INTEGRAL DISTINGUISHERS OF JH AND GRØSTL-512[1]

Li Yanjun[*,**,***] Wu Wenling[*,**] Dong Le[*,**]

[*](*State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China*)

[**](*Graduate University of Chinese Academy of Sciences, Beijing 100049, China*)

[***](*Beijing Electronic Science and Technology Institute, Beijing 100070, China*)

Abstract In December of 2010 NIST selected five SHA-3 finalists - BLAKE, Grøstl, JH, Keccak, and Skein to advance to the third (and final) round of the SHA-3 competition. At present most specialists and scholars focus on the design and the attacks on these hash functions. However, it is very significant to study some properties of their primitives and underlying permutations. Because some properties reflect the pseudo-randomness of the structures. Moreover, they help us to find new cryptanalysis for some block cipher structures. In this paper, we analyze the resistance of JH and Grøstl-512 against structural properties built on integral distinguishers. And then 31.5 (out of 42) rounds integral distinguishers for JH compression function and 11.5 (out of 14) rounds for Grøstl-512 compression function are presented.

Key words Hash function; SHA-3 candidates; Integral distinguisher; JH; Grøstl

CLC index TN918.4

DOI 10.1007/s11767-012-0779-x

I. Introduction

In 2007, National Institute of Standards and Technology (NIST) announced the SHA-3 competition calling for new hash function designs in order to find a replacement of SHA-2. Since then, many new hash function designs have been submitted to the competition and after almost four years of rat race only five of them were selected by NIST as the third round candidates. They are BLAKE, Grøstl, JH, Keccak and Skein, one of which will win the competition and eventually become the new hash function standard[1]. Some designers proved that properties of these hash functions can be reduced to the properties of the underlying building blocks. Notably, the joint analysis of hash functions and underlying block ciphers or permutations has led to considerations of new attack models for block ciphers, *e.g.* known key[2] or chosen key[3]. In fact, when block cipher inspired permutation structures are used as building blocks within hash functions, there is no secret key input, thus the building block is not only a known transformation, it is also a computable one. Interestingly enough, doing so has led to the discovery of more powerful new techniques to construct distinguishers and/or mount key recovery attacks for block ciphers back in the conventional unknown key model where ciphers are stand alone constructs instead of underlying hash functions, just as the first known related-key attacks on the full version of AES-256 and AES-192[3,4].

JH is a hash function that enters into the final round of SHA-3 competition[5]. In 2008, Mendel and Thomsen gave a generic preimage attack based on an observation on JH-512[6]. Besides, Rijmen, *et al.* utilized rebound attack to obtain a semi-free-start collision for 16 rounds of JH and a semi-free-start near-collision on the JH compression function[7]. Grøstl is also one of the five final round hash functions of SHA-3 competition. In 2009, Mendel, *et al.* proposed the rebound attack to attack hash functions Grøstl and Whirlpool[8]. Afterward, this method was improved in Refs. [9–11]. In Ref. [12]

[1] Manuscript received date: September 5, 2011; revised date: December 16, 2011.
Supported by the National Natural Science Foundation of China (No. 60873259 and No. 60903212), and Knowledge Innovation Project of the Chinese Academy of Sciences.
Communication author: Li Yanjun, born in 1979, female, Ph.D. candidate. State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China.
Email: liyjwuyh@163.com.

Peyrin gave an 11-round distinguisher of the compression function of Grøstl-512. Most of these cryptanalysis are based on differential character. For AES-like structure, there is another main cryptanalysis called integral cryptanalysis. As shown in Refs. [13,14], particular integral properties essentially depending on the structure of the linear part of the cipher could be exhibited and are powerful tool for cryptanalytic studies. Those integral properties starting from the middle of the cipher lead to distinguishers in the known key settings defined in Refs. [2,14] and for the underlying compression functions of hash functions. This property of many kinds hash compression functions was analyzed. For example, in Crypto-2010 Boura proposed a zero-sum property for the 18 rounds Keccak-f permutation[15]. Grøstl was analyzed by the designers[16]. And the integral distinguishers of compression functions of Hamsi and LANE also were constructed in Refs. [17,18].

In this paper, we mainly focus on integral properties and their applications in the known transformation model to find compression function structural properties of the two SHA-3 candidates: JH and Grøstl-512. In more detail, our contributions are as follows. For the compression function of JH, we present a new 31.5 (out of 42) rounds integral distinguisher. The distinguisher requires ignorable memory and 2^{768} time complexity. For the compression function of Grøstl-512, we improved the results of Minier[18] and propose an 11.5 rounds (out of 14) integral distinguisher, which needs ignorable memory and 2^{953} time complexity. Finally, we correct a slight error of Minier[18] in the Appendix A.

