Near Collisions for the Compress Function of Hamsi-256 Found by Genetic Algorithm

LI Yun-qiang[#1], Wang Ai-lan[#2]

[#] *Electronic Technique Institute, Information Engineering University*
450004 Zhengzhou, China

[1] yunqiangli@126.com

[2] yanjun_20082008@126.com

***Abstract*—Hamsi is one of 14 remaining candidates in NIST's Hash Competition for the future hash standard SHA-3 and Hamsi-256 is one of four kinds of Hamsi. In this paper we present a genetic algorithm to search near collisions for the compress function of Hamsi-256 , give a near collision on (256 − 20) bits and a near collision on (256 − 21) bits with four differences in the chaining value, and obtain a differential path for three rounds of Hamsi-256 with probability 2^{-24}, 2^{-23} respectively, which are better than previous work reported about near collisions.**

***key words*—Genetic Algorithm; near collisions; hash function; the SHA-3 hash function competition; the compress function of Hamsi-256; Hamsi.**

I. INTRODUCTION

Hash functions are fundamental components of many cryptographic applications such as digital signatures, random number generation, integrity protection, e-cash etc. For the sake of safety, National Institute of Standards and Technology (NIST) announced the SHA-3 hash function competition to select a new cryptographic hash function to be used as the new standard [1]. In November 2008, NIST received 64 submissions, among these, 51 candidates are considered to meet the minimum submission requirements. In July 2009, NIST announced 14 second round candidates.

Hamsi [2] is one of the second round candidates of the SHA-3 competition. Hamsi includes four kinds of Hash functions, Hamsi-224, Hamsi-256, Hamsi-384 and Hamsi-512. As usual, the research is focus on the 256-bit version Hamsi-256. Previous work by Nikolic reported [3] near collisions on (256 − 25) bits with 14 differences in the chaining value; work by Wang et al. reported [4] near collisions on (256 − 23) bits with 16 differences; work by Aumasson et al reported [5] near collisions on (256 − 25) bits with six differences in the chaining value. In this paper we present near collisions on (256 − 20) bits with only four differences in the chaining value by Genetic Algorithm (GA).

This paper is organized as follows. Section 2 briefly details the Hamsi-256 Hash function. Section 3 illustrates how to search near collisions for the compress function of Hamsi-256 by GA. Section 4 gives new collisions and new differential paths for three rounds compress function of Hamsi-256, which are better than previous work about near collisions. Finally, Section 5 summarizes the paper.

II. DESCRIPTION OF THE COMPRESSION FUNCTION OF HAMSI-256

In this section, we give a brief overview of the compression function of Hamsi-256. For the detail version, the reader may refer to the specification of Hamsi [2].

The compression function of Hamsi-256 inputs a 32-bit message block and a 256-bit chaining value and outputs a 256-bit chaining value. The function acts on a state of 512 bits, which can be considered as both a 4x4 matrix of 32-bit words and 128 columns each consisting of 4 bits. The compression function is a three round transformation consisting of the following transformations. Addition of Constants, Substitution and Diffusion are considered as a round.

Message Expansion and Concatenation. 32-bit message block is expanded to 256 bits using a linear code (128,16,70) over F_4. Then, the expanded message and the chaining value are concatenated to a 512-bit internal state of Hamsi-256.

Addition of Constants. The state is XOR'ed with predefined constants and a round counter.

Substitution. Each of the 128 columns of the state goes through a 4x4 s-box.

Diffusion. A linear transformation L, which inputs four 32-bit words, and outputs four 32-bit words is applied to the four independent diagonals of the state.

Truncation and Feed Forward. The second and fourth row of the state is dropped and the initial CV is feed-forwarded to the truncated state.

III. USING GENETIC ALGORITHM TO FIND NEAR COLLISIONS

GA[6] is inspired by Darwin's theory about evolution and is a powerful tool for global search and optimization. It can solve nonlinear problems by searching all spaces through selection, crossover and mutation operations to obtain the optimal solution.

The compress function of Hamsi-256 is a nonlinear function and searching its near collisions is an optimal problem, so we can consider to use GA to search its near collisions. In order to use GA , we must designed how to represent a solution, evaluate a solution and produce a new solution about the optimal problem.

Encoding. Encoding process is to design how to represent a solution. The data structure of an individual is represented as follows.

```
struct individual                  //data structure of individual
{
  unsigned long int diff[9];
  unsigned long int bestin[9];
  int bestweight;
 double fitness;
 bool changeflag;
};
```

Where *diff[9]* are chromosomes of an individual and represent the input XOR's difference about a 32-bit message and a 256-bit chaining value , *bestin[9]* store the best input gotten which the hamming distance is smallest about their hash values for the above input difference, *bestweight* stores the smallest hamming distance, *fitness* stores the fitness value of the input difference and *changeflag* records whether the input difference is a new input difference or not.

