2^{64}.

2.2 Initial values

Initial values are the same as those of SHA-256.

A_0=0x6a09e667, B_0=0xbb67ae85, C_0=0x3cbef37e, D_0=0xa54ff53a
E_0=0x510e537f, F_0=0x9b05688c, G_0=0x1f83d9ab, H_0=0x5b30cd19

2.3 Boolean Functions

The boolean functions used at each round are as follows.

$$f(x, y, z) = (x \wedge y) \vee (\neg x \wedge z)$$
$$g(x, y, z) = (x \wedge y) \oplus (y \wedge z) \oplus (z \wedge x)$$

2.4 Values of Shift Rotation

$SS_1(x)$, $SS_2(x)$, $S_1(x)$, $S_2(x)$, $\sigma_1(x)$ and $\sigma_2(x)$ are as follows.

$$SS_1(x) = x \oplus x^{\lll 11} \oplus x^{\lll 25}, \quad SS_2(x) = x \oplus x^{\lll 19} \oplus x^{\lll 29},$$
$$\sigma_1(x) = x \oplus x^{\lll 7} \oplus x^{\lll 22}, \quad \sigma_2(x) = x \oplus x^{\lll 13} \oplus x^{\lll 27},$$
$$S_1(x) = x^{\lll 17}, S_2(x) = x^{\lll 2}.$$

2.5 Constants

We use the constants which are the same as those of SHA-256 in Table 5. K_i is the constant which is used in step i. Constants represent the first thirty-two bits of the fractional parts of the cube roots of the first sixty four prime numbers.

2.6 Message Expansion Algorithm

For a 512-bit message block M, we separate M into 16 words, $W_0||W_1||\cdots||W_{15}$. From these 16 words we obtain W_i as follows.

$$W_i = \sigma_1(W_{i-1}) + W_{i-9} + \sigma_2(W_{i-15}) + W_{i-16}, \ (16 \leq i \leq 63)$$

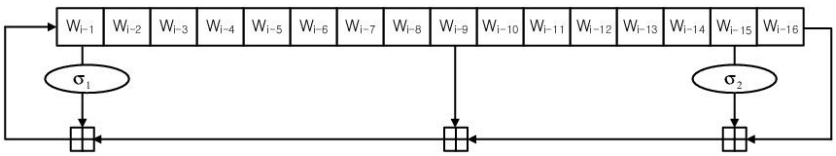


Fig. 1. Message Expansion of DHA-256

K_0	0x428a2f98	K_1	0x71374491	K_2	0xb5c0fbcf	K_3	0xe9b5dba5
K_4	0x3956c25b	K_5	0x59f111f1	K_6	0x923f82a4	K_7	0xab1c5ed5
K_8	0xd807aa98	K_9	0x12835b01	K_{10}	0x243185be	K_{11}	0x550c7dc3
K_{12}	0x72be5d74	K_{13}	0x80deb1fe	K_{14}	0x9bdc06a7	K_{15}	0xc19bf174
K_{16}	0xe49b69c1	K_{17}	0xefbe4786	K_{18}	0x0fc19dc6	K_{19}	0x240ca1cc
K_{20}	0x2de92c6f	K_{21}	0x4a7484aa	K_{22}	0x5cb0a9dc	K_{23}	0x76f988da
K_{24}	0x983e5152	K_{25}	0xa831c66d	K_{26}	0xb00327c8	K_{27}	0xbf597fc7
K_{28}	0xc6e00bf3	K_{29}	0xd5a79147	K_{30}	0x06ca6351	K_{31}	0x14292967
K_{32}	0x27b70a85	K_{33}	0x2e1b2138	K_{34}	0x4d2c6dfc	K_{35}	0x53380d13
K_{36}	0x650a7354	K_{37}	0x766a0abb	K_{38}	0x81c2c92e	K_{39}	0x92722c85
K_{40}	0xa2bfe8a1	K_{41}	0xa81a664b	K_{42}	0xc24b8b70	K_{43}	0xc76c51a3
K_{44}	0xd192e819	K_{45}	0xd6990624	K_{46}	0xf40e3585	K_{47}	0x106aa070
K_{48}	0x19a4c116	K_{49}	0x1e376c08	K_{50}	0x2748774c	K_{51}	0x34b0bcb5
K_{52}	0x391c0cb3	K_{53}	0x4ed8aa4a	K_{54}	0x5b9cca4f	K_{55}	0x682e6ff3
K_{56}	0x748f82ee	K_{57}	0x78a5636f	K_{58}	0x84c87814	K_{59}	0x8cc70208
K_{60}	0x90befffa	K_{61}	0xa4506ceb	K_{62}	0xbef9a3f7	K_{63}	0xc67178f2