II. Related Work

The basic idea of Integral cryptanalysis comes from Square attack, which was first proposed by Daemen in Ref. [19] and it was one of the efficient attacks on block ciphers include AES[13]. In Ref. [20], Knudsen and Wagner analyzed this cryptanalysis method as a dual to differential attacks particularly applicable to block ciphers with bijective components, and they first proposed the definition of integral cryptanalysis. Recently, integral cryptanalysis has been proposed in the new cryptanalysis model called known key setting where the key is known to the attacker[2,14]. In the same setting, compression functions of hash functions could also be analyzed and some distinguishers have been proposed against SHA-3 candidates using integral properties[15,17,18].

Integral attack considers a particular collection of m bytes in the plaintexts and ciphertexts. The aim of this attack is to predict the values in the sums (*i.e.* the integral) of the chosen bytes after a certain number of rounds of encryption. Knudsen and Wagner also generalized this approach to higher order integrals: the original set to consider becomes a set of vectors which differ in d components and where the sum of this set is predictable after a certain number of rounds. The sum of this set is called a d-th-order integral. In the rest of this paper, the following definitions are essential:

Active Set: A set $\{x_i \mid x_i \in F_{2^n}, 0 \le i \le 2^n - 1\}$ is active, if for any $0 \le i < j \le 2^n - 1$, $x_i \ne x_j$.

Constant Set: A set $\{x_i \mid x_i \in F_{2^n}, 0 \le i \le 2^n - 1\}$ is passive, if for any $0 \le i \le 2^n - 1$, $x_i = x_0$.

Balanced Set: A set $\{x_i \mid x_i \in F_{2^n}, 0 \le i \le 2^n - 1$ is balanced, if the sum of all element of the set is 0, that is $\sum_{i=0}^{2^n-1} x_i = 0$.

Usually, Active Set is denoted by A. Constant Set is denoted by C. And Balanced Set is denoted by B.

III. Integral Distinguisher of JH Compression Function

1. Description of JH

JH is one of the five finalists of SHA-3 competition[5]. It is an iterative hash function with a generalized AES design and a new compression function structure. The message that the function can process is a multiple of 512 bits, and it produces a digest of 224, 256, 384, and 512 bits. In JH hash function, padding is needed firstly to get a multiple of 512 bits message. It is performed by appending the bit "1", followed by $384 - 1 + (-l \bmod 512)$ zero bits, where l is the length of the message. At last, a 128-bit block is appended which is the binary representation of l. Note that this procedure guarantees that the length of the additional bits is at least 512. To process the iterated hash, the padded message is split into n 512-bit blocks, $M_1, M_2, \cdots, M_n$.

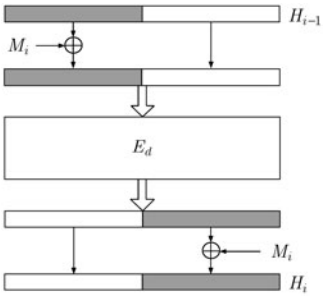


Fig. 1 The JH compression function structure

The compression function of JH adopts a new structure (see Fig. 1). It is applied to generate H_1, $H_2, \cdots, H_n$ iteratively:

$$\begin{aligned} H_i &= F_d\left(H_{i-1}, M_i\right) \\ &= E_d\left(H_{i-1} \oplus \left(M_i \left\| 0^{2^{d+1}}\right.\right)\right) \oplus \left(0^{2^{d+1}} \left\| M_i\right.\right) \end{aligned} \quad (1)$$

where F_d is the compression function and E_d is a bijective function. Note that the subscript d denotes the dimension of the JH state.

There are some operations used in JH as follows.

(1) The grouping operates in a bit-slice way which cut the input words and rearranges the bits to achieve efficient bit-slice software implementation.

(2) There are two 4×4 S-boxes used in JH. Which S-box is used depends on the round constant bit. We denote the S-box layer by S.

(3) The linear diffusion layer L transforms two 4-bit words (X_1, X_2) into two 4-bit words (Y_1, Y_2) as

$$\left(Y_1, Y_2\right) = \mathrm{L}\left(X_1, X_2\right) = \left(5X_1 + 2X_2, 2X_1 + X_2\right) \quad (2)$$

(4) The permutation P_d is similar to the ShiftRows operation in AES. If $a_1, \cdots, a_{15}$ and b_1, $\cdots, b_{15}$ denote the input and output of the permutation respectively, P_4 can be presented as Fig. 2.

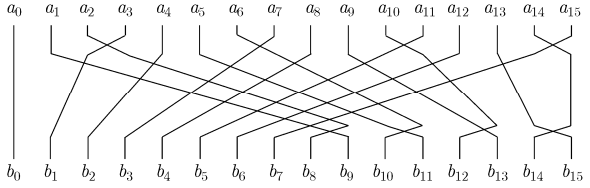


Fig. 2 The permutation P_4

The round function used in the compression function of JH can be denoted by

$$R_d = P_d \cdot \mathrm{L} \cdot \mathrm{S} \quad (3)$$

The bijective function E_d consists of three parts: grouping the input and then, calling the round function $6(d-1)$ times, finally being a de-group operation. The recommended number of dimensions is 8 so that it has 42 rounds in total. In addition, the size of the state is 1024 and the message digest is generated by truncating the last state H_n. For a detailed description of JH we refer to Ref. [5].