Evaluation. We use differential analysis method to evaluate every individual. The differential attack is one of the main attacks to hash function and some non-random properties of Hamsi-256 have been gotten by differential analysis [3-5,7]. The theory base of Evaluation is the fact: if an input difference has a high probability to get a near collision, using random input couples about the input difference, it's easy to find the smallest hamming distance of compress function value couples.

In order to evaluate an individual, firstly we generate a number of random input couples about the input difference. Secondly calculate the hamming distance between compress function value couples for every random input couples. Thirdly find the smallest hamming distance and the correspond input. Finally under the two cases the value of *bestweight* is instead of the smallest hamming distance and the values of *bestin[9]* are instead of the correspond input, one case is that the value of *changeflag* is true, the other case is that the value of *changeflag* is false and the smallest hamming distance is smaller than the value of *bestweight.*

New population. Selection, crossover and mutation are used to create a new population. The value of *fitness* is gotten by 257-*bestweight* for every individual of current population. We choose Roulette Wheel selection as selection operator, and-or crossover as crossover operator, bit inversion mutation as mutation operator. If an individual in new population isn't a copy individual for current population, the value of *changeflag* is instead of true.

Now we outline the genetic algorithm for searching near collisions as follows.

Algorithm 1
Step 1 Encoding.
Step 2 Generate initial population, set generation=0.
Step 3 Evaluation.
Step 4 If (generation==maximal generation) goto step 8.
Step 5 Generate new population, generation++.
Step 6 Evaluation.
Step 7 Keep the current best individual in new population, goto step 4.
Step 8 Hill climbing starts with the current best individual.
Step 9 Output report.

Where the chromosomes are generated randomly and the value of *changeflag* is set true for every individual in initial population. Hill climbing only change little for the value of *bestin[9]* and don't change chromosomes about the current best individual.

Parameters in Algorithm 1 are set more detail as follows: the population size is 100, the crossover probability is 0.6, the mutation probability was 0.01, the number of random input couples generated in Evaluation process is 2^{10} and the maximal generation is 20000.

IV. NEAR COLLISIONS FOUND FOR THE COMPRESSION FUNCTION OF HAMSI-256

We run Algorithm 1 forty times and got a near collisions on (256 − 20) bits in Table 1 and a near collisions on

(256 − 21) bits in Table 2 with only four differences in the chaining value. The chaining value and the message are used hexadecimal representation.

From above collisions, we respectively obtain a differential path for three rounds of Hamsi-256 with probability 2^{-24}, 2^{-23} in Table3,Table 4 . We compare our work with previous work reported about near collisions in Table 5 without using message (chaining value) modification technique. The input and the output of Sbox are also used hexadecimal representation.

TABLE 1 A NEAR COLLISION ON (256 − 20) BITS FOR THREE ROUNDS OF HAMSI-256

	CV	M
One input	f4c18c15 c66ecc08 019ec0ae f94d71b2 0a8b5e5a f9d9f8cf cbd0158f a57af069	0ce6d774
Other input	f4c18c15 c66ecc08 019cc0ae b94d71b2 0a8b5e5a f9d9f8cf cbd2158f e57af069	0ce6d774
Input difference	00000000 00000000 00020000 40000000 00000000 00000000 00020000 40000000	00000000
One output	905f3acc 9a9a3e57 7d092769 9a5d80dd 84ae8928 61010f33 2b1e1a63 4e230844	
Other output	905f2acc 9ada3e57 7503a729 da5d80dd 80ae882a 41012f73 2b1c1ae3 0e634944	
Output difference	00001000 00400000 080a8040 40000000 04000102 20002040 00020080 40404100	

TABLE2 A NEAR COLLISION ON (256 − 21) BITS FOR THREE ROUNDS OF HAMSI-256

	CV	M
One input	39e149b0 a3827634 2f98037f 8cf29769 e0f5fdfb 841974a3 a6fa7ff3 d5a895d3	6857c14a
Other input	39e149b0 a3827634 2f99037f acf29769 e0f5fdfb 841974a3 a6fb7ff3 f5a895d3	6857c14a
Input difference	00000000 00000000 00010000 20000000 00000000 00000000 00010000 20000000	00000000
One output	a9369570 728f65a5 e258fbeb b893d198 8146237 a0a24b7d c2f9089b 4c089661	
Other output	a9369d70 72af65a5 e65dbbcb b813d198 a14a236 b0a25b5d c2f808db 6c28b6e1	
Output difference	00000800 00200000 04054020 00800000 0200c001 10001020 00010040 20202080	