Table 5. DHA-256 Constants

2.7 Step Operation

Each i-th step function is defined as follows. (See the Fig. 2.)

$$H_{i+1} = A_i + SS_1(D_i) + f(B_i, C_i, D_i) + W_i + K_i, \tag{1}$$
$$B_{i+1} = C_i^{\lll 17}, \tag{2}$$
$$D_{i+1} = E_i + SS_2(H_i) + g(F_i, G_i, H_i) + W_i + K_i, \tag{3}$$
$$F_{i+1} = G_i^{\lll 2}, \tag{4}$$
$$A_{i+1} = B_i, \tag{5}$$
$$C_{i+1} = D_i, \tag{6}$$
$$E_{i+1} = F_i, \tag{7}$$
$$G_{i+1} = H_i. \tag{8}$$

2.8 Output of the Compression Function

$$(A_0 + A_{64}, B_0 + B_{64}, C_0 + C_{64}, D_0 + D_{64}, E_0 + E_{64}, F_0 + F_{64}, G_0 + G_{64}, H_0 + H_{64})$$

3 Design Principle

In this section, we will describe the design principles.

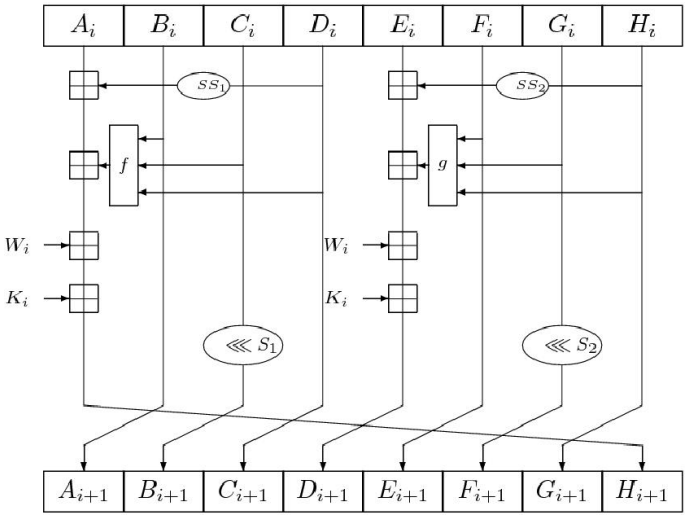


Fig. 2. Step Operation of DHA-256

3.1 Boolean Function

The boolean function F and G have SAC(Strict Avalanche Criterion) property which means that for any 1 bit input difference the output difference becomes zero with probability 1/2. In fact, it is easy to construct a differential characteristic with boolean function satisfying SAC property because it is possible to control the difference avalanche only with the property of boolean function.

3.2 Shift Rotation at Step Operation

In order to choose the rotation values of SS_1, SS_2, S_1 and S_2 at step function, we executed the exhaustive searching for each values. In the case of SS_1 and SS_2, one bit is fixed into '0'. When a one bit difference is injected at message word W_i, two 1 bit differences are injected to two register D_{i+1} and H_{i+1}. After eight steps, the hamming weight of the registers $A,\ B, \ldots,\ G$ and H is checked for every value of SS_1, SS_2, S_1 and S_2. As a result, we obtained a few candidates and we chose values that are satisfied with the following conditions:

- Each value of SS_1 and SS_2 is not a factor of 32 which is the length of a word.
- Each value is not close to the others.

3.3 Message Expansion

Our message expansion is similar to that of SHA-256. As σ_1 and σ_2 are invertible, given message words at 16 consecutive steps(Step (i-15) $\sim$ i), we can compute

the message word value at step (i-16). The message expansion is designed by following principles. These principles help the minimum hamming weight of the disturbance vector high.