2. The 31.5 rounds integral distinguisher

The function E_d in the JH compression function is based on SP structure. Observing E_4, the 3-round equivalent structure is easy to be obtained (Fig. 3). For other values of d, the $d-1$-round equivalent structures are easy to obtain just as Fig. 3. Based on the equivalent structure, we first construct integral distinguisher in the forward direction. Here, $x_{i,j}$ is denoted the j-th 4-bit word of the input of i-th round and $c_{i,j}$ is constant of the j-th 4-bit word of the input of i-th round. The following properties are helpful.

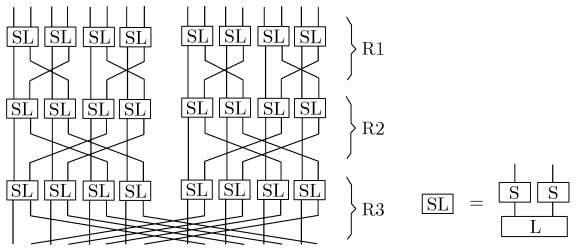


Fig. 3 The 3-round equivalent structure of E_4

Property 1 Let the 4 words of $x_{0,0}, x_{0,2}, x_{0,4}, x_{0,6}$ be independent active and the other words be constant, then after 5 rounds of encryption the output words will be balanced.

Proof Each input word will diffuse in all words after the 4th round of encryption. We consider the two words that are the input of one SL, for example, $x_{4,1}$ and $x_{4,2}$, the expressions of which are as follows:

$$x_{4,0} = 5S\left(x_{3,0}\right) \oplus 2S\left(c_{3,1}\right),\ x_{4,1} = 5S\left(x_{3,2}\right) \oplus 2S\left(c_{3,3}\right) \quad (4)$$

where

$x_{3,0} = 5S\{5S[5S(x_{0,0}) \oplus 2S(c_{0,1})] \oplus 2S[2S(x_{0,2}) \oplus$

$S(c_{0,3})]\} \oplus 2S\{2S[5S(x_{0,4}) \oplus 2S(c_{0,5})] \oplus S[2S(x_{0,6}) \oplus S(c_{0,7})]\}$,
$x_{3,2} = 5S\{5S[5S(x_{0,2}) \oplus 2S(c_{0,3})] \oplus 2S[2S(x_{0,0}) \oplus S(c_{0,1})]\} \oplus 2S\{2S[5S(x_{0,6}) \oplus 2S(c_{0,7})] \oplus S[2S(x_{0,4}) \oplus S(c_{0,5})]\}$.

It is easy to obtain that $x_{4,1}$ and $x_{4,2}$ are active words. So $x_{5,0} = 5S(x_{4,0}) \oplus 2S(x_{4,1})$ is a balanced word. Other words of $x_{5,i}$ are all balanced.

Q.E.D.

By choosing more active words instead of the only 4 bytes in Property 1, a distinguisher with more rounds in the forward direction will be obtained.

Property 2 Let $x_{0,8}, x_{0,10}, x_{0,12}, x_{0,14}$ be constant and the other words be independent active, then after 8 rounds encryption the output words will be balanced.

Proof The first half (32 bits) is still active after the 3rd round SP layer, which is independent of the last half (32 bits). That is to say, after 3 rounds encryption there are 2^{32} sets, and in each set the 4 words of $x_{3,0}, x_{3,2}, x_{3,4}, x_{3,6}$ are independent active. According to Property 1, after 5 more rounds encryption the output words are all balanced. An 8 rounds integral distinguisher will be obtained.

Q.E.D.

Using the same input as in Property 2, the backward rounds distinguisher will be constructed.

Property 3 Let $x_{0,8}, x_{0,10}, x_{0,12}, x_{0,14}$ be constant and the other words be independent active, then after 7.5 rounds decryption the output words will be balanced.

Proof This process is just like that of Property 2, however, in the direction of decryption the linear diffusion layer is before the inverse of S-box layer. So after the 8th linear diffusion layer all output words are balanced, and whether the output of the 8th inverse of S-box layer is balanced canot be analyzed. Therefore, there is 7.5 rounds distinguisher in the direction of decryption. Q.E.D.

According to Properties 2 and 3, there is a 15.5 rounds integral distinguisher for the permutation E_4. In fact, if the value of $x_{0,7}, x_{0,9}, x_{0,11}, x_{0,13}$ is constant and other words are independent active, there is another 15.5 rounds integral distinguisher for E_4. This method of distinguisher construction can be popularized to other values of d. The result is shown in the following Theorem 1.