TABLE 3 Differential path for three rounds of Hamsi-256 with probability 2^{-24}

round	Sbox input	Sbox output	Prob.
1	00000000 00000000 00000000 00000000 00020000 40000000 00000000 00000000 00000000 00000000 00000000 00000000 00020000 40000000 00000000 00000000	00020000 00000000 00000000 00000000 00000000 40000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	2^{-6}
2	00000008 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000 00000008 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000008 00000000 00000000 00000000	2^{-3}
3	00000000 00008000 00000000 00000200 00000010 00000000 00000000 00000000 00000000 00000002 00000000 00000001 00000400 00000000 00000000 00000000	00000000 00000000 00000000 00000000 00000410 00008000 00000000 00000201 00000010 00000002 00000000 00000000 00000400 00008002 00000000 00000201	2^{-15}
Truncate	00001000 00400000 080a8040 40000000 04000102 20002040 00020080 40404100		

TABLE 4 DIFFERENTIAL PATH FOR THREE ROUNDS OF HAMSI-256 WITH PROBABILITY 2^{-23}

round	Sbox input	Sbox output	Prob.
1	00000000 00000000 00000000 00000000 00010000 20000000 00000000 00000000 00000000 00000000 00000000 00000000 00010000 20000000 00000000 00000000	00010000 00000000 00000000 00000000 00000000 20000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	2^{-6}
2	00000004 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000	00000000 00000000 00000000 00000000 00000004 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000004 00000000 00000000 00000000	2^{-3}
3	00000000 00004000 00000000 00000100 00000008 00000000 00000000 00000000 00000000 00000001 00000000 80000000 00000200 00000000 00000000 00000000	00000000 00000000 00000000 00000000 00000208 00004000 00000000 80000100 00000008 00004000 00000000 00000000 00000200 00004001 00000000 80000100	2^{-14}
Truncate	00000800 00200000 04054020 80000000 0200c001 10001020 00010040 20202080		

TABLE 5 COMPARE OUR WORK WITH PREVIOUS WORK REPORTED ABOUT NEAR COLLISIONS

Whose work	Input difference number	Output difference number	probability
Nikolic's work	14	25	2^{-28}
Wang's work	16	23	2^{-31}
Aumasson's work	6	25	2^{-26}
Our work	4	20	2^{-24}
Our work	4	21	2^{-23}

V SUMMARY

In this study, we present a genetic algorithm to search near collisions of the three round compress function of Hamsi-256, which is one of the 14 second round candidates of the SHA-3 competition. Using the genetic algorithm, we give a near collision on (256-20) bits and a near collision on (256-21) bits with four differences in the chaining value. Using those near collision, we obtain new differential paths which are better previous work.

The genetic algorithm can be use further. On the one hand we can use it directly to get near collision for 4,5,6 rounds compress function of Hamsi-256, on the other hand we can modify it to try to find real collisions of Hamsi-256, which is a difficult and meaning work.

REFERENCES

[1] NIST. Announcing Request for Candidate Algorithm Nominations for a New Cryptographic Hash Algorithm (SHA-3) Family. Federal Register, 27(212):62212{62220, 2007. Available at:http://csrc.nist.gov/ groups / ST/hash/documents/FR Notice Nov07.pdf.

[2] Őzgül Kücük. : The hash function Hamsi. Submission to NIST, 2008.

[3] Nikolić I. : Near collisions for the compression function of Hamsi-256. CRYPTO rump session (2009), http://rump2009.cr.yp.to/ 936779b3afb9b48a404b487d6865091d.pdf.

[4] Wang, M., Wang, X., Jia, K., Wang, W.: New pseudo-near-collision attack on reduced-round of Hamsi-256.Cryptology ePrint Archive, Report 2009/484 (2009).

[5] J. P. Aumasson, E. Käsper, L.R. Knudsen et al.: Differential distinguishers for the compression function and output transformation of hamsi-256. Cryptology ePrint Archive, Report 2010/91 (2010).

[6] Xiaoping, C. Liming.: Genetic Algorithm.Theory, Application and Software Implementation.. Xi.an University of Communication Publishers. (2002) (in Chinese).

[7] Calik, C., Turan, M.S.: Message recovery and pseudo-preimage attacks on the compression function of Hamsi-256. Cryptology ePrint Archive, Report 2010/057 (2010)..