- Principle to Select 4 Word Positions Needed to Update One Word
  - Our task is to select a,b,c,d for $W_i = \sigma_1(W_{i-a}) + W_{i-b} + \sigma_2(W_{i-c}) + W_{i-d}$. At first, we choose 1, 15 and 16 because of following reasons :
    - 'a=1' : The message word at step i-1 is used to update the message word at step i. This condition and σ_1 make non-zero differences expand rapidly in several steps.
    - 'd=16' : If there is no this condition, the first message word does not influence any of the expanded message words.
    - 'c=15' : In the case of the inverse operation of message expansion, the reason to select '15' is same as the reason to select '1'. If there is no this condition, this makes non-zero differences expand slowly in the case of the inverse operation. This weakness can be used to find minimum hamming weight of the disturbance vector.
  - 'b=9' : We select b values and rotation values of σ_1 and σ_2 together by exhaustive search so as to make the hamming weight of the disturbance vector high.
- Principle to Select rotation values of σ_1 and σ_2
  - We select b values and rotation values of σ_1 and σ_2 together by exhaustive search so as to make the hamming weight of the disturbance vector high. In the case of σ_1 and σ_2, one bit is fixed into '0'. As it is impossible to search all 2^{512} cases of differences which we have to consider, we use a trick to find the hamming weight of the disturbance vector as low as possible. The simulation will be explained in the next section.

4 Security Analysis

In this section, our aim is to quantify the security of DHA-256 against Wang *et al.*'s attack. In order to accomplish this task, the following two analyses have to be considered.

- Finding Best Inner Collision Pattern
  - Wang's attack uses the best inner collision pattern repeatedly. So it is crucial to lower the probability of the pattern. As the collision pattern is determined by the step operation, we designed the step operation of DHA-256 such that the probability of the pattern are as low as possible.

- Finding Minimum Hamming Weight of the Disturbance Vector
  - The disturbance vector is determined only by the message expansion. And the hamming weight of the disturbance vector means the frequency of the repetition of the inner collision pattern. Therefore it is important to design the message expansion function so that the hamming weight of the vector is as high as possible.

- In Wang *et al.*'s attack, part of the disturbance vector corresponding to step 0-23 of DHA-256 or SHA-256 may not be helpful to lower the success probability of the attack because first 24 steps may be corrected by modification method.
- Before Wang *et al.*'s attack was introduced, -5,-4,-3,-2,-1th position of the vector of SHA-0,1(in the case of SHA-256 and DHA-256, -8,-7,...,-3,-2,-1th position) have to be all zeros in order to find a collision pair because cryptanalysts thought that the non-zero differences on -5,-4,-3,-2,-1th position mean the non-zero differences of the initial values. But Wang *et al.* broke down the barrier by showing that non-zero differences on different bit positions can offset by using both the relation between the addition operation and the XOR operation and the properties of boolean functions.
- Wang *et al.*'s attack uses a near collision pair which can help to find collisions of more than 1 block. Therefore no condition is granted on last 5 position of the disturbance vector of SHA-0,1(In the case of SHA-256 and DHA-256, last 8 position).

4.1 Finding Best Inner Collision Pattern

When 1 bit difference is injected, each propagates three bit differences by function SS_1 and SS_2. That is, if more than one bit difference are injected, we predict that more bits of differences are propagated by function SS_1 and SS_2. Thus to inject a one bit difference at a message W_i is the best strategy in order to construct a minimum weight difference pattern. A one bit difference injected at a message word W_i is canceled by (9) :

$$\begin{aligned} W_{i+2} &= SS_2(SS_1(W_i)) \\ W_{i+5} &= SS_1(W_i^{\lll 17}) + SS_2(W_i^{\lll 2}) \\ W_{i+8} &= W_i^{\lll 19} \end{aligned} \tag{9}$$

t	W_t	A	B	C	D	E	F	G	H
i	1	0	0	0	1	0	0	0	1
$i+1$	0	0	0	1	3	0	0	1	3
$i+2$	9	0	1	3	0	0	1	3	0
$i+3$	0	1	3	0	0	1	3	0	0
$i+4$	0	3	0	0	1	3	0	0	1
$i+5$	6	0	0	1	0	0	0	1	0
$i+6$	0	0	1	0	0	0	1	0	0
$i+7$	0	1	0	0	0	1	0	0	0
$i+8$	1	0	0	0	0	0	0	0	0