Theorem 1 Let $3 \times 2^{d-2}$ words including $x_{0,2i}$ $(i = 0, 1, \cdots, 2^{d-1} - 1)$ and $x_{0,2i+1} (i = 0, 1, \cdots, 2^{d-2} - 1)$ independent active, then $4(d-1) + 3.5$ rounds integral distinguishers of E_d will be constructed with $2(d-1) + 2$ forward rounds and $2(d-1) + 1.5$ backward rounds.

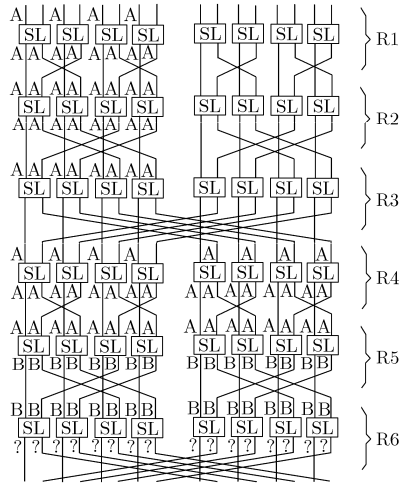


Fig. 4 The 4th order 6 rounds integral distinguisher

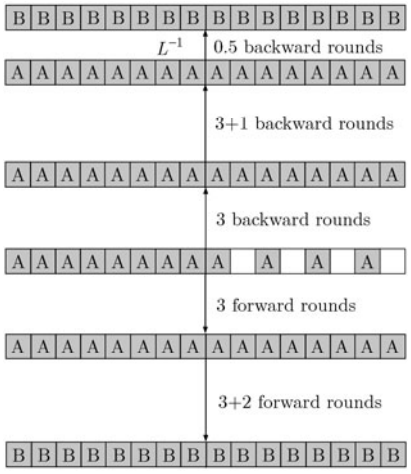


Fig. 5 The 15.5-round integral distinguishers for E_4

For the compression function of JH, $d = 8$, and we could combine those properties starting from the middle of E_d to build a structure property on the compression function when $31.5 = 4(8-1) + 3.5$ rounds are considered. For the permutation E_d, start from the middle with 2^{768} middletexts with 192 words, then go backward on 15.5 rounds to obtain inputs that sum to 0 on all words and go forward on 16 rounds to obtain outputs that sum to 0 on all words. That is to say, the sum taken over all

the 2^{768} outputs of the compression function is zero on all words and the corresponding inputs of H_{i-1} and M_i have 0-sum on all words. The time complexity is about 2^{768} operations and memory requirements could be ignored.

IV. Integral Distinguisher of Grøstl-512 Compression Function

1. Specification of Grøstl compression function

The compression function f of Grøstl is constructed using two AES-like permutations P and Q. A t-block message M (after padding) is hashed using the compress function $f(H_{i-1}, M_i)$ and the output transformation $g(H_t)$ as follows.

$$\begin{cases} H_0 = IV \\ H_i = f(H_{i-1}, M_i) = H_{i-1} \oplus P(H_{i-1} \oplus M_i) \oplus Q(M_i), \\ \qquad 1 \le i \le t \\ h = g(H_t) = \text{trunc}(H_t \oplus P(H_t)) \end{cases} \tag{5}$$

P and Q are constructed using the wide trail strategy, and their design is very similar to the AES with a fixed key input. Both permutations of Grøstl-512 act on a 1024-bit state represented as an 8×16 matrix of bytes and have 14 rounds. The round transformations of Grøstl-512 are the following ones:

AddRoundConstant (AC) adds different one-byte round constants to the 8×16 states of P and Q;

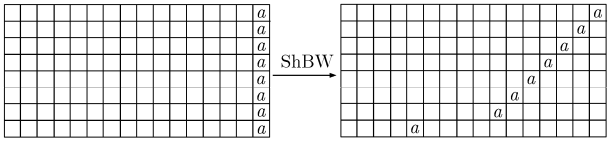
SubBytes (SB) is the non-linear layer that applies the AES Sbox to each byte of the state;

ShiftBytesWide (ShBW) rotates the bytes of row j left in the following ways: in the permutation of P, $j-1$ bytes for $j = 1,2,\cdots,7$, and 11 bytes for $j = 8$; in the permutation of Q, $2 \times (j-1)+1$ bytes for $j = 1,2,3$, $2 \times (j-5)$ bytes for $j = 5,6,7$, 8, and 11 bytes for $j = 4$ (in Fig. 6);

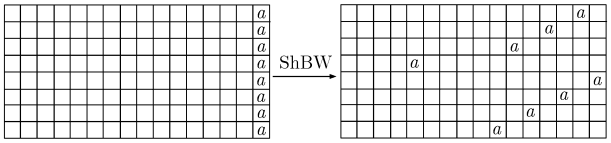
MixBytes (MB) is the linear diffusion layer where each column of the state is multiplied by a constant matrix.

2. The 11.5 rounds distinguisher of Grøstl-512 compression function

P and Q are two permutations used in Grøstl-512 compression function. Since the diffusion layer of them are slow, an 11.5 rounds distinguisher will be constructed in two steps. Firstly, we construct a 6 rounds integral distinguisher in the forward directon. In the following descriptions the linear transformation including ShBW and MB are dealt with as one half round, AC and SB as the other half round. And $x^r_{i,j}$ is the i-th row and j-th column byte of the r-th round input. $X_{r,j}$ denotes the j-th column of the r-th round output.



(a) The ShiftBytesWide transformations of P



(b) The ShiftBytesWide transformations of Q

Fig. 6 The ShiftBytesWide transformations

Property 4 Let $x^0_{0,3}, x^0_{1,4}, x^0_{5,8}, x^0_{6,9}$ four bytes be active and other bytes be constant as input, then after 3.5 encryption rounds there are still active bytes of the output state (in Fig. 7).

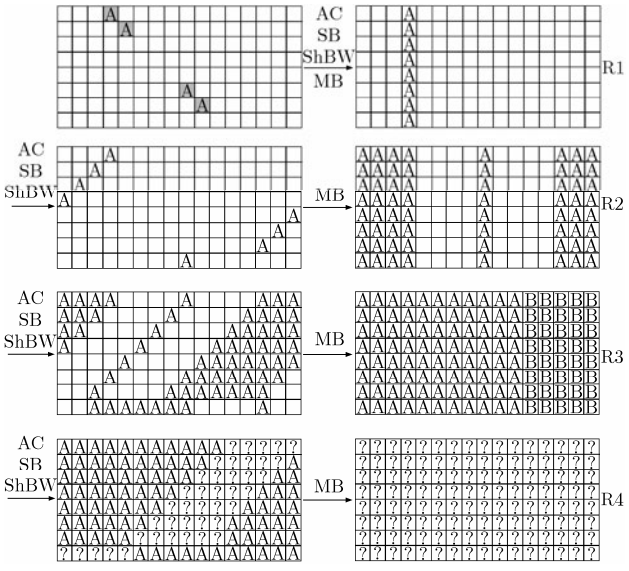


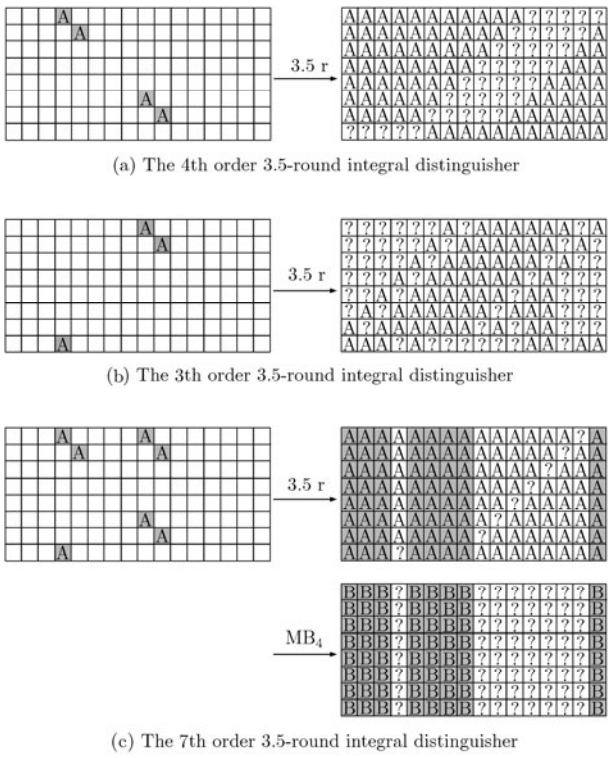
Fig. 7 The 4th order 3.5-round integral distinguisher

Proof Because there are four active bytes in the position of (0,3), (1,4), (5,8), (6,9) of the input state, after the 3rd ShBW transformation the columns that include no more than 4 active bytes

will be active after MB transformation. These active bytes can saturate the SB and ShBW of the 4th round and the result of Property 4 can be obtained. Q.E.D.

In order to saturate the integral property to the output of 4th round, we should use more active bytes as the input, just as shown in Property 5.

Property 5 Let $x^0_{0,3}, x^0_{1,4}, x^0_{5,8}, x^0_{6,9}, x^0_{0,8}, x^0_{1,9}, x^0_{7,3}$ seven bytes be active and the other bytes be constant as input, then after the 4th round there are 8 columns bytes balanced of the output state (in Fig. 8).