Table 6. Minimum weight differences in eight registers by (9)

Best message differences and register differences in Table 6 are computed by followings :

– Message differences

$$\begin{aligned} W_i &= 10000000000000000000000000000000 \\ W_{i+2} &= 10110001001001000000110010000000 \\ W_{i+5} &= 00000100000000010011000000001010 \\ W_{i+8} &= 00000000000001000000000000000000 \end{aligned}$$

– Differences of registers A, B, C, D, E, F, G, H at each steps

round	A	B	C	D	E	F	G	H
i	0	0	0	80000000	0	0	0	80000000
$i+1$	0	0	80000000	90040000	0	0	80000000	81000400
$i+2$	0	00010000	90040000	0	0	00000002	81000400	0
$i+3$	00010000	00012008	0	0	00000002	04001002	0	0
$i+4$	00012008	0	0	00000002	04001002	0	0	00010000
$i+5$	0	0	00000002	0	0	0	00010000	0
$i+6$	0	00040000	0	0	0	00040000	0	0
$i+7$	00040000	0	0	0	00040000	0	0	0
$i+8$	0	0	0	0	0	0	0	0

Table 7. The Best Inner Collision Pattern of DHA-256

In order to find the optimized inner collision pattern, we tested injecting one bit, two bit or three bit differences into a message word W_i. As a result of the test, we confirmed that the minimum weight difference is constructed when a one bit difference is injected to a message word W_i. When more than four bit differences are injected to a message word W_i, It has lower probability than the probability of collision pattern we found. We proved it at appendix A.

Here, we compute the probability of the inner collision pattern suggested in Table 7. Addition operations in each step functions are performed at the register A and D. Therefore we obtain the probability that the addition operations work like XOR operations from the hamming weight of differences in the register A and D. However differences of the most significant bits(MSB) are not considered because addition and XOR operations work identically for the most significant bits.

In the case of addition operations, the number of differential bits by addition for 9 steps are 36 bits(see Table 8). Since the probability that one bit difference makes higher bit difference is $\frac{1}{2}$, the probability that all of one-bit differences don't make higher bit differences is 2^{-36}.

In the case of the non-linear boolean functions such as the Choice function and the Majority function, the input differences of the non-linear boolean functions for 9 steps are 28 bits. Since the probability that one bit input difference of the boolean functions become a zero output difference is $\frac{1}{2}$ on an average, the

	Register A	Rigister D
i	$1(MSB)$	$1(MSB)$
$i+1$	$1(MSB)+2$	$1(MSB)+2$
$i+2$	$1(MSB)+8$	$1(MSB)+8$
$i+3$	0	0
$i+4$	1	1
$i+5$	6	6
$i+6$	0	0
$i+7$	0	0
$i+8$	1	1
Sum	36 (except for MSB)	

Table 8. Hamming Weight

probability that an output difference of the boolean functions for 9 steps is zero is bounded to 2^{-28}.

When we consider the previous two cases, the probability that there are the inner collision patterns for 9 steps is 2^{-64}. This is the highest probability that there are the inner collision patterns for 9 steps, and we explain it in detail at the appendix A.

4.2 Finding Minimum Hamming Weight of the Disturbance Vector Corresponding to Step 24 to Step 55

In order to choose the rotation values of σ_1 and σ_2 functions in the message expansion, we should find the values that the disturbance vector corresponding to step $24 \sim 55$ have the biggest minimum hamming weight.

For 16 words $\hat{W}_{16}, \ldots, \hat{W}_{31}$, one, two or three bits of differences are injected. Then we obtain $\hat{W}_{32}, \ldots, \hat{W}_{47}$ using the LFSR. Since there exists an inverse LFSR, we also obtain $\hat{W}_0, \ldots, \hat{W}_{15}$. For each case, we check a consecutive 32-words having the minimum hamming weight, and find the biggest one among them. The found 32-words correspond to step $24 \sim 55$ of the disturbance vector.