(a) The 4th order 3.5-round integral distinguisher

(b) The 3th order 3.5-round integral distinguisher

(c) The 7th order 3.5-round integral distinguisher

Fig. 8 The integral distinguisher with saturation technique

Proof When the 3 bytes of $x^0_{0,8}, x^0_{1,9}, x^0_{7,3}$ are active, the bytes of the column $X_{1,8}$ are active. After the 3rd round encryption the bytes of the 7 columns $X_{3,6}, X_{3,8}, X_{3,9}, X_{3,10}, X_{3,11}, X_{3,12}, X_{3,14}$ are all active as shown in Fig. 8(b). Combined with Property 4 the bytes of 8 columns are active before the transformation MB of the 4th round. So after the 4th round encryption the bytes of the 8 columns $X_{4,0}, X_{4,1}, X_{4,2}, X_{4,4}, X_{4,5}, X_{4,6}, X_{4,7}, X_{4,15}$ are all balanced as shown in Fig. 8(c). So the result of Property 5 can be obtained. Q.E.D.

The 4 rounds integral distinguisher can be improved by 2 more rounds by choosing more active bytes of the input, and then the 6 round integral distinguisher in the forward direction will be constructed as shown in Fig. 9(a). In the direction of decryption, the 5.5 round integral distinguisher will be constructed using the method described above as shown in Fig. 9(b). Based on the 6 round distinguisher in the forward direction and 5.5 round distinguisher in the backward direction the 11.5 round integral distinguisher is constructed.

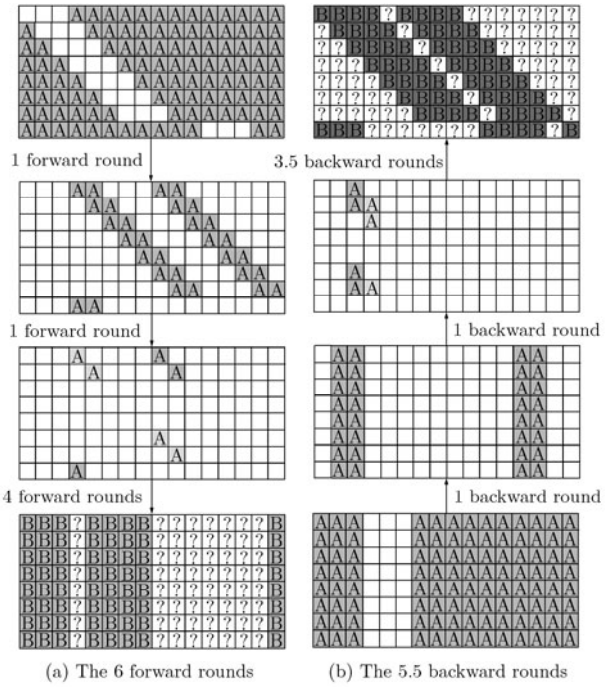


Fig. 9 The 6 forward rounds and 5.5 backward rounds

The set of active bytes we need to choose is the union of active sets of the two distinguishers inputs, just as shown in Fig. 10(a). This result is also shown in Theorem 2.

Theorem 2 Let the nine bytes of $x^0_{1,3}, x^0_{2,3}, x^0_{3,3}, x^0_{2,4}, x^0_{3,4}, x^0_{4,4}, x^0_{3,5}, x^0_{4,5}, x^0_{5,5}$ be constant and the other bytes be active, then an 11.5 round integral distinguisher of the permutation P will be constructed with 6 forward rounds and 5.5 backward rounds.

Although the diffusion layer of Q is different from that of P, the 11.5 rounds distinguisher of Q can be obtained by transforming the rows of P with the permutation (5, 1, 6, 2, 7, 3, 8, 4) (Fig. 10(b)). Based on these two 11.5 round distinguishers of P and Q, we build structure property on the compression function of Grøstl-512. For the permute-

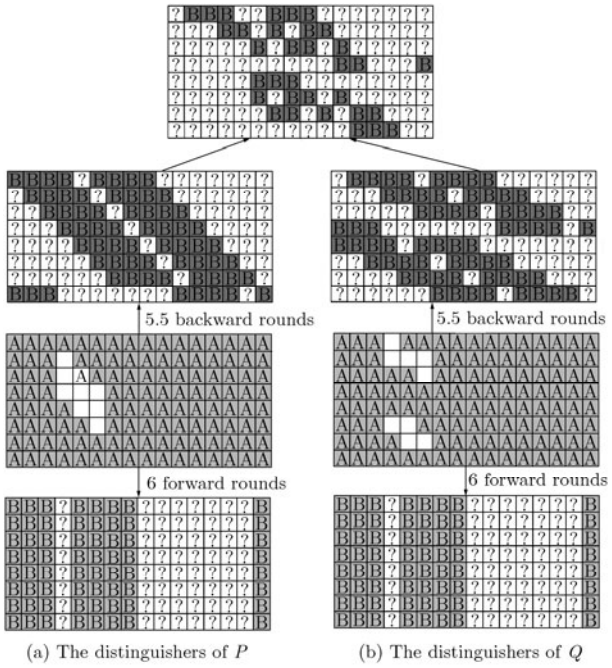


Fig. 10 The 11.5-round distinguishers of P and Q

tion P, start from the middle with 2^{952} middletexts with 119 active bytes, then go backward on 5.5 rounds to obtain inputs that sum to 0 on 8 shifted columns and go forward on 6 rounds to obtain outputs that sum to 0 on 8 columns. Do the same for the permutation Q. Using Q, we get the 2^{952} corresponding M_i messages. Using those messages and the inputs of P, we can compute the corresponding 2^{952} H_{i-1} values. Those 2^{952} values also verify that their sums taken over all the 2^{952} values on 8 columns are equal to 0 (Fig. 10). Then with the knowledge of H_{i-1}, of the outputs of P and of the outputs of Q, the corresponding H_i values are such that the sums taken over all the 2^{952} values on the intersection of the 8 shifted columns (for the backward sense) and of the 8 columns (for the forward sense) are equal to 0. In other words, the sum taken over all the 2^{952} outputs of the compression function is zero at 8 column positions whereas the corresponding inputs H_{i-1} and M_i have 0-sum on 33 bytes. Thus, we have exhibited a structural property of the Grøstl-512 compression function when P and Q are limited to 11.5 rounds. The time complexity is about 2^{953} operations and memory requirements could be ignored. Our new structural property really improves the one described in Ref. [18] that reaches 10 rounds.

V. Conclusion

In this paper, we analyzed the compression functions of two SHA-3 candidates (JH and Grøstl-512) in regard of integral properties. Due to the slow diffusion of the linear part of the structures, integral properties could be exhibited for a number of rounds greater than expected. And then a 31.5 round distinguisher of JH compression function and an 11.5 round distinguisher of Grøstl-512 compression function were proposed respectively. At last we sum up our results and the related works concerning distinguishers on the compression functions of these two SHA-3 candidates in Tab. 1.

Tab. 1 Results of distinguishers on JH and Grøstl-512

HASH	Attacks	D-Rounds*	Memory	Time**	Source
JH	semi-free-start meat coll.	22	$2^{143.7}$	$2^{156.77}$	Ref. [7]
	Integral	31.5	small	2^{768}	Subsection III.2 in this paper
Grøstl-512	semi-free-start coll.	8	2^{64}	2^{152}	Ref. [10]
	Integral	9	small	2^{704}	Ref. [16]
	Integral	10	small	2^{929}	Ref. [18]***
	Trunc.Diff	11	2^{64}	2^{640}	Ref. [12]
	Integral	11.5	small	2^{953}	Subsection IV.2 in this paper

*D-Rounds is Distinguisher Rounds.

**Time complexity is measured in encryption units.

***The result of Ref. [18] is the corrected result in Appendix A.

According to Tab. 1, the integral distinguishers presented in this paper make significant improvements. However, the full round compression functions of JH and Grøstl-512 provide a sufficient safety margin. How to evaluate the security of SP structure against integral distinguisher is more impor-

tant, which will be our future work.

Appendix A

There is an error in Fig. 7 in the page of 117 in Ref. [18], in which the active bytes in the middletext should be those bytes that shown in Fig. 11(a).

The active bytes set in the middle state of 10 rounds distinguisher is the intersection of the active bytes sets of the two middletext, just as shown in Fig. 11(b).

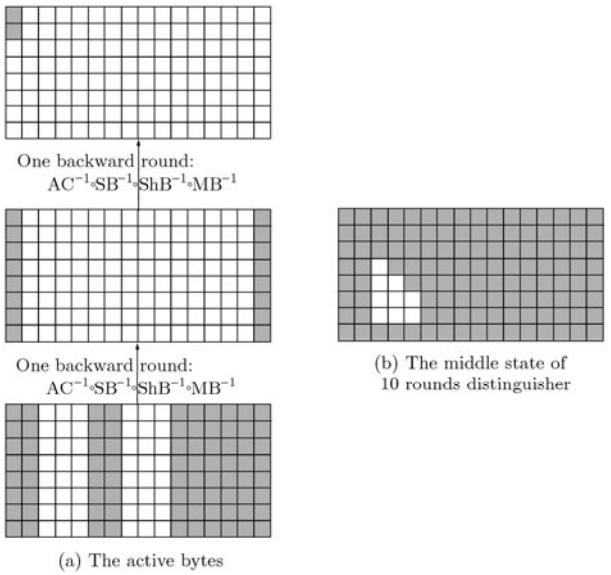


Fig. 11 The Corrected Integral Property

It is easy to obtain that the time complexity of 10 rounds distinguisher should be 2^{958} instead of 2^{512} proposed in Ref. [18] and the time complexity of the two function P and Q should be 2^{959}. The memory can be ignored just as in Ref. [18].

References

[1] The SHA-3 Zoo http://ehash.iaik.tugraz.at/wiki/The SHA-3 Zoo.