Table 9 shows the disturbance vectors corresponding to step 24 to step 55 of DHA-256(left) and SHA-256(right) where '0' is the position of a 1 bit difference. The minimum hamming weight of the disturbance vector corresponding to step $24 \sim 55$ is 63 in DHA-256 and 15 in SHA-256 at most.

5 Comparison between DHA-256 and SHA-256 with the number of Operation

Total number of operations in DHA-256 is similar to that of SHA-256. The step function of DHA-256 is designed to be processed in parallel. Table 10 compares the number of operations used in DHA-256 and SHA-256.

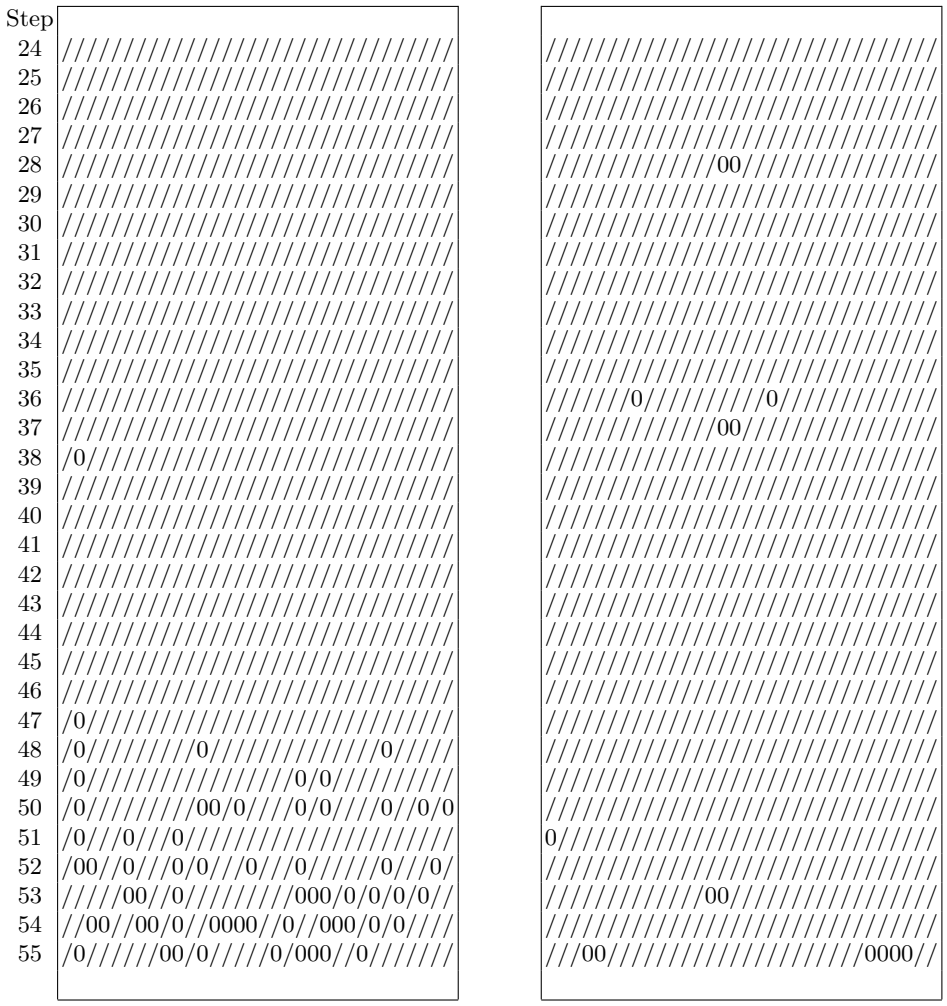


Table 9. The differences of the disturbance vector at 32-steps in DHA-256 and SHA-256(Here, '0' is a position of one bit difference. '0' also means the starting point of the best inner collision pattern.)

	DHA-256	SHA-256
step function	$+$: 8×64=512 $\oplus$: 10×64=640 $\ll$: 12×64=768	$+$: 7×64=448 $\oplus$: 10×64=640 $\ll$: 12×64=768
Message Expansion	$+$: 3×48=144 $\oplus$: 8×48=384 $\ll$: 8×48=384	$+$: 3×48=144 $\oplus$: 8×48=384 $\ll$: 10×48=480
Boolean Function	$\wedge$: 5×64=320 $\vee$: 1×64=64 $\neg$: 1×64=64 $\oplus$: 2×64=128	
Output	$+$: 8	

Table 10. Comparison between DHA-256 and SHA-256 with the number of Operation (We regard $\lll$ as 2 $\ll$ and 1 $\oplus$)

6 Conclusion

In this paper we have proposed a new dedicated 256-bit hash function DHA-256, which is designed not only to be secure but also to enhance the security of SHA-256. The main features are the followings;

- The step function is designed so that the probability of the inner collision pattern is as high as possible.
- The message expansion is designed so that the repetition of the inner collision pattern is as many as possible.
- By using one word twice at each step, it is difficult to construct a differential characteristic with high probability.
- These properties make it difficult to analyze DHA-256 with known attack methods including Wang *et al.*'s attack.

References

1. E. Biham and R. Chen. Near-Collisions of SHA-0. *Advances in Cryptology–Crypto'04*, volume 3152 of *Lecture Notes in Computer Science*, pages 290–350. Springer-Verlag, 2004.
2. E. Biham and R. Chen. New Results on SHA-0 and SHA-1. Rump Session at *Crypto'04*, August 2004.
3. E. Biham, R. Chen, A. Joux, P. Carribault, C. Lemuet and W. Jalby. Collisions of SHA-0 and Reduced SHA-1. *Advances in Cryptology–Eurocrypt'05*, volume 3494 of *Lecture Notes in Computer Science*, pages 36–57. Springer-Verlag, 2005.
4. B. den Boer and A. Bosselaers. Collisions for the Compression Function of MD5. *Advances in Cryptology–Eurocrypt'93*, volume 765 of *Lecture Notes in Computer Science*, pages 293–304. Springer-Verlag, 1994.
5. F. Chabaud and A. Joux. Differential collisions in SHA-0. *Advances in Cryptology–Crypto'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 56–71. Springer-Verlag, 1998.

6. H. Dobbertin. Cryptanalysis of MD4. *Journal of Cryptology 11:4 (1998)*, pages 253–271.
7. H. Dobbertin, A. Bosselaers and B. Preneel. RIPEMD-160, a strengthened version of RIPEMD. *FSE'96*, volume 1039 of *Lecture Notes in Computer Science*, pages 71–82. Springer-Verlag, 1996.
8. H. Gilbert and H. Handschuh. Security Analysis of SHA-256 and Sisters. *SAC'03*, volume 3006 of *Lecture Notes in Computer Science*, pages 175–193. Springer-Verlag, 2004.
9. P. Hawkes, M. Paddon and G. G. Rose. Security On Corrective Patterns for the SHA-2 Family. Cryptology ePrint Archive, Report 2004/207.
10. A. Joux. Collisions for SHA-0. Rump session at *Crypto'04*, August 2004.
11. FIPS 180 (superseded by FIPS 180-1 and FIPS 180-2).
12. FIPS 180-2: Secure Hash Standard (SHS) (change notice: February 2004).
13. Research and Development in Advanced Communication Technologies in Europe. RIPE Integrity Primitives: Final Report of RACE Integriy Primitives Evaluation (R1040). RACE, June 1992.
14. R. L. Rivest. The MD4 Message Digest Algorithm. *Advances in Cryptology–Crypto'90*, volume 537 of *Lecture Notes in Computer Science*, pages 303–311. Springer-Verlag, 1990.
15. R. L. Rivest. The MD5 Message Digest Algorithm. *RFC 1321 (1992)*.
16. R. L. Rivest. The MD4 Message Digest Algorithm. *Advances in Cryptology–Crypto'90*, volume 537 of *Lecture Notes in Computer Science*, pages 303–311. Springer-Verlag, 1990.
17. B. Van Rompay, A. Biryukov, B. Preneel and J. Vandewalle. Cryptanalysis of 3-pass HAVAL. *Advances in Cryptology–Asiacrypt'03*, volume 2894 of *Lecture Notes in Computer Science*, pages 228–245. Springer-Verlag, 2003.
18. X. Wang. The Collision attack on SHA-0. to appear on www.infosec.edu.cn, 1997.
19. X. Wang. The Improved Collision attack on SHA-0. to appear on www.infosec.edu.cn, 1998.
20. X. Wang, D. Feng, X. Lai and H. Yu. Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD. Cryptology ePrint Archive, Report 2004/199.
21. X. Wang, X. Lai, D. Feng, H. Chen and X. Yu. Cryptanalysis of the Hash Functions MD4 and RIPEMD. *Advances in Cryptology–Eurocrypt'05*, volume 3494 of *Lecture Notes in Computer Science*, pages 1–18. Springer-Verlag, 2005.
22. X. Wang and H. Yu. How to Break MD5 and Other Hash Functions. *Advances in Cryptology–Eurocrypt'05*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer-Verlag, 2005.
23. X. Wang and H. Yu. How to Break MD5 and Other Hash Functions. *Advances in Cryptology–Eurocrypt'05*, volume 3494 of *Lecture Notes in Computer Science*, pages 19–35. Springer-Verlag, 2005.
24. X. Wang, H. Yu and Y. L. Yin. Efficient Collision Search Attacks on SHA-0. *Advances in Cryptology–Crypto'05*, volume 3621 of *Lecture Notes in Computer Science*, pages 1–16. Springer-Verlag, 2005.
25. X. Wang, Y. L. Yin and H. Yu. Finding Collisions in the Full SHA-1. *Advances in Cryptology–Crypto'2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 17–36. Springer-Verlag, 2005.
26. Y. Zheng, J. Pieprzyk and J. Seberry. HAVAL – a one-way hashing algorithm with variable length of output. *Advances in Cryptology–Auscrypt'92*, volume 718 of *Lecture Notes in Computer Science*, pages 83–104. Springer-Verlag, 1993.