[2] L. R. Knudsen and V. Rijmen. Known-key distinguishers for some block ciphers. In: K. Kurosawa, (ed.), the 13th Annual International Conference on the Theory and Application of Cryptology & Information Security, Springer, Heidelberg, 2007, Lecture Notes in Computer Science, Vol. 4833, 315–324.

[3] A. Biryukov, D. Khovratovich, and I. Nikolic. Distinguisher and related-key attack on the full AES-256. In: S. Halevi (ed.), the 29th International Cryptology Conference, Springer, Heidelberg, 2009, Lecture Notes in Computer Science, Vol. 5677, 231–249.

[4] A. Biryukov and D. Khovratovich. Related-key cryptanalysis of the full AES-192 and AES-256. In: M. Matsui (ed.), the 15th Annual International Conference on the Theory and Application of Cryptology & Information Security, Springer, Heidelberg, 2009, Lecture Notes in Computer Science, Vol. 5912, 1–18.

[5] H. Wu. The Hash function JH. Submission to NIST, http://icsd.i2r.astar.edu.sg/ staff/ hongjun/jh/jh.pdf, 2008.

[6] F. Mendel and S. S. Thomsen. An observation on JH-512. http://ehash.iaik.tugraz.at/uploads/d/da/Jhpreimage.pdf.

[7] V. Rijmen, D. Toz, and K. Varici. Rebound attack on reduced-round versions of JH. The 17th International Workshop on Fast Software Encryption, Springer, Heidelberg, 2010, Lecture Notes in Computer Science, Vol. 6147, 286C303.

[8] F. Mendel, C. Rechberger, M. Schlaffer, *et al.*. The rebound attack: cryptanalysis of reduced whirlpool and Grøstl. The 16th International Workshop on Fast Software Encryption, Springer, Heidelberg, 2009, Lecture Notes in Computer Science 5665, 260–276.

[9] K. Ideguchi, E. Tischhauser, and B. Preneel. Improved collision attacks on the reduced-round Grøstl Hash function. The 13th Information Security Conference, Springer, Heidelberg, 2010, Lecture Notes in Computer Science 6531, 1–16.

[10] F. Mendel, C. Rechberger, M. Schlaffer, *et al.*. Rebound attacks on the reduced Grøstl hash function. The Cryptographers' Track at the RSA Conference 2010, Springer, Heidelberg, 2010, Lecture Notes in Computer Science, Vol. 5985, 350–365.

[11] Y. Sasaki, Y. Li, L. Wang, K. Sakiyama, and K. Ohta New Non. Ideal properties of AES-based permutations: applications to ECHO and Grøstl. The 16th Annual International Conference on the Theory and Application of Cryptology & Information Security, Springer, Heidelberg, 2010, Lecture Notes in Computer Science 6477, 38–55.

[12] T. Peyrin. Improved differential attacks for ECHO and Grøstl. Cryptology ePrint Archive, Report 2010/223, to appear in Crypto 2010. http://eprint.iacr.org/.

[13] S. Galice and M. Minier. Improving integral attacks against Rijndael-256 up to 9 rounds. In: S. Vaudenay, (ed.), AFRICACRYPT 2008, Springer, Heidelberg, 2008, Lecture Notes in Computer Science, Vol. 5023,

1–15.

[14] M. Minier, R. C. W. Phan, and B. Pousse. Distinguishers for ciphers and known key attack against Rijndael with large blocks. In: B. Preneel, (ed.), AFRICACRYPT 2009, Springer, Heidelberg, 2009, Lecture Notes in Computer Science, Vol. 5580, 60–76.

[15] C. Boura and A. Canteaut. A zero-sum property for the Keccak-f permutation with 18 rounds. National Institute of Standards and Technology mailing list, 2010.

[16] P. Gauravaram, L. R. Knudsen, K. Matusiewicz, *et al.*. Grøstl - a SHA-3 candidate. Submission to NIST, 2008.

[17] J. P. Aumasson, E. Kasper, L. R. Knudsen, *et al.*. Distinguishers for the compression function and output transformation of Hamsi-256. Cryptology ePrint Archive, Report 2010 / 091. The 15th Australasian Conference on Information Security and Privacy, http://eprint.iacr.org/.

[18] M. Minier, R. C. W. Phan, and B. Pousse. Interal distinguishers of some SHA-3 candidates. The 9th International Conference on Cryptology and Network Security, Springer, Heidelberg, 2010, Lecture Notes in Computer Science, Vol. 6467, 106–123.

[19] J. Daemen, L. Knudsen, and V. Rijmen. The block cipher Square. In: E. Biham, (ed.), the 4th International Workshop on Fast Software Encryption, Springer, Heidelberg, 1997, Lecture Notes in Computer Science, Vol. 1267, 149–165.

[20] L. Knudsen and D. Wagner. Integral cryptanalysis. In: J. Daemen, V. Rijmen, (eds.), the 9th International Workshop on Fast Software Encryption, Springer, Heidelberg, 2002, Lecture Notes in Computer Science, Vol. 2365, 112–127.