Appendix A

In this section, we show the following Lemma.

Lemma 1. *If $HW(W_i) = a \geq 4$, the probability that there exist the inner collision pattern is smaller than or equal to $\frac{1}{2^{64}}$.*

Proof. Assume that $HW(W_i) = a \geq 4$ and $\min\{HW(SS_1(W_i)) + HW(SS_2(W_i))\} = a + b$. The probabilities for the input differences of the boolean functions:

- $i+1$ th :
  $f(B,C,D) : HW(B_{i+1}) = 0, HW(B_{i+1}) = 0, HW(B_{i+1}) = a$
  $g(F,G,H) : HW(F_{i+1}) = 0, HW(G_{i+1}) = 0, HW(H_{i+1}) = a$
- $i+2$ th :
  $f(B,C,D) : HW(B_{i+2}) = 0, HW(B_{i+2}) = a, HW(B_{i+2}) = b$
  $g(F,G,H) : HW(F_{i+2}) = 0, HW(G_{i+2}) = a, HW(H_{i+2}) = c$
- $i+3$ th :
  $f(B,C,D) : HW(B_{i+3}) = a, HW(B_{i+3}) = bHW(B_{i+3}) = 0$
  $g(F,G,H) : HW(F_{i+3}) = a, HW(G_{i+3}) = c, HW(H_{i+3}) = 0$
- $i+4$ th :
  $f(B,C,D) : HW(B_{i+4}) = b, HW(B_{i+4}) = 0HW(B_{i+4}) = 0$
  $g(F,G,H) : HW(F_{i+4}) = c, HW(G_{i+4}) = 0, HW(H_{i+4}) = 0$

By the hamming weight of the input differences of the boolean functions from $(i+1)$-th step to $(i+4)$-th step, We can see that $6a + 3b + 3c = 6a + 3(b + c)$. According to the simulation, if $HW(X) \geq 4$, we can see that $Minb + c = 14$ where $HW(SS_1(X)) = b$ and $HW(SS_2(X)) = c$. Thus, $6a + 3(b + c) \geq 6 \cdot 4 + 3 \cdot 14 = 66$. Therefore, the probability that there exists the inner collision pattern when $HW(W_i) \geq 4$ is smaller than the probability of our optimized pattern.